Państwowa Wyższa Szkoła Zawodowa
w Tarnowie

Informatyka Stosowana

INSTYTUT POLITECHNICZNY

PRACA INŻYNIERSKA

PIOTR KOZŁOWSKI

STEROWANIE GŁOSOWE ROBOTEM KAWASAKI W WYBRANYCH GRACH PLANSZOWYCH

PROMOTOR:
dr Robert Wielgat

Tarnów 2012

OŚWIADCZENIE AUTORA PRACY

OŚWIADCZAM, ŚWIADOMY ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZĄ PRACĘ DYPLOMOWĄ WYKONAŁEM OSOBIŚCIE I SAMODZIELNIE, I NIE KORZYSTAŁEM ZE ŹRÓDEŁ INNYCH NIŻ WYMIENIONE W PRACY.

. . . . . .

PODPIS

Serdecznie dziękuję mojemu promotorowi za cenne wskazówki, które pomogły mi ukończyć tą pracę.

Spis treści

1. Wstęp

Wiek w którym żyjemy, zwany jest często wiekiem komunikacji w różnych jej formach. Jest ona bardzo ważnym aspektem życia człowieka. Tematyka tej pracy związana jest z komunikacją werbalną na lini człowiek - maszyna. Jeszcze do niedawna sterowanie sprzętem elektronicznym za pomocą komend głosowych, było spotykane częściej w filmach s-f niż w rzeczywistych zastosowaniach. Obecnie jest to zadanie możliwe do wykonania, niemniej jednak, ze względu na skomplikowany charakter sygnału mowy, często trudne do realizacji.

Systemy rozpoznawania mowy są rozwijane od wielu lat, zarówno jako projekty komercyjne jak i naukowe. Obecnie większość telefonów komórkowych, tzw. „smartfonów" posiada aplikacje ułatwiające korzystanie z nich, pozwalające na głosowe wybieranie połączeń lub zamieniające mowę na tekst. Liczba zastosowań systemów rozpoznawania mowy ciągle rośnie, ze względu na oczywisty fakt, że mowa jest naturalnym oraz jakże wygodnym sposobem przekazywania myśli i odczuć człowieka. Uzyskiwana jest również coraz większa skuteczność takich systemów, jednakże nie tak duża, by pozwalała na bezpieczne i efektywne wykorzystywanie ich w każdym przypadku. Co za tym idzie, poprawianie skuteczności wiąże się z dalszym rozwojem tej dziedziny a więc może być ona bardzo ciekawym obiektem zainteresowań.

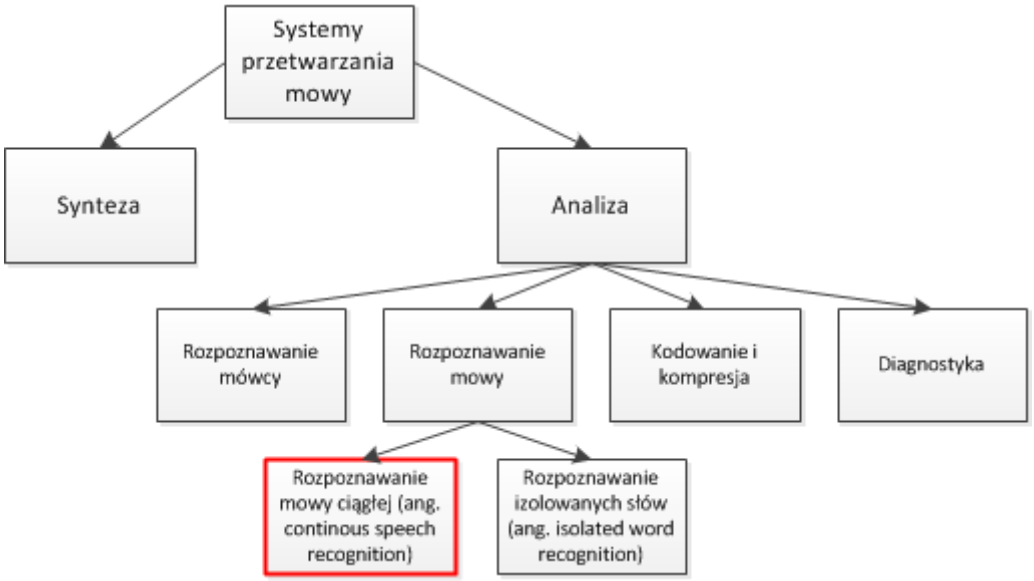
Przeglądając artykuły i prace dotyczące tematyki tutaj poruszanej często spotkać się można ze stwierdzeniem że problem rozpoznawania mowy dla języka polskiego, nie został dotychczas tak dobrze opracowany jak to zostało zrobione w przypadku innych języków, tzw. języków dominujących, takich jak angielski, niemiecki, francuski, hiszpański czy chiński. [13] Ta granica powoli zaczyna się zacierać, coraz więcej rozwiązań tworzonych jest również z myślą o języku polskim.

Niniejsza praca koncentruje się na wykorzystaniu osiągnięć projektu HTK [2] i stworzeniu systemu rozpoznawania mowy dla języka polskiego na przykładzie sterowania robotem przemysłowym w grze planszowej. Rozgrywka polega na wydawaniu komend głosowych które sterują ruchami robota. Jest on zarówno przeciwnikiem użytkownika jak również wykonuje za niego wszystkie ruchy podczas rozgrywki, użytkownik skupia się zatem tylko na wydawaniu poleceń głosowych. Gra z robotem jest formą demonstracji systemu stworzonego na potrzeby tej pracy, a więc jest jej poświęcone mniej uwagi niż tematyce rozpoznawania mowy.

2. Systemy rozpoznawania mowy

2.1. Podział zagadnień

Na początek warto określić miejsce systemów rozpoznawania mowy na tle innych zagadnień z tej dziedziny (rys. 2.1). Na czerwono zaznaczono dziedzinę w której znajduje się system stworzony na potrzeby tej pracy.



Rysunek 2.1: Ogólny podział systemów przetwarzania mowy

Systemy automatycznego rozpoznawania mowy, zwane ASR (ang. Automatic Speech Recognition) cechują się tym, że działają w czasie rzeczywistym (ang. real time). System ten ma za zadanie przeprowadzić detekcję sygnału mowy (wykryć kiedy została wypowiedziana jakaś sentencja) następnie jako wynik rozpoznawania utworzyć jej transkrypcję fonetyczną (zamienić mowę na tekst). Wszystko to powinno zostać zrobione automatycznie, tzn. bez ingerencji użytkownika. Taka sama idea przyświeca systemowi stworzonemu na potrzeby tej pracy, który działa dla mowy ciągłej.

2.2. Zastosowanie

Mowa jest najbardziej naturalnym oraz skutecznym sposobem porozumiewania się ludzi. Jest znacznie szybsza niż pisanie czy gestykulacja, a co najważniejsze nie wymaga użycia rąk. Nie dziwi więc

fakt, że wraz z rozwojem sprzętu komputerowego, wzrostem jego mocy obliczeniowej oraz miniaturyzacji, systemy rozpoznawania mowy mają coraz częstsze zastosowanie.

Oto lista przykładowych zastosowań takich systemów:

- sterowanie głosem urządzeń elektronicznych i elektromechanicznych,
- teleinformatyczne systemy informacyjne,
- systemy STT (ang. Speech To Text),
- urządzenia dla osób niepełnosprawnych,
- programy do nauki języków obcych,
- systemy diagnostyki medycznej i logopedycznej,
- systemy identyfikacji mówców. [19]

Warto zaznaczyć, że systemy rozpoznawania mowy, nie znajdą zastosowania tam gdzie sterowanie jest bardziej skomplikowane oraz wymaga od użytkownika szybkiej reakcji. Są sytuacje w których szybciej dotrzemy do odbiorcy (pozwolimy mu zrozumieć naszą ideę) używając rąk, np. za pomocą gestów lub rysunków niż przy pomocy komunikacji głosowej.

2.3. Skuteczność

Parametry wpływające na skuteczność rozpoznawania:

- stopień zależności od mówcy (systemy zależne (ang. speaker-depentent) i niezależne od mówcy (ang. speaker-independent)),
- liczba roznawanych słów (mała < 20 słów, duża > 20 000 słów),
- podobieństwo akustyczne i fonetyczne,
- SNR sygnału mowy (mały < 10 dB, duży > 30 dB),
- parametry transmisji sygnału mowy (zniekształcenia analogowego sygnału mowy wnoszone przez kanał transmisyjny, liczba poziomów kwantyzacji 8-16 bitów/próbkę, częstotliwość próbkowania 8-44 kHz). [19]

Skuteczność rozpoznawania zależy również od środowiska w jakim działa dany system. Może być ona inna dla zamkniętego pomieszczenia oraz otwartej przestrzeni. Negatywny wpływ mają tutaj wszelkiego rodzaju szumy (np. ruch uliczny, odgłosy rozmów innych ludzi). Nie bez wpływu pozostają również parametry sprzętu użytego do nagrań próbek głosu (np. redukcja szumu, czułość mikrofonu) jak również cechy samego mówcy. Istotny jest sposób wymowy (gwara, akcent), szybkość oraz głośność.

2.4. Przegląd gotowych rozwiązań

Wśród rozwiązań komercyjnych obecnie przodujących na rynku warto wymienić te przeznaczone dla urządzeń mobilnych, czyli m.in. telefonów komórkowych, które w obecnych czasach często posiadają możliwości zbliżone do komputerów osobistych. Firmy których produkty zdominowały rynek systemów przeznaczonych na urządzenia mobilne mają opracowane własne technologie rozpoznawania mowy połączone ze sztuczną inteligencją oraz syntezą mowy. I tak kolejno, Apple posiada Siri [9], Google - Google Now [11] a Microsoft - TellMe [16]. Produkty te, mają za zadanie ułatwić korzystanie z telefonu. Pełnią rolę wirtualnego asystenta, z którym można porozmawiać, zapytać o pogodę, najbliższą restaurację, poprosić o wpis do kalendarza lub przeczytanie sms'a. Produkt firmy Apple był najwcześniej wprowadzony na rynek więc wydaje się być technologią najbardziej zaawansowaną. Natomiast Google prócz nowej technologii Google Now, posiada również aplikację Google Mobile App [10], która służy m.in. do wyszukiwania głosowego (np. kontaktów, wiadomości lub innych rzeczy korzystając z wyszukiwarki internetowej Google). Poza produktami wymienionych wcześniej firm, powstaje coraz więcej aplikacji typu third-party (czyli tworzone przez osoby lub firmy niezwiązane z producentami systemów operacyjnych) posiadających podobne funkcjonalności (np. S-Voice firmy Samsung lub Iris, obie przeznaczone na platformę Android firmy Google).

Do najstarszych projektów komercyjnych korzystających z rozpoznawania mowy można zaliczyć IBM ViaVoice, Microsoft Speech API, Oracle Java Speech API oraz Dragon NaturallySpeaking firmy Nuance, która ostatecznie przejęła od IBM technologię ViaVoice [17]. Wymienione wyżej technologie (poza aplikacją Google Mobile App) nie są dostępne dla języka polskiego. Rozwiązania przeznaczone na urządzenia mobilne są uzupełniane o nowe języki więc w najbliższej przyszłości będą one dostępne również dla języka polskiego. W Polsce powstaje coraz więcej firm projektujących i wdrażających rozwiązania biznesowe oparte na rozpoznawaniu mowy, zwane systemami IVR (ang. Interactive Voice Response), umożliwiające interaktywną obsługę osoby dzwoniącej. Są to m.in. firma Primespeech (produkująca systemy przeznaczone na rynek telekomunikacyjny, np. dla Zarządu Transportu Miejskiego w Warszawie, działające na zasadzie wirtualnych konsultantów, informujących o rozkładach jazdy, cenach biletów, aktualnościach lub przyjmujący różnego rodzaju skargi klientów) [5], MagicScribe (systemy zamieniające mowę na tekst, MagicScribeMedical - system stworzony dla medycyny, wpomagający obsługę pacjenta, tworzenie dokumentacji medycznej itp. oraz MagicScribeLegal, rozwiązanie dla adwokatów, notariuszy, radców prawnych) [4], Stanusch Technologies S.A., VOICE LAB lub SkryBot.pl.

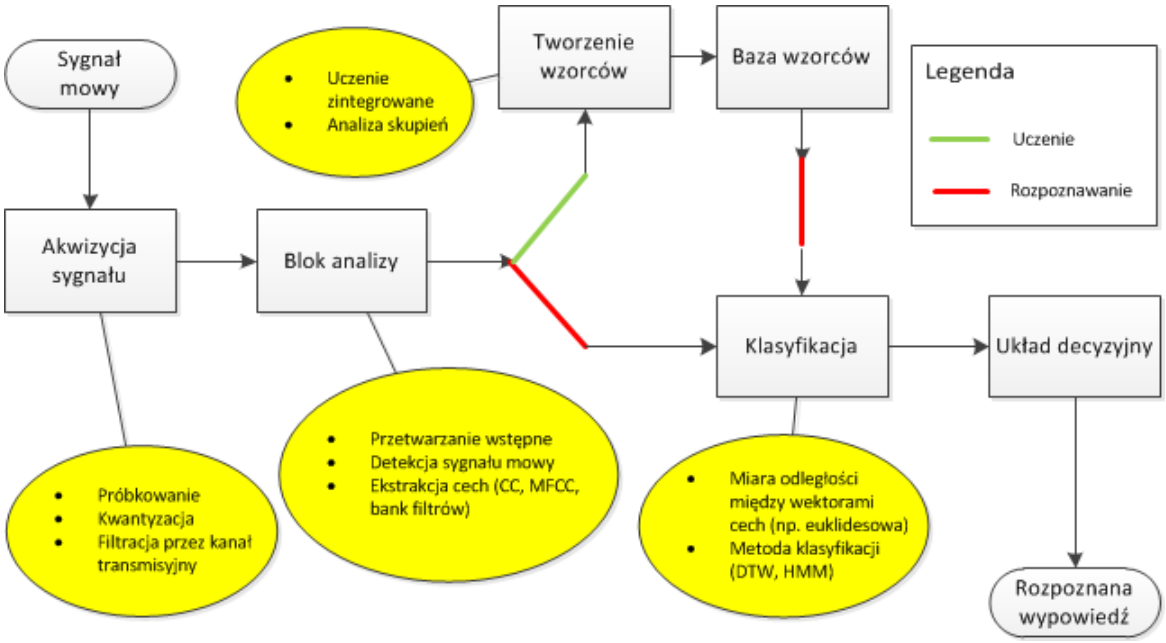
Z pośród projektów naukowych związanych z rozpoznawaniem mowy warto wymienić HTK [2], Julius [3], CMUSphinx [1] lub rodzimy projekt SARMATA (używany m.in. przez instytucje wymiaru sprawiedliwości do zarządzania dokumentacją procesową) [6].

W obecnych czasach tworzenie systemów rozpoznawania mowy jest dużo prostsze, ze względu na obecność gotowych narzędzi wspomagających, m.in. HTK czy CMUSphinx, które posiadają zaimplementowane algorytmy, pozwalające stworzyć system rozpoznawania mowy nawet osobie nie znającej szczegółów aparatu matematycznego. Użytkownik korzystający z takich bibliotek często używa ich jak „czarnej skrzynki" niemniej jednak potrafi stworzyć w pełni działający system. Na tyle skuteczny, by nie musieć zagłębiać szczegółów zaimplementowanych algorytmów. Co prawda taka niewiedza stwarza

problemy, przy doborze optymalnych parametrów mających istotny wpływ na działanie systemu lecz nie przeszkadza w doborze ich w sposób eksperymentalny i osiągnięciu zadowalającej skuteczności.

2.5. Schemat działania systemu

Na rysunku 2.2 jest widoczny ogólny schemat działania systemu rozpoznawania mowy.



Rysunek 2.2: Schemat blokowy systemu rozpoznawania mowy

Proces tworzenia systemu rozpoznawania mowy możemy podzielić na dwe fazy, uczenie (trenowanie) oraz rozpoznawanie. Trenowanie ma na celu poprawę skuteczności rozpoznawania, co za tym idzie odbywa się najpierw. Podczas tej fazy przy pomocy wypowiedzi uczących tworzone są i ulepszane wzorce z których korzystamy w trakcie rozpoznawania. Podczas rozpoznawania system dopasowuje najbardziej prawdopodobny wzorzec czego wynikiem jest transkrypcja fonetyczna nieznanej wypowiedzi. Szczegóły wymienionych wyżej faz opisane są w kolejnych rozdziałach.

2.6. Podział metod klasyfikacji

Współczesne systemy rozpoznawania mowy zakładają, że sygnał mowy jest sekwencją pewnych elementarnych jednostek fonetycznych (np. głosek, fonemów lub słów). Charakteruzyje się on również pewna przypadkowością. Wypowiedź tego samego mówcy za każdym razem posiada niepowtarzalny przebieg. A coza tym idzie nie możemy przyporządkować jej jednoznacznej sekwencji jednostek mowy. Wiąże się to z tym, że nagrany sygnał mowy często bywa zaszumiony i zniekształcony.

Metody klasyfikacji wykorzystywane w systemach rozpoznawania mowy możemy podzielić na dwie kategorie:

– metody deterministyczne, gdzie obliczane są błedy porównania dźwięku ze wzorcem,

– metody niederministyczne, gdzie obliczane są wartości prawdopodobieństw, które reprezentują dopasowanie dźwięku do stosowanych modeli probalistycznych (wzorców). [12]

Do metod deterministycznych należy m.in. nieliniowa transformata czasowa w skrócie DTW (ang. dynamic time wrapping), metoda skuteczna dla izolowanych słów.

Inne ograniczenia metod deterministycznych:

– duża zajętość pamięci potrzebnej na przechowywanie wzorców (dla dużych słowników),

– czas rozpoznawania proporcjonalny do rozmiaru słownika. [19]

Biorąc pod uwagę to, że bardziej naturalnym dla człowieka jest wypowiadanie słów ciągiem (bez znaczących przerw między nimi) oraz, że przy analizie sygnału mowy warto uwzględnić jego przypadkowość, metody niederministyczne osiągają lepszą skuteczność rozpoznawania. Do takich właśnie metod należą Ukryte Modele Markowa, w skrócie HMM (ang. hidden markov models).

3. Sygnał mowy

3.1. Mowa w języku polskim

Alfabet języka polskiego składa sie˛ z 32 liter. Łacińskie litery q, v i x występują jedynie w pisowni wyrazów obcych. Alfabet fonetyczny języka polskiego składa się z około 78 dźwięków. Dokładna liczba głosek zależy od sposobu traktowania wariantów. Relacja litera – głoska jest typu wiele – wiele. Tak jest również w wielu innych językach. Strukturę języka możemy przedstawić w postaci modelu warstwowego (rys. 3.1). Warstwa wyższa zawiera pewną liczbę elementów warstwy niższej.

myśl
wypowiedź
zdania
słowa
sylaby
głoski/fonemy

Rysunek 3.1: Warstwowy model języka

Badaniem struktury dźwiękowej języka zajmują się dwa pokrewne działy, fonetyka oraz fonologia. Fonetyka jest nauką o głoskach, czyli dźwiękach mowy. „Bada i opisuje dźwięki mowy ze wględu na ich właściwości fizyczne, tzn. ustala artykulacyjne i akustyczne ich cechy.” [8] Fonologia natomiast bada dźwięki mowy, pod kątem pełnionych przez nich funkcji w procesie komunikacji. Podstawową jednostką fonologii jest fonem. Fonem uznaje się za najmniejszy rozróżnialny segment dźwiękowy mowy, który może odróżniać znaczenie dzwieków mowy danego jezyka o różnicach wynikajacych wyłacznie z charakteru indywidualnej wymowy lub kontekstu. Dla systemów rozpoznawania mowy istotne jest to, że każdy fonem, jest zespołem cech dystynktywnych pozwalających na odróżnienie go od pozostałych fonemów, co przekłada się bezpośrednio na parametry akustyczne sygnału. [8]

Miarę podobieństwa między sygnałem mowy możemy podzielić na dwie kategorie, podobieństwo akustyczne oraz fonetyczne.

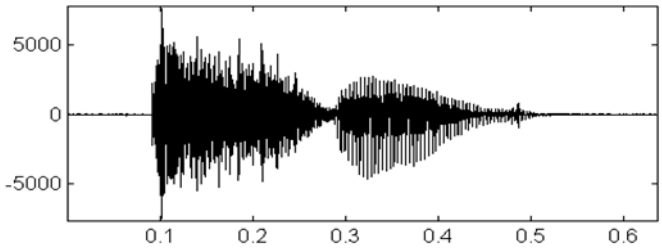
– Podobieństwo fonetyczne dwóch słów jest ustalane na podstawie tzw. odległości Levenshteina określanej jako minimalny koszt przekształcenia jednego słowa w drugie.

– Podobieństwo akustyczne dwóch słów jest ustalane na podstawie odległości (prawdopodobieństwa identyczności) między dwoma słowami wynikającej z przyjętych cech sygnału mowy w ramce oraz przyjętej metody klasyfikacji.

Zazwyczaj im słowa są bardziej podobne fonetycznie, tym bardziej są podobne akustycznie. Zdarzają się jednak sytuacje, w których zależność taka nie zachodzi. Jest to jedna z przyczyn pojawiania się błędów w rozpoznawaniu mowy. [19]

3.2. Reprezentacja sygnału mowy

Sygnał mowy może być opisywany na różne sposoby. Podstawowym i najprostszym jest opis w dziedzinie czasu (rys. 3.2). Ma on jednak skomplikowany przebieg, będący odzwierciedleniem złożonego charakteru artykulacji i często zawierający różne przypadkowe szumy pochodzące z otoczenia.



Rysunek 3.2: Reprezentacja słowa „trzy” w dziedzinie czasu

Matematycznie możemy go przedstawić jako splot przebiegu czasowego sygnału źródła $u_z\,(t)$i odpowiedzi impulsowej kanału głosowego $h\,(t)$, czyli:

$$u\,(t) = \int_0^{\tau} h(t-\tau)u_z(\tau)d\tau$$

Źródłem sygnału mogą być drgania wiązadeł głosowych (głoski dźwięczne) jak i szum powstający w skutek przepływu powietrza przez narządy mowy (głoski bezdźwięczne). [15]

Zanim jednak otrzymamy powyższy przebieg sygnału mowy, musi być wykonana jego dyskretyzacja, czyli próbkowanie i kwantyzacja. Z punktu widzenia rozpoznawania mowy, reprezentacja czasowa sygnału mowy nie przenosi wystarczającej ilości informacji.[14] Potrzeba zatem dokonać transformaty Fouriera sygnału w celu umożliwienia jego analizy w dziedzinie częstotliwości. Wadą klasycznej transformacji Fouriera jest brak jawnych informacji o czasie w widmie sygnału, która jest bardzo istotna w analizie sygnału mowy, ponieważ jest on ciągiem pewnych zdarzeń (zmian częstotliwości, amplitudy oraz następstw fonemów i słów), których kolejność jest bardzo istotna. Dlatego stosuje się tzn. krótkookresową transformację Fouriera (STFT ang. Short-Time Fourier Transform) zwaną też okienkową transformacją Fouriera (ang. Windowed Fourier Transform). Polega ona na dokonywaniu transformat krótkich fragmentów sygnału wyznaczonych za pomocą okna $w(t)$. Dyskretna realizacja tego przekształcenia ma postać:

$$F(k,\tau)=\frac{1}{\sqrt{M}}\sum_{t=0}^{\mathrm{M-1}}[f_{\tau+t}e^{-i2\pi\mathrm{kt/M}} * w_{\tau}^{t}]$$

$F(k,\tau)$ jest transformatą Fouriera, dla pojedynczej ramki (o indeksie τ, częstotliwości k, oraz szerokości M) otrzymanej w chwili $t=\tau$ uzyskaną wskutek złożenia sygnału $f(t)$ z funkcją okna w.

Szerokość okna determinuje rozdzielczość częstotliwościową i czasową otrzymanego widma czasowo-częstotliwościowego. Najprostszą funkcją okna jest okno prostokątne:

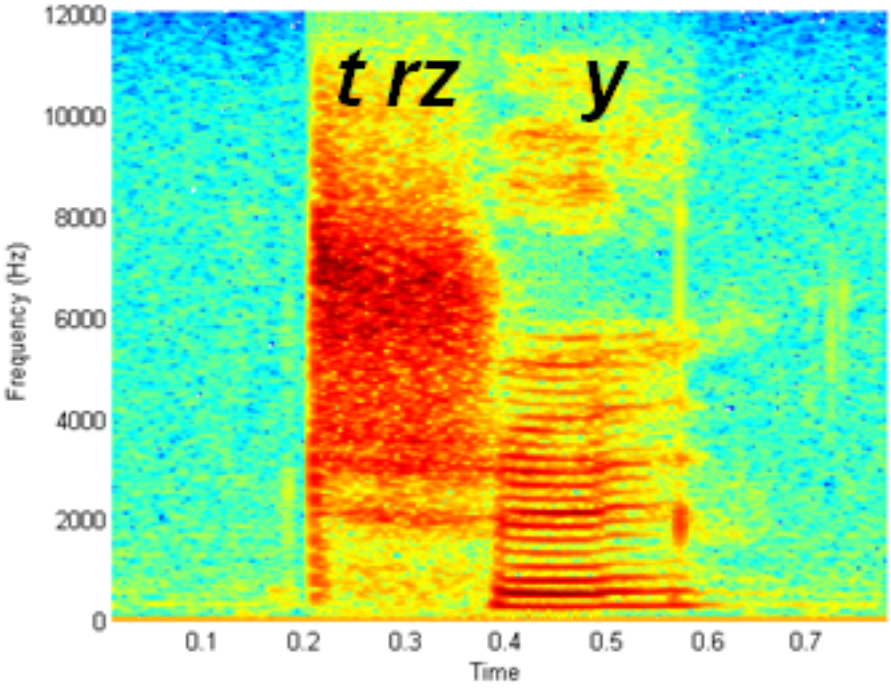
$$w_{\tau}^{t} = \begin{cases} 1 & dla\, t = \{\tau{+}0, \tau{+}1, ..., \tau{+}M{-}1\}, \\ 0 & w\, przeciwnym\, razie. \end{cases}$$

Wprowadza ono jednak znaczne zniekształcenia widma analizowanego sygnału, które są związane z efektami brzegowymi wyciętego fragmentu. [18]To zjawisko możemy minimalizować stosując inny typ okna, np. okno Hamminga, które ma węższe widmo:

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

Niestety okno Hamminga posiada również pewne negatywne właściwości, mianowicie poszerza wszystkie pasma filtrów analizujących widmo co wiąże się pogorszeniem rozdzielczości częstotliwościowej prowadzonej analizy. [22]

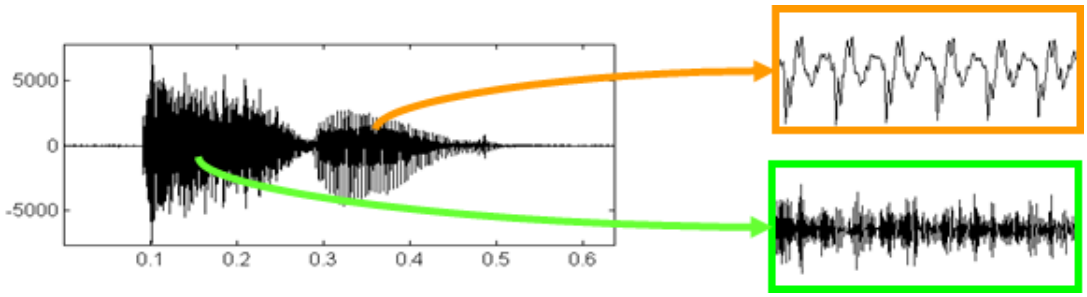
Po dokonaniu transformaty sygnał mowy może zostać przedstawiony na spektrogramie (rys. 3.3).



Rysunek 3.3: Reprezentacja słowa „trzy” w dziedzinie częstotliwości

Patrząc na spektrogram (rys. 3.3) możemy zauważyć różnice pomiędzy głoskami bezdźwięcznymi a głoską dźwięczną „y”. W przypadku głosek dźwięcznych możemy wyróżnić pewną okresowość, natomiast głoski bezdźwięczne często mieszają się z szumem (rys. 3.4). Co za tym idzie, słowa różniące się tylko częścią bezdźwięczną trudno rozpoznać.

Typowym formatem zapisu sygnału analogowego w postaci cyfrowej jest WAV (ang. wave form audio format), został on opracowany przez Microsoft oraz IBM w roku 1991. Jest to format bezstratny. Opis struktury pliku WAV można znaleść w wielu źródłach, np. na stronie [7].



Rysunek 3.4: Segment dźwięczny i bezdźwięczny słowa „trzy”

3.3. Detekcja sygnału mowy

W każdym systemie analizy mowy możemy wyróżnić początkowe etapy analizy tzn. przetwarzanie wstępne do którego mogą należeć następujące kroki:

- preemfaza, stosuje się ją w celu wzmocnienia wyższych częstostliwości w sygnale mowy osłabionych na skutek filtracji przez kanał transmisyjny,
- usuwanie zakłóceń impulsowych z sygnału za pomocą fitru medianowego,
- normowanie parametrów zależnych od mówcy,
- odszumianie sygnału mowy (metoda odejmowania widmowego (ang. spectral substraction), filtr Kalmana, filtr Wienera, metody perceptualne),
- inne rodzaje fitracji. [19]

W widmie mowy więcej energii znajduje się w niskich częstotliwościach, to niekorzystne z punktu widzenia przetwarzania sygnałów zjawisko nazywane jest przekrzywieniem widma (ang. spectral tilt). Aby usunąć skutki przekrzywienia stosuje się w/w preemfazę według następującego wzoru:

$$y(n) = x(n) - a * x(n-1),$$

gdzie a to współczynnik preemfazy $0.9 \leq a \leq 1$. [22]

Detekcję sygnału mowy wykonuje się w celu podniesienia skuteczności rozpoznawania oraz skrócenia czasu obliczeń. Dokonuje się jej po uprzednim podziale sygnału mowy na ramki. Można ją przeprowadzić w oparciu o kilka parametrów m.in.:

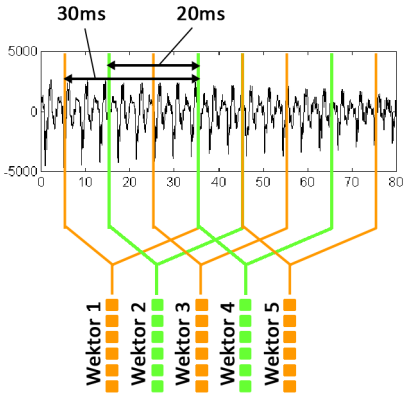
- moc sygnału w ramce,
- liczba przejść przez zero w ramce,
- entropia widma.

W każdej ramce obliczany jest wybrany parametr a następnie na podstawie jego wartości podejmowana jest decyzja o zaakceptowaniu ramki jako sygnału mowy lub jej odrzuceniu.

Możemy też przyjąć zasadę, że po wstępnej detekcji sygnału mowy odrzuca się fragmenty sygnału krótsze niż 100 ms, chyba że leżą w odległości czasowej mniejszej niż 100 ms od innych fragmentów sygnału. [19]

3.4. Ekstrakcja cech sygnału mowy

W wyniku ekstrakcji cech otrzymujemy wartości parametrów zawierających informację o treści wypowiedzi i będących niezależnymi od indywidualnych cech głosu mówcy. Parametry te tworzą wektory cech, na podstawie których dokonuje się klasyfikacji sygnału. Przykładowy podział na ramki pokazano na rysunku 3.5, wektory 6 parametrów ekstrahowanych w ramkach o czasie trwania 30 ms, zachodzących na siebie na odcinkach 20 ms.



Rysunek 3.5: Ekstrakcja cech

Najczęściej stosowane cechy sygnału mowy:

– bank filtrów,

– współczynniki LPC (ang. Linear Prediction Coefficients),

– parametry cepstralne,

– parametry mel-cepstralne (MFCC ang. Mel-frequency cepstral coefficients),

– parametry mel-spektralne (MFC ang. Mel-frequency cepstrum),

– parametry dyskretnej transformacji falkowej (ang. DWT - Discrete Wavelet Transform). [19]

Każdy zestaw cech może być uzupełniony o tzn. cechy dynamiczne, czyli współczynniki różnicowe. Polega to na obliczeniu pierwszej i drugiej pochodnej powyższych współczynników (tzw. współczynników delta oraz delta-delta) względem kilku ramek.

Pakiet HTK może korzystać zarówno z cech mel-cepstralnych jak i LPC. Opisane zostaną jedynie cechy mel-cepstralne oraz mel-spektralne ponieważ są ze sobą powiązane. W systemie stworzonym na potrzeby tej pracy zostały użyte cechy mel-cepstralne.

Ucho człowieka reaguje nieliniowo na zmieniającą się częstotliwość dźwięku. Częstotliwości powyżej 1 kHz są słabiej odczuwane niż różnice w zakresie niskich częstotliwości. Dlatego im wyższa częstotliwość tym są potrzebne coraz większe odstępy między kolejnymi pasmami dla zrekompensowania nieliniowości. Po to właśnie wprowadzono skalę melową (Mel) zamiast hercowej (Hz). [22]

$$\omega_{Mel} = 2595log(1 + \frac{\omega}{700Hz})$$

Widma ramek sygnału po uprzednim przeprowadzeniu DFT poddawane są filtracji za pomocą melowego banku filtrów pasmowo przepustowych. W systemach rozpoznawania mowy zazwyczaj stosuje się banki 26 filtrów melowych. Filtry tworzone są dla kolejnych pasm częstotliwości, rozmieszczonych w nieliniowy sposób wyznaczony przez skalę Mel. Zdefiniowane są w dziedzinie częstotliwości co umożliwia łatwe wymnożenie ich przez przekształcony sygnał. Do wyznaczenia cech mel-spektralnych dla każdej ramki sygnału wykorzyje się zbiór l trójkątnych filtrów D(l, k).

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

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

Chcąc wyliczyć spektrum energii dla każdej ramki, stosujemy FFT dla każdego splotu sygnału z kolejnym oknem i obliczamy kwadrat amplitudy każdego współczynnika zespolonego Fouriera.

$$FC(k, \tau) = \mid \sum_{t=0}^{M-1}[x(\tau + t)e^{-i2\pi kt/M} * w_\tau(t)] \mid^2 \text{ dla } k = 0, ..., M-1$$

Współczynniki mel-cepstralne możemy wyznaczyć ze wzoru:

$$MFCC(k, \tau) = \sum_{l=0}^{L-1}[logMFC(l, \tau) * cos(\frac{k*(2l+1)\pi}{2L})], k = 1, ..., K$$

Natężenie dźwięku jest odczuwane przez ludzi w skali logarytmicznej dlatego przy obliczaniu cepstrum sygnał otrzymany po przejściu przez bank filtrów melowych jest logarytmowany.

„Ponieważ układ głosu ma charakter ciągły, zatem poziomy energii w sąsiednich pasmach są skorelowane. Dlatego też niebędna do tego transformata odwrotna Fouriera (tu wystarczy przekształcenie cosinusowe) zamienia zbiór logarytmów energii na nieskolerowane ze sobą współczynniki cepstralne." [22]

Aby usunąć szkodliwy wpływ podstawowych drgań krtaniowych na zestaw cech możemy zastosować przekształcenie zwane liftrowaniem.

$$c_n^{lift} = (1 + \frac{\lambda}{2}sin(\frac{\pi n}{\lambda}))c_n \text{ dla } n = 1, ..., \lambda,$$

gdzie c_n to n-ty współczynnik MFCC a stała λ odpowiada indeksowi cechy związanej z częstotliwością podstawową.

Podstawowy wektor cech możemy uzupełnić o tzn. cechę „energetyczną", która pomaga odróżnić sygnał ciszy od sygnału mowy a także słabe bezdźwięczne spółgłoski od silnych dźwięcznych samogłosek. Sumaryczną energię w ramce sygnału τ obliczamy sumując kwadraty próbek w dziedzinie czasu według nastepującego wzoru:

$$E^{(\tau)} = \sum_{i=1}^{M} f_i^2 \qquad [14, 22]$$

Podsumowując, wyznaczanie współczynników MFCC możemy przestawić w następujących krokach:

– blokowanie sygnału w ramki, okienkowanie oknem Hamminga,

– przeprowadzenie FFT na zokienkowanych ramkach sygnału,

– filtracja za pomocą melowego banku filtrów,

– obliczenie mocy FFT w określonych pasmach częstotliwościowych,

– obliczenie logarytmu zakumulowanych współczynników widmowych,

– przeprowadzenie DCT na zlogarytmowanych współczynnikach widmowych,

– opcjonalnie, wyznaczenie cech dynamicznych.

DCT (ang. discrete cosine transform) przeprowadzamy według następującego wzoru:

$$x(n) = c(n) \sum_{k=0}^{K-1} ln(S_k) cos(\tfrac{\pi(2k+1)n}{2K}),$$

$$\text{gdzie } c(0) = \sqrt{\tfrac{1}{K}}, \text{dla } n > 0 \; c(n) = \sqrt{\tfrac{2}{K}} \text{ [18]}$$

Otrzymany wektor cech MFCC zawiera liczbę elementów równą liczbie pasm melowych. Do dalszego przetwarzania bierzemy zazwyczaj pierwsze 12 współczynników, do których dokładamy energię sygnału w ramce oraz jeśli chcemy uwzględnić dynamikę zmian współczynników w czasie (wielkość zmian oraz ich tempo), poszerzamy nasz wektor o ich przyrosty kolejno pierwszego i drugiego rzędu (czyli współczynniki delta oraz delta-delta). W wyniku tych operacji otrzymujemy 39-elementowy wektor cech mel-cepstralnych.

4. Ukryte Modele Markowa

4.1. Wprowadzenie teoretyczne

„Ukryte Modele Markowa stanowią serce większości współczesnych systemów rozpoznawania mowy." [13] Sprawdzają się również w wielu innych zastosowaniach, najczęściej tam gdzie mamy doczynienia z sygnałem niederministycznym, np.:

– rozpoznawanie obrazów,

– modelowanie DNA,

– modelowanie danych ekonomicznych.

Pierwsze prace na temat Ukrytych (niejawnych) Modeli Markowa prowadził Baum wraz ze współpracownikami w latach 60-tych oraz 70-tych. Dotyczyły one metody estymacji parametrów modeli HMM nazwanej algorytmem Bauma-Welcha. [12] Do celów rozpoznawania mowy zostały użyte poraz pierwszy w połowie lat siedemdziesiątych przez pracowników firmy IBM. W obecnych czasach jest to najczęściej wykorzystywane narzędzie klasyfikacji w systemach rozpoznawania mowy. [13] Opisane zostało w wielu pozycjach literaturowych, m.in. w [20], tutaj zostanie przedstawiony jedynie ogólny zarys tej metody.

W HMM wykorzystywane są modele statystyczne posiadające własności Markowa pierwszego rzędu co oznacza, że aktualny stan modelu zależy tylko od stanu poprzedniego. W ukrytym modelu Markowa zakłada się, że stany są niewidoczne dla obserwatora, natomiast wyjście (wartości obserwowane) jest jawne, stąd w nazwie występuje słowo 'ukryte'. [22] W przypadku mowy, wartości obserwowane bądź też obserwacje są to wektory cech wyekstrahowane z pojedynczej ramki sygnału mowy. Ukryty model Markowa opisuje pewien układ (proces stochastyczny), który w danym momencie może znajdować się tylko w jednym ze stanów.

Ogólne założenia metody HMM z punktu widzenia rozpoznawania mowy można przedstawić następująco:

– dla każdej klasy (np. głoski, fonemu lub słowa) tworzy się pewien model stochastyczny Λ_m, m=1, ... , M, gdzie M to liczba klas,

– dla każdej nieznanej sekwencji stanów X oblicza się prawdopodobieństwa generowania sekwencji wektorów p(O, X|Λ_m) przez modele dla poszczególnych klas,

– klasyfikacja odbywa się na zasadzie największego prawdopodobieństwa, X jest przypisywane do takiej klasy Λ_m, która posiada największe prawdopodobieństwo p(O, X|Λ_m). [19]

4.2. Parametry modeli Markowa

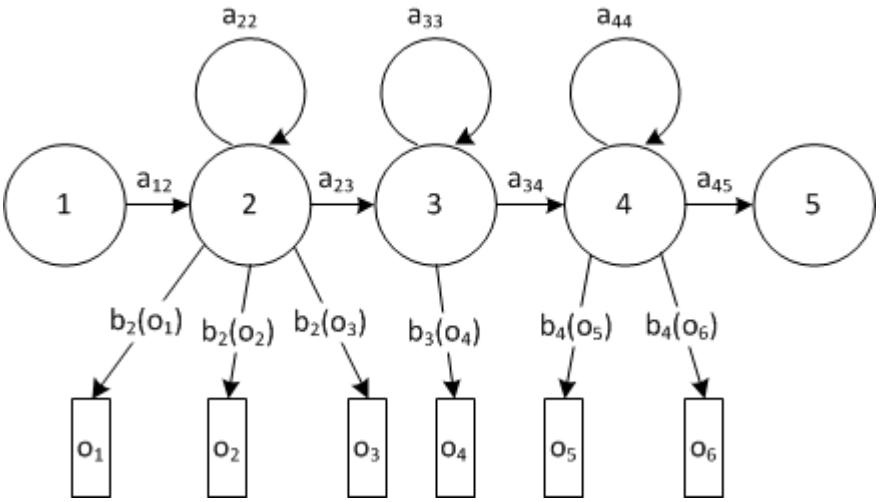
Macierz przejść definiuje możliwe przejścia pomiędzy stanami oraz wartości ich prawdopodobieństw:

$$\begin{bmatrix} 0 & a_{12} & 0 & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 & 0 \\ 0 & 0 & a_{33} & a_{34} & 0 \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Dla każdego przejścia z chwili czasowej t do t+1 przejście ze stanu x_i do stanu x_j następuje z prawdopodobieństwem a_{ij}.

Suma prawdopodobieństw w wierszach a więc suma wszystkich wyjść ze stanu zawsze wynosi jeden.

Na rysunku 4.1 widoczny jest 5-cio stanowy model markowa typu left-to-right z przejściami zapisanymi w powyższej macierzy.



Rysunek 4.1: Model Markowa 5-cio stanowy typu left-to-right

W rozpoznawaniu mowy zazwyczaj wykorzystywane są modele typu left-to-right (wyjątek to np. model ciszy) oznacza to, że z aktualnego stanu możliwe jest przejście do stanu kolejnego, pozostanie w tym stanie, lub przejście o kilka stanów do przodu, nie można się cofać.

Stan początkowy oraz końcowy czyli w przypadku 5-cio stanowego modelu stan 1 oraz stan 5 zwane są stanami nieemitującymi ponieważ nie generują żadnych obserwacji, mają za zadanie jedynie rozpocząć oraz zakończyć proces.

Dla każdej ramki czasowej t, może być przez każdy stan s_ϑ modelu Λ_m generowany wektor obserwacji o_t. Suma prawdopodobieństw generowania obserwacji dla jednego stanu zawsze wynosi jeden.

Dla modelu na rysunku 4.1 do wygenerowania wszystkich wektorów obserwacji, w kolejności od o_1 do o_6 potrzebna jest sekwencja stanów $X = 1, 2, 2, 2, 3, 4, 4, 5$.

Obserwacje (wektory cech) o_t emitowane są z prawdopodobieństwem b_j określanym zazwyczaj jako wielowymiarowy rozkład Gaussa:

$$b_j(o_t) = \frac{1}{\sqrt{(2\pi)^n \left|\sum_j\right|}} e^{-\frac{1}{2}(o_t-\mu_j)' \sum_j^{-1}(o_t-\mu_j)}$$

Gdzie:

$\sum_j$- macierz kowariancji,

μ_j- wektor średni obserwacji,

n - wymiar wektora obserwacji.

Prawdopodobieństwo obserwacji możemy również aproskymować za pomocą mieszaniny wielowymiarowych rozkładów Gaussa:

$$p_j(o_t) = \sum_{k=0}^{K-1} c_{jk} b_{jk}(o_t)$$

$$\sum_{k=0}^{K-1} c_{jk} = 1$$

Gdzie:

K - liczba rozkładów Gaussa,

c_{jk} - k-ty współczynnik wagowy mieszaniny dla stanu j,

b_{jk} - k-te prawdopodobieństwo obserwacji dla stanu j.

Duża liczba parametrów w przypadku mieszanin o większej liczbie rozkładów wymusza użycie dużej ilości danych treningowych. Można jednak uprościć model redukując liczbę jego parametrów np. poprzez:

– wiązanie parametrów, podczas którego stosuje się wspólną macierz kowariancji dla tych rozkładów dla których wartości kowariancji są zbliżone,

– zastosowanie diagonalnej macierzy kowariancji, gdzie zakłada się, że elementy wektora cech są wzajemnie nieskorelowane, dlatego dopuszczalne jest zastosowanie macierzy posiadającej elementy tylko na diagonali, powstają przez to błędy aproksymacji, które można minimalizować przez zastosowanie większej liczby rozkładów.

Prawdopodobieństwo $P(O,\ X|\Lambda_m)$ (gdzie sekwencja wektorów X oznacza sekwencję obserwacji o) wygenerowania przykładowej sekwencji wektorów obserwacji o_1,o_2,o_4,o_6 nawiązując do modelu przedstawionego na rysunku 4.1 oblicza się następująco:

$$P(O,\ X|\Lambda_m) = a_{12}b_2(o_1)a_{22}b_2(o_2)a_{23}b_3(o_4)a_{34}b_4(o_6)a_{45} \quad [19]$$

W praktyce jedynie sekwencja obserwacji O jest znana, natomiast sekwencja stanów X jest ukryta. Wymagane prawdopodobieństwo jest wyliczane przez sumę wszystkich możliwych sekwencji stanów X = x(1), x(2), x(3), ..., x(T) daną wzorem:

$$P(O|\Lambda_m) = \sum_X a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t)a_{x(t)x(t+1)}$$

Jako alternatywę dla powyższego wzoru możemy rozważać w przybliżeniu najbardziej prawdopodobną sekwencję stanów daną wzorem:

$$P(O|\Lambda_m) = \max_X \left\{ a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t)a_{x(t)x(t+1)} \right\}$$

Gdzie:

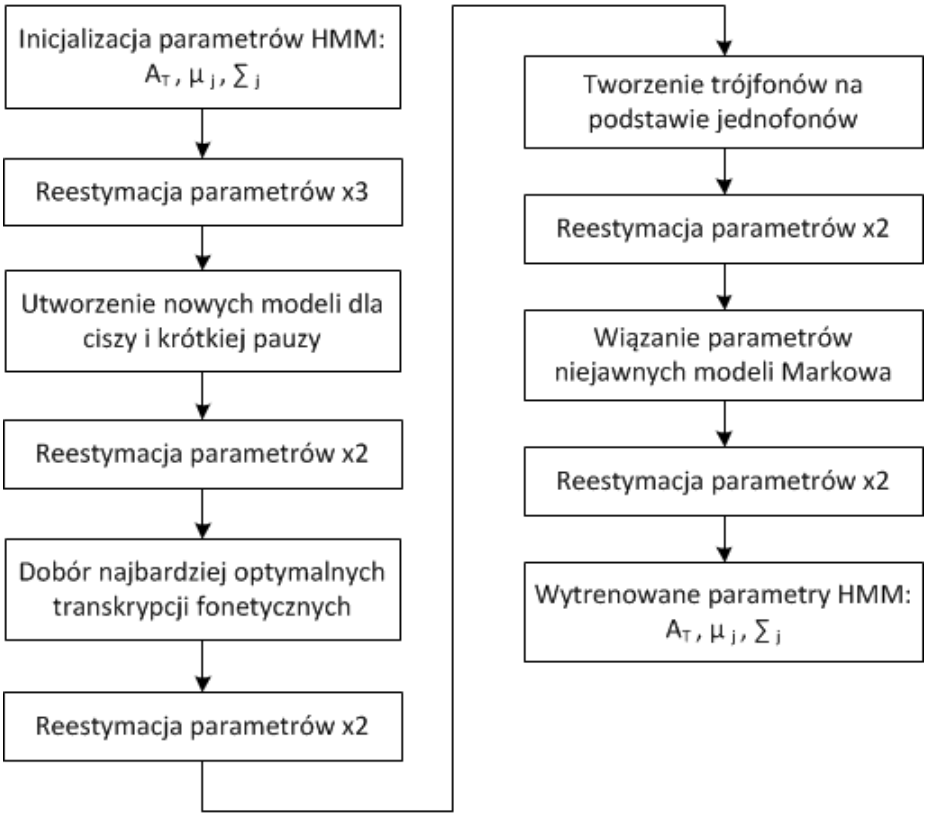
$x(0)$ i $x(T+1)$ to stan pierwszy i ostatni modelu czyli stany nieemitujące.

Zakłada się, że parametry a_{ij} oraz $b_j(o_t)$ są znane dla każdego modelu. „Tutaj właśnie leży elegancja i skuteczność metody HMM”[21]

4.3. Trenowanie

4.3.1. Przebieg

Schemat blokowy fazy trenowania modeli Markowa widoczny jest na rysunku 4.2.



Rysunek 4.2: Trenowanie modeli Markowa [19]

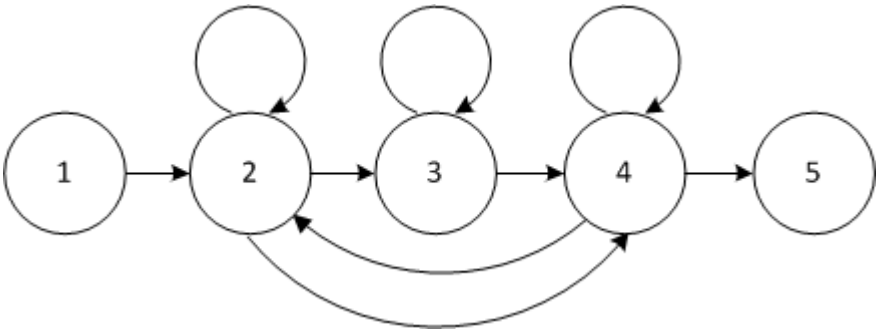
Gdzie:

$\sum_j$- macierz kowariancji,

μ_j- wektor średni obserwacji,

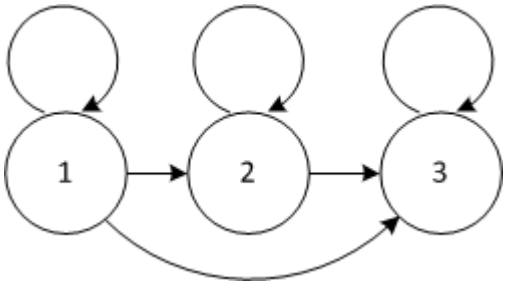
A_T- macierz przejść modelu.

Jednofonowe modele Markowa dla ciszy i krótkich pauz są przedstawione kolejno na rysunku 4.3 oraz 4.4.



Rysunek 4.3: Model HMM dla ciszy

W modelu ciszy mamy dodatkowe przejście ze stanu 2 do stanu 4 oraz ze stanu 4 do stanu 2.



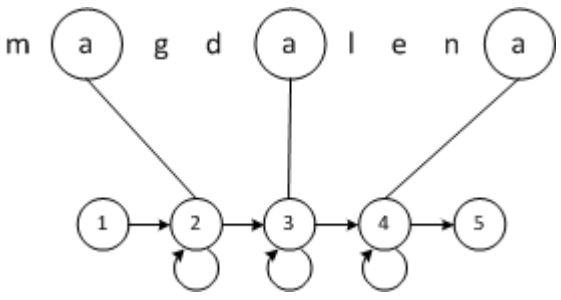
Rysunek 4.4: Model HMM dla krótkiej pauzy

W modelu krótkiej pauzy mamy dodatkowe przejście pomiędzy stanami 1 oraz 3 czyli stanami nieemitującymi.

Stan 3 z modelu ciszy oraz stan 2 z modelu krótkiej pauzy są takie same. Model krótkiej pauzy tworzymy na podstawie modelu ciszy, usuwając stany 2 oraz 4, powstaje wówczas model trzy stanowy, gdzie poprzedni stan 3 staje się stanem 2 modelu.

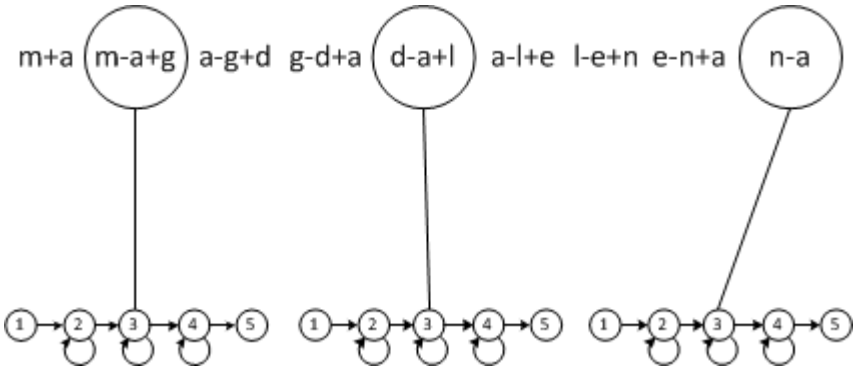
W przypadku gdy najmniejszą niepodzielną jednostką fonetyczną systemu są fonemy musimy rozpatrywać modele nie tylko dla jednofonów lecz również dla trójfonów, czyli możliwych kombinacji fonemów w otoczeniu innych fonemów.

Zapis jednofonowy pokazany jest na rysunku 4.5.



Rysunek 4.5: Zapis jednofonowy

Natomiast zapis trójfonowy na rysunku 4.6.



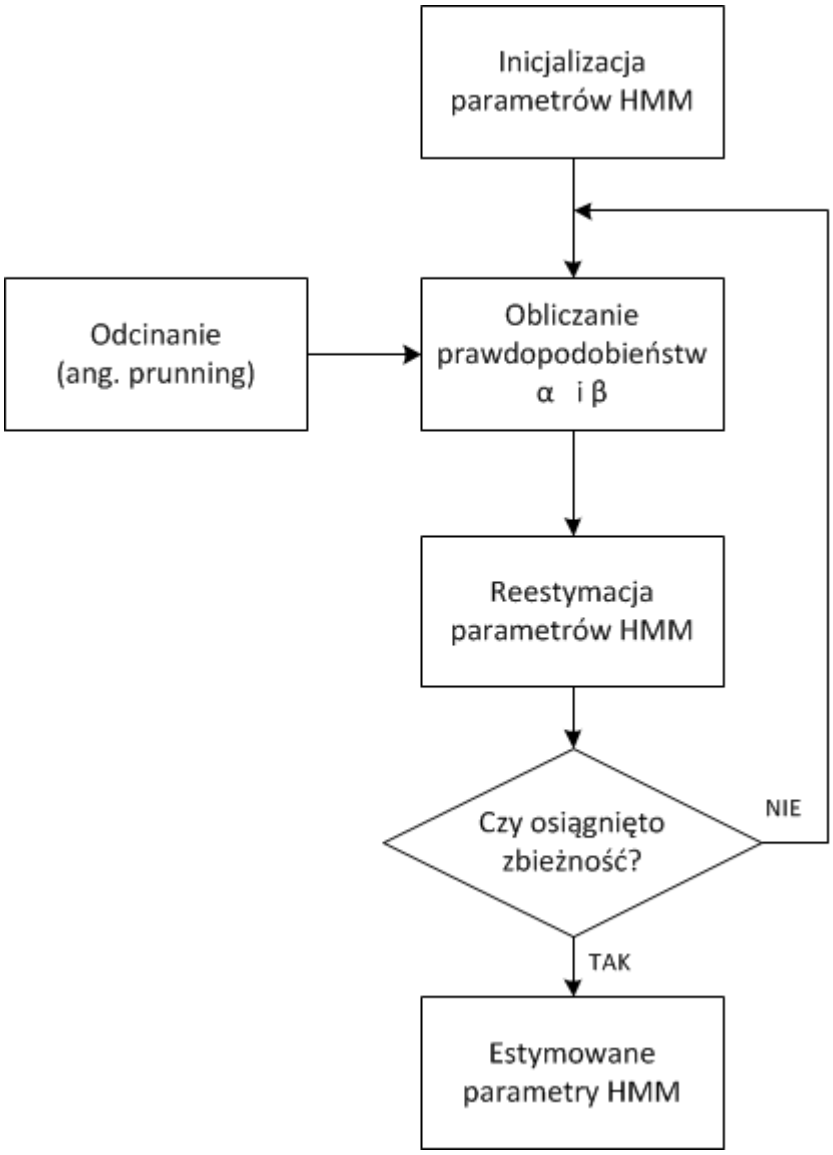
Rysunek 4.6: Zapis trójfonowy

Znak ’-’ oznacza fonem poprzedzający, natomiast znak ’+’ fonem następujący.

Modele HMM są trenowane dla każdego słowa znajdującego się w słowniku systemu poprzez próbki dźwięku nagrane dla tego słowa.

4.3.2. Algorytm Bauma-Welcha

Algorytm ten jest oparty na kryterium ML (ang. Maximum Likelihood) - maksymalizacji prawdopodobieństwa wygenerowania sekwencji obserwacji przez model HMM. Wykorzystywany jest w procesie trenowania HMM. Jest to procedura iteracyjna w wyniku której możemy otrzymać lepsze estymaty parametrów HMM. Schemat blokowy przedstawiony jest na rysunku 4.7.



Rysunek 4.7: Reestymacja Bauma-Welcha [19]

Początkowe parametry HMM określane są na zasadzie zgadywania, dokładniejsze (w sensie największego pradopodobieństwa) parametry mogą być znalezione dopiero podczas reestymacji. Estymaty parametrów μ_j oraz $\sum_j$ wyznaczane są ze wzorów:

$$\hat{\mu_j} = \frac{\sum_{t=1}^{T} L_j(t) o_t}{\sum_{t=1}^{T} L_j(t)}$$

$$\hat{\sum_j} = \frac{\sum_{t=1}^{T} L_j(t)(o_t - \mu_j)'}{\sum_{t=1}^{T} L_j(t)}$$

Gdzie:

$L_j(t)$ - oznacza prawdopodobieństwo bycia w stanie j w czasie t.

Podobne, lecz trochę bardziej skomplikowane są wzory na estymaty prawdopobieństw przejść pomiędzy stanami, szczegóły znajdują się w rozdziale 8 pozycji [21].

Do obliczenia prawdopodobieństwa $L_j(t)$ wykorzystuje się algorytm *Forward-Backward.*

Prawdodobieństwo *forward* dla modelu M zawierającemu N stanów jest zdefiniowane jako:

$$\alpha_j(t) = P(o_1, ..., o_t, x(t) = j \mid M)$$

Jest to połączone prawdopodobieństwo obserwacji pierwszego t wektora cech i bycia w stanie j w czasie t. Obliczone jest za pomocą rekursji danej wzorem:

$$\alpha_j(t) = \left[\sum_{i=2}^{N-1} \alpha_j(t-1)a_{ij}\right] b_j(o_t)$$

Ta rekursja polega na tym, że prawdopodobieństwo bycia w stanie j w czasie t oraz emitowania obserwacji można wywnioskować poprzez zsumowanie prawdopodobieństw *forward* dla wszystkich możliwych poprzedników stanów determinowanych przez prawdopodobieństwa przejść a_{ij}. Stany 1 oraz N są stanami nieemitującymi. Początkowe warunki powyższej rekursji są następujące:

$$\alpha_1(1) = 1$$

$$\alpha_j(1) = a_{1j}b_j(o_1)$$

dla $1 < j < N$.

Zatem finalne waruki:

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T)a_{iN}$$

Trzeba zauważyć, że z definicji $\alpha_j(t)$,

$$P(O \mid M) = \alpha_N(T)$$

Stąd obliczenie prawdopodobieństwa *forward* również daje prawdopodobieństwo całkowite $P(O \mid M)$.

Z kolei prawdopodobieństwo *backward* jest zdefiniowane jako:

$$\beta_j(t) = P(o_{t+1}, ..., o_T, x(t) = j \mid M)$$

Jak w przypadku prawdopodobieństwa *forward,* prawdopodobieństwo *backward* również liczone jest za pomocą rekursji:

$$\beta_j(t) = \sum_{j=2}^{N-1} a_{1j} b_j(o_{t+1}) \beta_j(t+1)$$

z warunkami początkowymi:

$$\beta_j(T) = a_{iN}$$

dla $1 < i < N$ i finalne warunki:

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(o_1) \beta_j(1)$$

Trzeba zauważyć, że w powyższych definicjach prawdopodobieństwo *forward* jest prawdopodobieństwem połączonym podczas, gdy prawdopodobieństwo *backward* jest prawdopodobieństwem warunkowym.

Prawdopodobieństwo przebywania w danych stanie jest wyznaczane przez iloczyn dwóch prawdopodobieństw zgodnie ze wzorem:

$$\alpha_j(t)\beta_j(t) = P(O, x(t) = j \mid M)$$

Stąd:

$$L_j(t) = P(x(t) = j \mid O, M) = \frac{P(O, x(t) = j \mid M)}{P(O|M)} = \frac{1}{P}\alpha_j(t)\beta_j(t)$$

Gdzie:
$P = P(O|M)$.

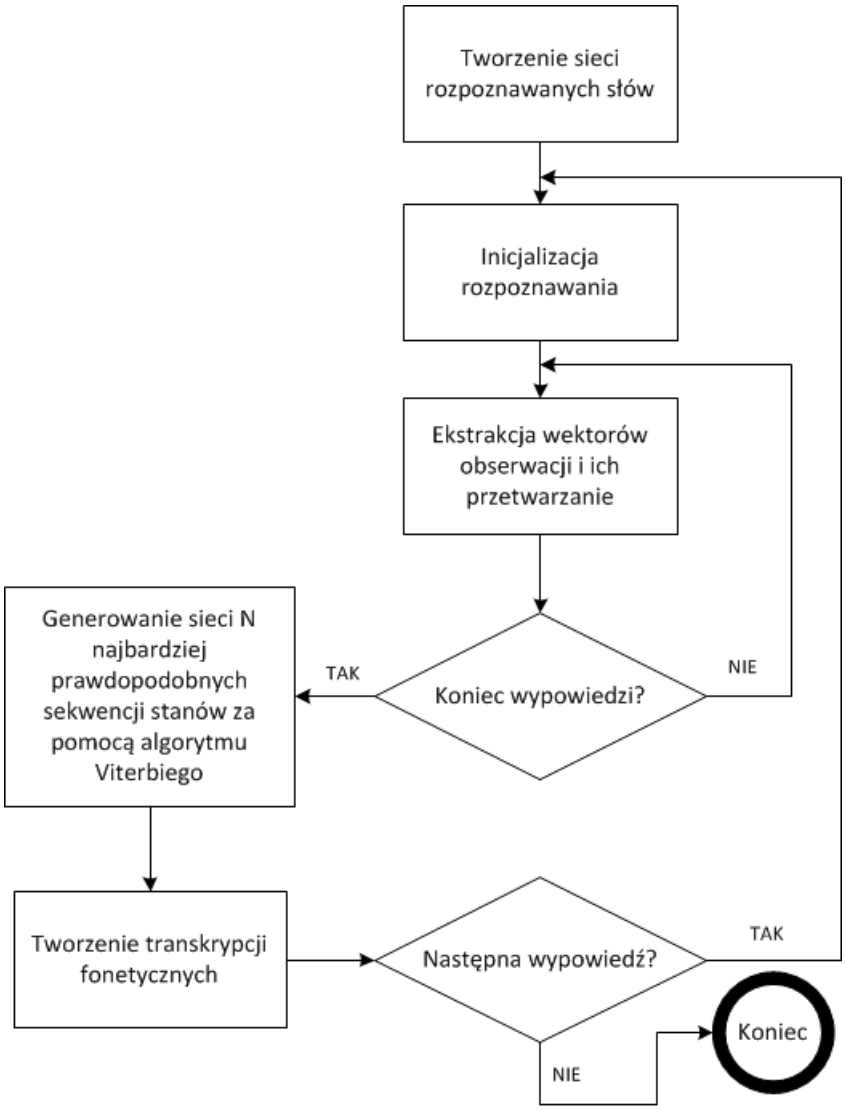
Zakłada się, że parametry dla HMM są reestymowane dla pojedyńczej sekwencji obserwacji, co odpowiada pojedyńczej próbce mowy nagranej do wytrenowania modelu. Do uzyskania dobrych estymat parametrów potrzebne jest wiele próbek tego samego słowa. [21]

Podsumowując, problem estymacji parametrów polega na doborze jak największych prawdopodobieństw przejść pomiędzy stanami oraz prawdopodobieństw wystąpienia obserwacji. Najpierw ustalamy początkowe przybliżone wartości tych parametrów a następnie przebiega reestymacja parametrów, czyli próba znalezienia ich dokładniejszych wartości. [12]

4.4. Rozpoznawanie

4.4.1. Przebieg

Fazę rozpoznawania przedstawia schemat blokowy widoczny na rysunku 4.8.

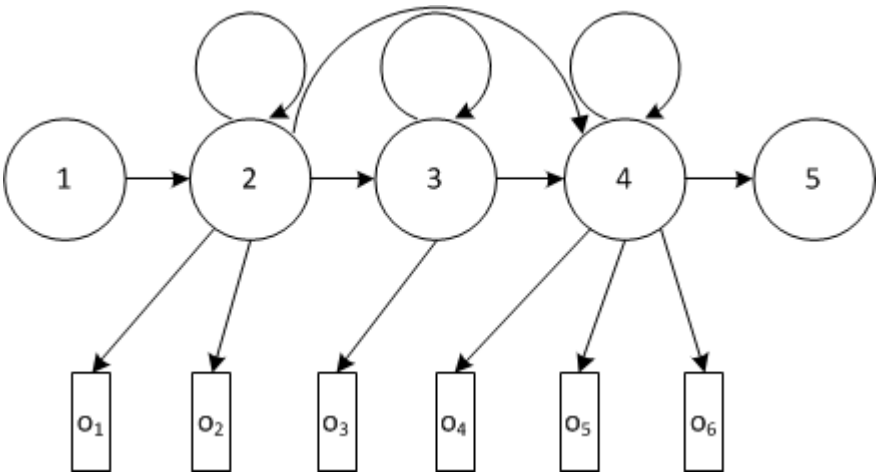


Rysunek 4.8: Schemat blokowy rozpoznawania [19]

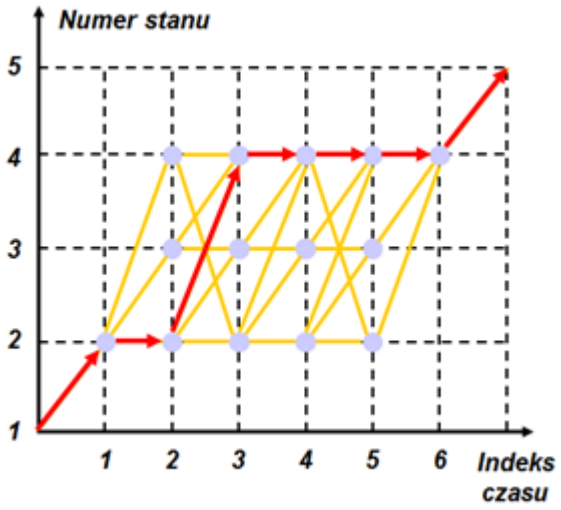
Najważniejszym etapem tej fazy jest wyznaczenie sekwencji najbardziej prawdopodobnych stanów a więc optymalnej ścieżki w modelu. Do rozwiązania tego problemu stosowany jest zazwyczaj algorytm Viterbiego.

4.4.2. Algorytm Viterbiego

Na rysunku 4.10 przedstawiony jest przykład użycia algorytmu Viterbiego dla modelu z rysunku 4.9.



Rysunek 4.9: Przykładowy model HMM



Rysunek 4.10: Algorytm Viterbiego [19]

W każdym kolejnym kroku algorytmu zapamiętywany jest argument maksymalny, wynikiem działania algorytmu jest ścieżka dająca największe prawdopodobieństwo czyli najbardziej prawdopodobna sekwencja jednostek fonetycznych które zostały przyjęte. [12]

Od strony matematycznej prawdopodobieństwo to, jest wyliczane podobnie jak prawdopodobieństwo *forward* za pomocą rekursji:

$$\phi_j(t) = \max_t\{\phi_i(t-1)a_{ij}\}b_j(o_t)$$

$$\phi_1(1) = 1$$

$$\phi_j(1) = a_{1j} b_j(o_1)$$

dla $1 < j < N$.

Największe prawdopodobieństwo $\hat{P} = (O|M)$ dane jest wzorem:

$$\phi_N(T) = \max_i \{\phi_i(T) a_{iN}\}$$

Jak w przypadku reestymacji tutaj też jest stosowana skala logarytmiczna, stąd równanie ma postać:

$$\psi_j(t) = \max_i \{\psi_i(t-1) + log(a_{ij})\} + log(b_j(o_t))\}$$

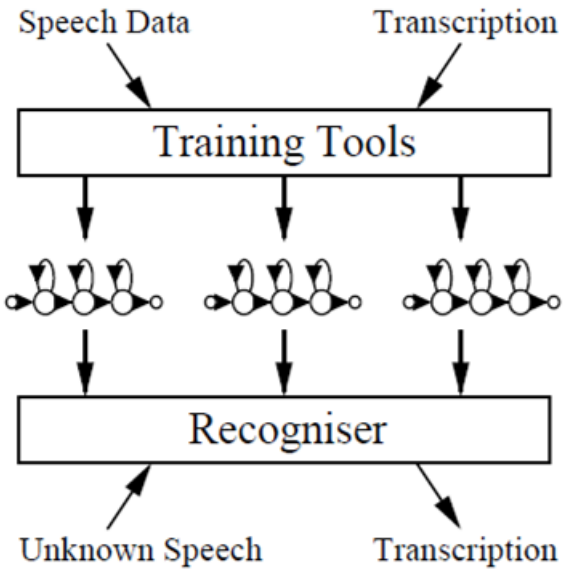
[Htkb]

5. HTK (Hidden Markov Model Toolkit)

5.1. Podstawy HTK

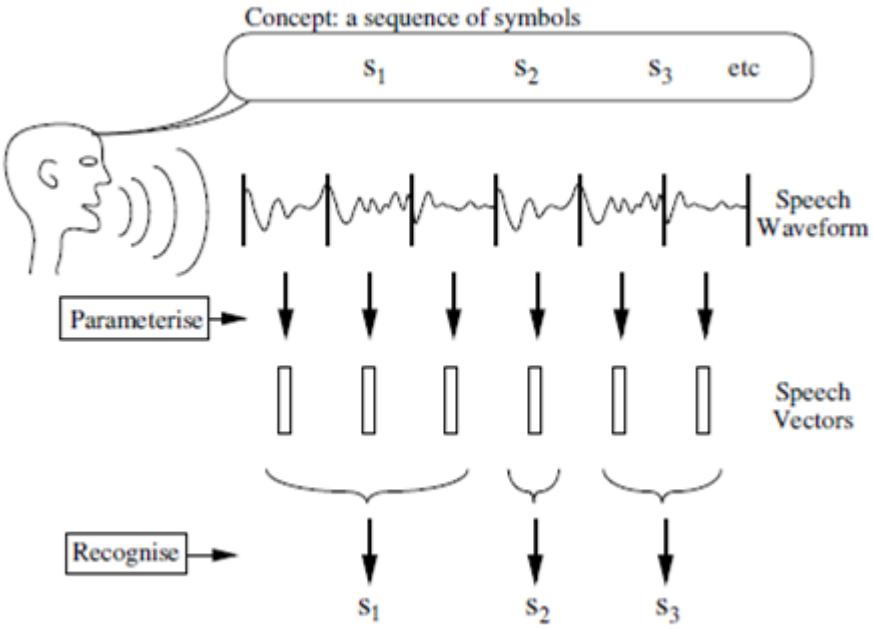
HTK jest zbiorem narzędzi napisanych w języku C i korzystających z HMM w kontekście rozpoznawania mowy. Pierwsza wersja projektu powstała w 1995 na uniwersytecie w Cambridge. Zespołem kierował profesor Steve Young. Projekt początkowo był finansowany przez Microsoft Corporation, ostatecznie jego prawa autorskie przejął Cambridge University Engineering Department. Ostatnia jego wersja o numerze 3.4 powstała w 2006 roku. Na tej właśnie wersji opiera się niniejsza praca.

HTK posiada kompleksowy zestaw bibliotek pozwalających przetwarzać sygnał mowy. Jak widać na rysunku 5.1, działanie HTK można ogólnie podzielić na dwie główne fazy. Pierwsza w której odbywa się estymacja parametrów HMM i druga podczas której przy użyciu narzędzi do rozpoznawania mowy otrzymujemy transkrypcję fonetyczną nieznanej wypowiedzi.



Rysunek 5.1: Organizacja bibliotek HTK [21]

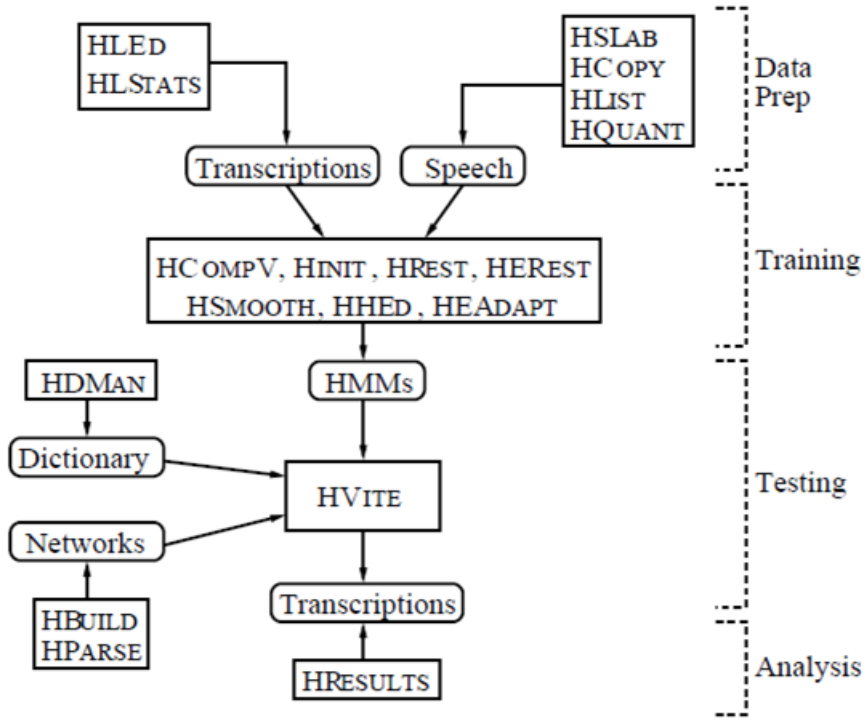
Podstawowe założenia HTK można zobrazować prostym schematem widocznym na rysunku 5.2.



Rysunek 5.2: Enkodowanie/dekodowanie wiadomości [21]

Założenie, że mowa jest sekwencją pewnych dyskretnych jednostek fonetycznych jest typowym założeniem współczesnych systemów rozpoznawania mowy o czym zostało wspomniane w rozdziale 2.

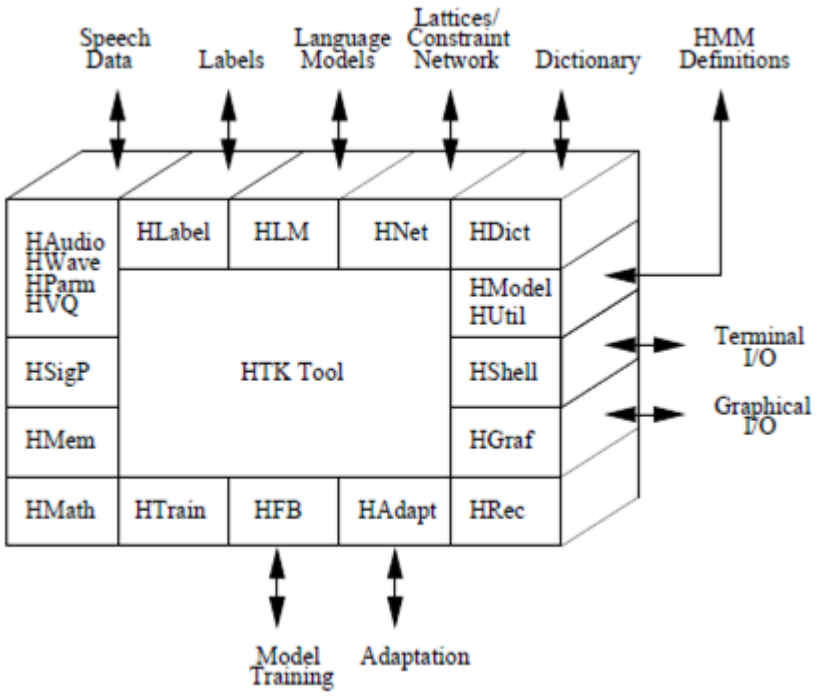
Budowa systemu rozpoznawania mowy ciągłej opartego o fonemy przy pomocy narzędzi HTK przebiega w czterech fazach (rys. 5.3). Zostały one opisane w jednym z kolejnych podrozdziałów.



Rysunek 5.3: Cztery główne fazy budowy systemu rozpoznawania mowy z HTK [21]

5.2. Architektura

Większość funkcjonalności HTK jest wbudowane w moduły, które są bibliotekami systemowymi (pliki z rozszerzeniem *.dll w systemach Windows lub *.lib dla systemów Unix). Te moduły zapewniają każdemu narzędziu interfejs do komunikowania się ze światem zewnętrznym w podobny sposób. Dostarczają również centralny zasób najczęściej używanych funkcji. Na rysunku 5.4 przedstawione są typowe narzędzia HTK i pokazane interfejsy wejś oraz wyjś.



Rysunek 5.4: Architektura HTK [21]

Nad interakcją z użytkownikiem czuwa biblioteka *HShell* oraz biblioteka zarządzająca pamięcią *HMem*. Wsparcie matematyczne zapewnia *HMath*, a operacje związane z przetwarzaniem sygnałów dźwiękowych są kontrolowane przez bibliotekę *HSigP.* Dla każdego interfejsu są przeznaczone odpowiednie typy plików. *HLabel* dostarcza pliki etykiet, *HLM* pliki modelu językowego, *HNet* sieci i krat słów, *HDict* słownika, *HModel* definicji HMM. Wszystkie wejściowe oraz wyjściowe pliki dźwiękowe w formacie WAVE kontroluje bilbioteka *HWave,* parametry tych plików *HParm.* Bezpośrednie wejście dzwięku zapewnia *HAudio*, z kolei *HGraf* rysuje wykresy. *HUtil* zawiera narzędzia manipulujące modelami HMM podczas gdy *HTrain* oraz *HFB* zapewnia wsparcie w trakcie trenowania modeli. *HRec* z kolei zawiera główne funkcje procesu rozpoznawania. Szczegóły wspomnianych jak i pozostałych narzędzi z pakietu HTK znajdują się w pozycji [21].

5.3. Parametry narzędzi

Narzędzia HTK posiadają tradycyjny konsolowy interfejs użytkownika. Każde narzędzie posiada zestaw wymaganych argumentów oraz argumenty opcjonalne. Przykład ich użycia zostanie zaprezentowany na nieistniejącym narzędziu nazwanym *HFoo.*

```
HFoo -T 1 -f 34.3 -a -s myfile file1 file2
```

To narzędzie posiada dwa główne argumenty nazwane *file1* i *file2* oraz argumenty opcjonalne. Opcje zawsze zaczynają się od pojedyńczej litery poprzedzonej znakiem minus a następnie podawana jest war-

tość opcji. Kolejne opcje rozdzielone są znakiem spacji. Czasem wartość opcji jest wartością typu zmiennoprzecinkowego (opcja *-f)* a czasem jest to wartość całkowita (opcja *-T,* która zawsze kontroluje wyjście narzędzia czyli co zostanie wydrukowane w konsoli) lub też ciąg znaków (opcja *-s*). Bywa też, że opcje są zwykłą flagą (tzw. przełącznikiem), nie potrzebują żadnej wartości, np. opcja *-a*, która odblokowuje lub zablokowuje pewną cechę narzędzia.

Narzędzia mogą być również parametryzowane za pomocą pliku konfiguracyjnego. Dla przykładu:

```
HFoo -C config -f 34.3 -a -s myfile file1 file2
```

Wywołanie powyższego polecenia ustawia parametry zapisane w pliku. Pliki konfiguracyjne mogą być czasami alternatywą dla argumentów podawanych podczas wywołania narzędzia w konsoli.

Konsolowe działanie narzędzi HTK może wydawać się trochę staromodne w porównaniu z graficznymi interfejsami użytkownika różnych współczesnych aplikacji, lecz daje przejrzysty podgląd działania poszczególnych narzędzi oraz pozwala w prosty sposób zautomatyzować wykonanie kilku narzędzi poprzez pisanie skryptów. Aby mieć podgląd wszystkich możliwych opcji wybranego narzędzia wystarczy wywołać je w konsoli bez żadnych dodatkowych argumentów.

5.4. Fazy budowy systemu rozpoznawania mowy

5.4.1. Przygotowanie danych

Pierwszym krokiem jest stworzenie słownika systemu w którym zapisane są wszystkie używane słowa wraz z ich zapisem fonetycznym. Następnie trzeba ustalić gramatykę systemu, czyli format wypowiedzi które będą rozpoznawane np.:

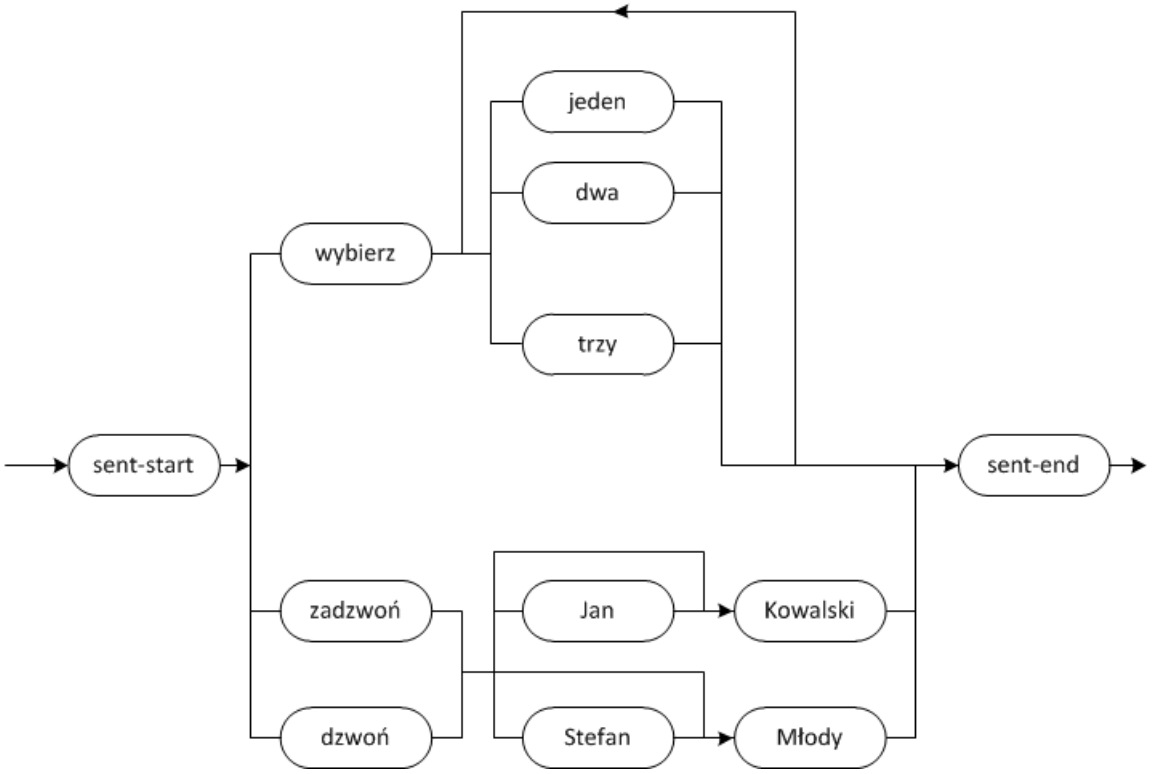
```
Wybierz jeden dwa cztery jeden pięć
Wybierz dwa zero zero osiem
Zadzwoń do Stefan Młody
Dzwoń do Kowalski
```

HTK dostarcza język definicji gramatyki. Składa się on z zestawu definicji zmiennych zawierających wyrażenia regularne będące słowami które będą rozpoznawane.

Dla powyższych wypowiedzi poniżej jest pokazany przykład definicji gramatyki:

```
$cyfry = JEDEN | DWA | TRZY | CZTERY | PIĘĆ |
SZEŚĆ | SIEDEM | OSIEM | DZIEWIĘĆ | ZERO;
$osoba = [ JAN ] KOWALSKI |
[ STEFAN ] MŁODY;
( SENT-START ( WYBIERZ <$cyfry> | (ZADZWOŃ|DZWOŃ) DO $osoba) SENT-END )
```

Znak „|” oznacza alternatywe, kwadratowe nawiasy oznaczają opcjonalność a nawiasy ostre („<” oraz „>”) występowanie powtórzeń. Kompletna gramatyka systemu może być przedstawiona jako sieć słów pokazana na rysunku 5.5.



Rysunek 5.5: Gramatyka systemu rozpoznawania mowy

Do poprawnego wytrenowania modeli HMM potrzeba wielu próbek zawierających wielokrotne nagrania wszystkich słów ze słownika oraz powiązane z nimi pliki transkrypcji. Transkrypcje te muszą być zapisane w odpowiednim formacie, dla HTK będzie to format *MLF*. Przykład dla pliku dźwiękowego *sample1.wav* zawierającego nagranie słów *jeden dwa trzy*:

```
#!MLF!#
"*/sample1.wav"
JEDEN
DWA
TRZY
.
```

Do nagrania próbek sygnału mowy może zostać użyte narzędzie *HSLab* lub inny rejestrator dźwięku.

Otrzymane nagrania trzeba zparametryzować (dokonać ekstrakcji cech). Dokonuje się tego przy użyciu narzędzia *HCopy,* które konwertuje pliki dźwiękowe do plików zawierających odpowiednie cechy sygnału mowy.

Transkrypcje plików dźwiękowych również muszą zostać odpowiednio przygotowane. Jeśli tworzy się system oparty o fonemy, narzędzie *HLEd* konwertuje pliki transkrypcji, do plików zawierających transkrypcje fonemowe, czyli zamienia zwykły alfabet języka na alfabet fonetyczny.

5.4.2. Trenowanie

Przed rozpoczęciem trenowania trzeba utworzyć prototyp modelu HMM, zapisany jako zwykły plik tekstowy, gdzie definiuje się topologię modelu (określa ilość stanów, dopuszczalne przejścia między nimi oraz kilka innych parametrów). HTK zawiera skrypty generujące gotowe prototypy.

Modele HMM są tworzone dla każdego słowa ze słownika. Procedura reestymacji przebiega równolegle dla każdego modelu w przypadku mowy ciągłej, lub indywidualnie w przypadku izolowanych słów.

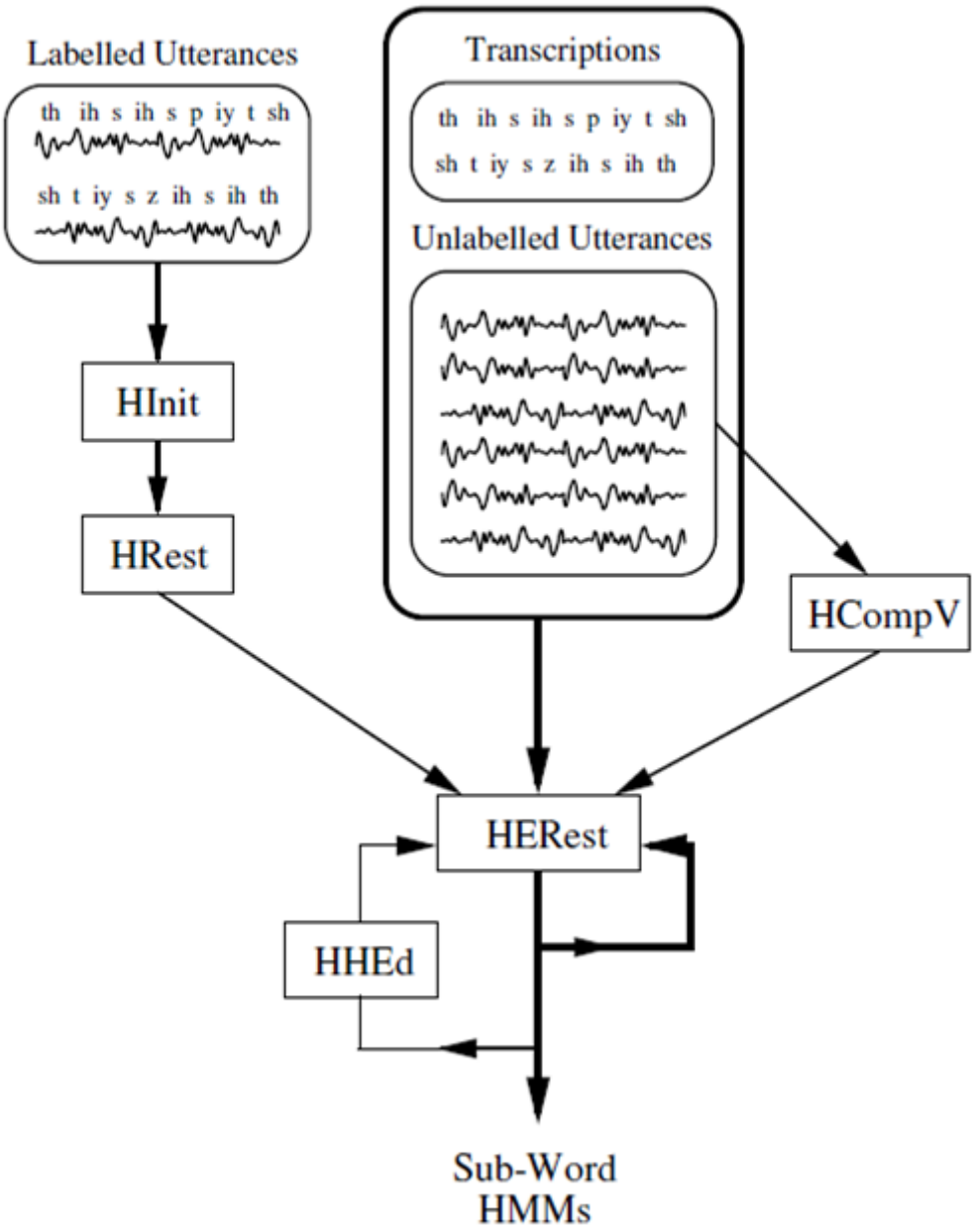
Inicjalizacja niektórych parametrów HMM jest wykonywana za pomocą narzędzia zwanego *HInit*. Narzędzie to, najpierw przydziela po równo wektory obserwacji do każdego ze stanów modelu (prócz stanów nieemitujących) a następnie wyznacza dla nich początkowe wartości parametrów za pomocą wzorów:

$$\hat{\mu_j} = \frac{1}{T}\sum_{t=1}^{T} o_t$$

$$\hat{\sum}_j = \frac{1}{T}\sum_{t=1}^{T}(o_t - \mu_j)(o_t - \mu_j)'$$

Tak dzieje się w przypadku izolowanych słów. Dla mowy ciągłej zamiast narzędzia *HInit* wykorzystywane jest narzędzie *HCompV* a początkowe parametry są jednakowe dla wszystkich modeli.

Warto wspomnieć, że podczas wyznaczania prawdopodobieństw *forward* i *backward* liczony jest iloczyn dużej liczby prawdopodobieństw. Oznacza to, że przetwarzane liczby mogą stać się bardzo małe. Stąd aby uniknąć problemów numerycznych, powyższe obliczenia przeprowadzane są przez HTK w skali logarytmicznej. HTK używa algorytmu Bauma-Welcha do reestymacji parametrów HMM. Ten algorytm jest zaimplementowany w narzędziu zwanym *HERest* oraz *HRest*. Narzędzie *HRest* wykorzystywane jest tylko w przypadku izolowanych słów. Na rysunku 5.6 pokazano jak przebiega trenowanie za pomocą narzędzi HTK.



Rysunek 5.6: Proces trenowania [21]

Narzędzie *HHEd* przygotowuje modele HMM w zależności od kontekstu wypowiedzi.

5.4.3. Testowanie

Testowanie czyli właściwie faza rozpoznawania, polega na znalezieniu najbardziej prawdopodobnej sekwencji przejść między stanami modelu HMM za pomocą algorytmu Viterbiego zaimplementowanego w narzędziu *HVite*. Narzędzie to na wejściu przyjmuje sieć opisującą dostępne sekwencje słów, słownik definiujący dostępne słowa wraz z ich zapisem fonetycznym oraz zestaw modeli HMM.

Przykładowy wynik rozpoznawania może wyglądać następująco:

```
#!MLF!#
"*/sample1.rec"
0 5000000 SENT-START -2374.491211
5000000 22100000 DZWOŃ -8464.383789
22100000 24200000 DO -2054.89383
24200000 31200000 KOWALSKI -804.563265
31200000 42100000 SENT-END -1054.473267
.
```

W wygenerowanym pliku podana jest transkrypcja rozpoznanej wypowiedzi, czas (region czasowy) w którym poszczególne słowo zostało wykryte oraz prawdopodobieństwo podane w skali logarytmicznej.

5.4.4. Analiza

Po stworzeniu systemu rozpoznawania mowy warto przetestować jego skuteczność. HTK udostępnia narzędzie zwane *HResults,* które na wejściu przyjmuje testowe próbki wypowiedzi wraz z ich transkrypcjami a następnie porównuje transkrypcje otrzymane w wyniku rozpoznawania i określa procentową skuteczność takiego systemu, np.:

```
====================== HTK Results Analysis =============
Date: Sun Oct 22 16:14:45 1995
Ref : testrefs.mlf
Rec : recout.mlf
------------------------ Overall Results --------------------
SENT: %Correct=98.50 [H=197, S=3, N=200]
WORD: %Corr=99.77, Acc=99.65 [H=853, D=1, S=1, I=1, N=855]
===========================================================
```

Gdzie w linii z *SENT*:

- H oznacza liczbę sentencji rozpoznanych,
- S sentencji nierozpoznanych,
- a N oznacza liczbę wszystkich sentencji.

W przypadku linii z *WORD* analogicznie tylko, że tutaj chodzi o pojedyńcze słowa i tak kolejno:

- H słowa nierozpoznane,
- S słowa rozpoznane,
- N oznacza liczbę wszystkich słów,

– a D oraz I to są błędy.

Acc oznacza skuteczność rozpoznawania podaną oczywiście w procentach. [21]

6. System sterujący robotem za pomocą komend głosowych

6.1. Schemat działania systemu

connected speech recognition

6.2. Robot przemysłowy

6.3. Urządzenie na którym działa cały system

6.4. Gry planszowe

7. Podsumowanie

7.1. Jakość systemu

7.2. Możliwości rozwoju

7.3. Wnioski

Bibliografia

[1] *CMUSphinx*. http://cmusphinx.sourceforge.net.

[2] *HTK (Hidden Model Markov Toolkit)*. http://htk.eng.cam.ac.uk/.

[3] *Julius*. http://julius.sourceforge.jp/en_index.php.

[4] *MagicScribe*. http://www.magicscribe.pl.

[5] *Primespeech*. http://www.primespeech.pl.

[6] *SARMATA*. http://www.dsp.agh.edu.pl/doku.php?id=pl:resources:asr.

[7] *WAVE PCM soundfile format*. https://ccrma.stanford.edu/courses/422/projects/WaveFormat/.

[8] Ostaszewska D. *Fonetyka i Fonologia współczesnego języka polskiego*. Wydawnictwo Naukowe PWN, Warszawa, 2000.

[9] Apple Inc. *Siri*. http://www.apple.com/ios/siri.

[10] Google Inc. *Google Mobile App*. http://www.google.com/mobile/google-mobile-app.

[11] Google Inc. *Google Now*. http://www.google.com/landing/now.

[12] Szostek K. *Rozpoznawanie mowy metodami niejawnych modeli Markowa*. PhD thesis, AGH Kraków, 2006.
http://winntbg.bg.agh.edu.pl/rozprawy/9792/full9792.pdf.

[13] Gałka J. *Optymalizacja parametryzacji sygnału w aspekcie rozpoznawania mowy polskiej*. PhD thesis, AGH Kraków, 2008.
http://winntbg.bg.agh.edu.pl/rozprawy2/10009/full10009.pdf.

[14] Kowalski A. B. Kasprzak W. Analiza sygnału mowy sterowana danymi dla rozpoznawania komend głosowych. *Postępy Robotyki, WKiŁ*, 2006. http://www.ia.pw.edu.pl/~wkasprza/PAP/kkr06_rkg.pdf.

[15] Grad L. Obrazowa reprezentacja sygnału mowy. *Biuletyn Instytutu Automatyki i Robotyki WAT nr. 8*, 1997. http://www.ita.wat.edu.pl/~l.grad/sieci%20neuronowe/ekstrakcja_cech_mowy.pdf.

[16] Microsoft. *TellMe*. http://www.microsoft.com/en-us/Tellme.

[17] Nuance. *Dragon NaturallySpeaking*. http://www.nuance.com/dragon/index.htm.

[18] Zieliński T. P. *Cyfrowe przetwarzanie sygnałów od teorii do zastosowań*. WKiŁ, Warszawa, 2005.

[19] Wielgat R. Wykłady z przedmiotu Techniki Multimedialne. PWSZ Tarnów., 2012.

[20] Juang B-I Rabiner L. *Fundamentals of Speech recognition*. Prentice Hall, Warszawa, 1993.

[21] Young S. *The HTK Book*. Cambridge, 2006. http://htk.eng.cam.ac.uk/docs/docs.shtml.

[22] Kasprzak W. *Rozpoznawanie obrazów i sygnałów mowy*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2009.

Spis rysunków

Dodatek A

Tu, trzeba zamieszczać treść dodatkową np. fragmenty kodu aplikacji itp.