Instytut Podstawowych Problemów Techniki
Polska Akademia Nauk

Rozprawa doktorska

Układ rozpoznający struktury intonacyjne w sygnale mowy

Mikołaj Wypych

Promotor:
prof. dr hab. Wiktor Jassem

Warszawa, 2011

Układ rozpoznający struktury intonacyjne w sygnale mowy

Mikołaj Wypych
Informatyka
Instytut Podstawowych Problemów Techniki
Polska Akademia Nauk
2011
Rev. 1.312

Streszczenie

Celem rozprawy było wykonanie układu (oprogramowania) rozpoznającego struktury intonacyjne w sygnale mowy polskiej. Przez strukturę intonacyjną rozumiemy kategorialną synchroniczną reprezentację przebiegu cech tonalnych (częstotliwości podstawowej), która podlega językowo specyficznej gramatyce. Rozprawa składa się z trzech części: terminologicznej, przeglądowej oraz badawczej. W części terminologicznej przedstawiono system pojęć fonetyczno–informatycznych stosowany w dalszych czaściach rozprawy. W części terminologicznej wprowadzono model komunikacji głosowej, w którym segmentalne oraz suprasegmentalne cechy sygnału mowy są traktowane równorzędnie. W części przeglądowej rozprawy opisano ponad 80 reprezentacji, algorytmów oraz układów analizy wysokości tonu (analizy tonalnej). W cześci badawczej opisano prace eksperymentalne oraz implementacyjne, przeprowadzone w ramach realizacji układu rozpoznającego struktury intonacyjne. W układzie wyróżniono trzy poziomy analizy mowy: sygnałowy, fonetyczny oraz fonologiczny. Każdy z poziomów zrealizowano w oparciu o autorskie algorytmy analizy tonalnej. Na poziomie sygnałowym zastosowano model maskowania tonalnego oraz filtr grzebieniowy w dziedzinie częstotliwości o współczynnikach wyznaczanych metodą gradientową. Na poziomie fonetycznym zastosowano algorytm parametryzacji przebiegu wysokości tonu dla odcinków pół-sylabowych z użyciem średnich ważonych harmonicznością sygnału mowy. Na poziomie fonologicznym zastosowano model statystyczny łączący metody wektorów nośnych (*Support Vector Machine*, SVM) oraz warunkowych pól losowych (*Conditional Random Field*, CRF). Zastosowany model statystyczny uczono w trybie pod całkowitym oraz częściowym nadzorem. Uczenie w trybie pod (całkowitym) nadzorem wykonano na przygotowanym w ramach prac korpusie mowy obejmującym ponad 400 fraz intonacyjnych z korpusu PoInt. Eksperymenty z uczeniem w trybie pod częściowym nadzorem wykonano na zbiorze ponad 12 tysięcy fraz intonacyjnych z korpusów PoInt oraz Babel (70 mówców). Ponadto, w pracy pokazano metodę automatycznej wizualizacji intonacji w postaci tonetycznej transkrypcji międzyliniowej oraz metodę automatycznej subkategoryzacji zbioru melodii rdzennych za pomocą algorytmu analizy skupień EM/BIC (*Expectation Maximization/Bayesian Information Criterion*). Moduły układu rozpoznawania struktur intonacyjnych zintegrowano przy użyciu autorskiego środowiska programowego opartego na architekturze tablicowej (*blackboard*).

A system recognizing pitch structures in the speech signal

Mikołaj Wypych
Computer science
Institute of Fundamental Technological Research
Polish Academy of Sciences
2011
Rev. 1.312

Abstract

The goal of the research was to develop a software system that accurately recognizes pitch structures in the Polish speech signal. The pitch structure, as understood here, is a categorical time-synchronous representation of the tonal features of the speech, which is assumed to be governed by a language-specific grammar. The thesis comprises three parts: i) terminological part, ii) review part and iii) research part. In the terminological part a set of definitions was introduced in order to clarify and specify the relationship between speech technology and phonetics. A speech communication model was presented, where both segmental and suprasegmental speech features are treated paralelly. In the review part, more than 80 representations, algorithms and systems for tonal analysis were described using terminology defined in the first part. In the research part, the design and implementation of the proposed pitch structures recognizer was described and experiments were proposed to implement and evaluate the approach. In the recognizer three levels of speech analysis are considered: signal level, phonetic level and phonological level. Processing at each of these levels was based on an original algorithm for tonal analysis. On the signal level a frequency-domain comb filter was trained using a gradient descent algorithm. Input data for the comb filter were preprocessed using tonal masking model. On the phonetic level an algorithm based on semi-syllable approximation of pitch contour weighted by harmonicity was developed. On the phonological level a hybrid discriminative statistical model based on Support Vector Machines (SVM) and Conditional Random Fields (CRF) was used. The model was trained in a supervised and semi-supervised mode. In the supervised mode a dedicated, annotated speech corpus consisting of over 400 intonation phrases was used. Experiments on the semi-supervised traiining were performed on a much larger speech corpus containing over 12 thousand intonation phrases spoken by 70 speakers. Software modules of the system were integrated by means of a blackboard type software architecture developed by the author. Two additional contributions of the thesis resulting from the research and development effort included the automatic system for interlinear tonetic transcription of speech, and a demonstration of applicability of EM/BIC (Expectation Maximization/Bayesian Information Criterion) algorithm to the problem of nuclear tones clustering.

Spis treści

I Terminologia fonetyczno–informatyczna **4**

1 Analiza anotacyjna **5**
1.1 Anotacja . . . 5
1.2 Analiza anotacyjna . . . 11
1.3 System fonologiczny . . . 13
1.4 Korpus i leksykon . . . 15

2 Komunikacja głosowa **17**
2.1 Głosowy system komunikacyjny . . . 17
2.2 Cechy tonalne i intonacja . . . 21
2.3 Akcent potencjalny i realny . . . 22
2.4 Struktura intonacyjna i interpretacja komunikatu . . . 25

II Przegląd metod analizy cech tonalnych **27**

3 Sygnałowa analiza tonalna **28**
3.1 Wstępna analiza sygnałowa . . . 31
3.1.1 Algorytmy redukcji pasma . . . 31
3.1.2 Algorytmy heurystyczne w dziedzinie czasowej . . . 32
3.1.3 Algorytmy transformacji częstotliwościowej . . . 33
3.1.4 Algorytmy artykulacyjne . . . 34
3.1.5 Algorytmy percepcyjne . . . 36
3.2 Segmentalna sygnałowa analiza tonalna . . . 39
3.2.1 Periodogramy samopodobieństwa . . . 39
3.2.2 Periodogramy cepstralne . . . 40
3.2.3 Periodogramy harmoniczne . . . 40
3.2.4 Periodogramy grzebieniowe . . . 40
3.2.5 Periodogramy percepcyjne . . . 41
3.3 Suprasegmentalna sygnałowa analiza tonalna . . . 43
3.3.1 Wygładzanie . . . 43
3.3.2 Minimalizacja kosztu . . . 43

4 Fonetyczna analiza tonalna **45**
4.1 IPO . . . 46
4.2 MOMEL . . . 48
4.3 Fujisaki . . . 50
4.4 Tilt . . . 52
4.5 Prosogram . . . 55

5 Fonologiczna analiza tonalna **57**
5.1 Fonologiczne anotacje tonalne . . . 57
5.1.1 Szkoła Brytyjska . . . 58
5.1.2 IPO . . . 61
5.1.3 SAMPROSA . . . 63
5.1.4 ToBI . . . 63
5.1.5 PROLAB . . . 67
5.1.6 INTSINT . . . 68
5.1.7 Anotacje statystyczne . . . 69
5.1.8 Anotacje intonacyjne języka polskiego . . . 72
5.2 Fonologiczna analiza tonalna . . . 75
5.2.1 Techniki oparte na wiedzy . . . 77
5.2.2 Niejawne modele Markowa . . . 78
5.2.3 Drzewa klasyfikacji i regresji . . . 79
5.2.4 Sieci neuronowe . . . 80
5.2.5 Nowe kierunki rozwoju . . . 83

III Prace badawcze **85**

6 Architektura programowa układu **86**
6.1 Typologia architektur programowych . . . 87
6.2 Realizacja architektury potokowej . . . 89
6.3 Reprezentacja wiedzy współdzielonej . . . 91
6.4 Realizacja architektury tablicowej . . . 92
6.5 Przeszukiwanie przestrzeni rozwiązań . . . 93
6.6 Konfigurowalność oraz trwałość . . . 95

7 Układ sygnałowej analizy tonalnej **101**
7.1 Tryb analizy . . . 101
7.2 Tryb uczenia . . . 105
7.3 Uczenie i wyniki . . . 108

8 Układ fonetycznej analizy tonalnej **111**
8.1 Podukład analizy ortograficznej . . . 111
8.1.1 Tokenizer . . . 112
8.1.2 Normalizer . . . 113
8.1.3 SentenceSplitter . . . 113
8.1.4 Speller . . . 114
8.1.5 Phonetizer . . . 114
8.1.6 MorphoAnalyser . . . 117
8.1.7 PhonologicalSyllabifier . . . 120
8.1.8 MorphoSyllabifier . . . 120
8.2 Podukład analizy akustycznej . . . 121

8.2.1 PhoneAligner . . . 121
8.2.2 PhoneticSyllabifier . . . 123
8.2.3 ExcursionStylizer . . . 123
8.2.4 ExcursionTracker . . . 126
8.2.5 ExcursionNormalizer . . . 129
8.3 Podukład wizualizacji . . . 129
8.3.1 LowerTadpoleStylizer . . . 130
8.3.2 UpperTadpoleStylizer . . . 131
8.3.3 PostscriptTadpole . . . 133
8.4 Wyniki . . . 134
8.4.1 Rozkłady jednowymiarowe . . . 135
8.4.2 Analiza zależności liniowych . . . 136

9 Anotacja intonacyjna języka polskiego 140
9.1 Korpus anotacji wzorcowych . . . 140
9.2 Subkategoryzacja melodii rdzennych . . . 142
9.3 Uczenie i wyniki . . . 144

10 Układ fonologicznej analizy tonalnej 152
10.1 Podukład analizy ortograficznej . . . 152
10.1.1 PotentialPositionalAccentuator . . . 153
10.1.2 PotentialLexicalAccentuator . . . 153
10.2 Podukład analizy akustycznej . . . 154
10.2.1 StateAssigner . . . 155
10.2.2 Contextualizer . . . 156
10.2.3 SVMRecognizer . . . 156
10.2.4 CRFRecognizer . . . 158
10.2.5 ToneAssigner . . . 160
10.3 Uczenie i wyniki . . . 160
10.3.1 Założenia dotyczące pomiaru skuteczności . . . 160
10.3.2 Uczenie pod nadzorem . . . 162
10.3.3 Uczenie pod częściowym nadzorem . . . 164
10.3.4 Wizualizacja wyników . . . 165

A Pseudokod wybranych algorytmów analizy tonalnej 171
A.1 IPO . . . 171
A.2 MOMEL . . . 173
A.3 Fujisaki . . . 175
A.4 Tilt . . . 177
A.5 Prosogram . . . 179
A.6 UHCF0C . . . 181

B Wizualizacje struktur intonacyjnych 183
B.1 Głos „joju” z korpusu PoInt . . . 183
B.2 Głos „mawa” z korpusu PoInt . . . 185
B.3 Głos „pano” z korpusu PoInt . . . 186

Spis rycin

2.1 Głosowy system komunikacyjny. Diagram klas UML. 18
2.2 Intonacyjny system komunikacyjny. Diagram klas UML. 23

3.1 Krótkookresowy algorytm ekstrakcji F_0. Diagram klas UML. 29
3.2 Progi percepcji tonu maskowanego przez przebieg szumowy wąskopasmowy o częstotliwości 1kHz oraz energii L_{CB} (Fastl i Zwicker 2007, 65). 36
3.3 Model percepcji wysokości tonu według Lyona (za: Slaney 1988, 65). 38

4.1 Przykładowa anotacja fonetyczna Steele'a (1775). 46
4.2 Przykładowa tonetyczna transkrypcja międzyliniowa (Cruttenden 1997, 36). 46
4.3 Wizualizacja anotacji IPO przykładowego sygnału mowy w języku angielskim (Hermes 2006, 37). 48
4.4 Wizualizacja anotacji MOMEL przykładowego sygnału mowy w języku francuskim (Hirst i Espesser 1993). 50
4.5 Wizualizacja anotacji Fujisaki przykładowego sygnału mowy w języku japońskim (Fujisaki 2004). 52
4.6 Wizualizacja anotacji Tilt przykładowego sygnału mowy w języku angielskim (Taylor 2000). 54
4.7 Prozogram przykładowego sygnału mowy w języku francuskim (Mertens 2009). 56

5.1 Gramatyka BT frazy intonacyjnej języka angielskiego brytyjskiego. 61
5.2 Gramatyka IPO frazy intonacyjnej języka niderlandzkiego. 62
5.3 Gramatyka ToBI frazy intonacyjnej języka angielskiego amerykańskiego (w zapisie sekwencyjnym). 66
5.4 Gramatyka PROLAB frazy intonacyjnej języka niemieckiego (w zapisie sekwencyjnym). 68
5.5 Gramatyka INTSINT ciągu etykiet intonacyjnych dowolnego języka. 68
5.6 Topologia HMM w algorytmie UHCF0C (Lolive i inni 2007). 71

6.1 Architektura potokowa w SLOPE. Diagram klas UML. 89
6.2 Reprezentacje sygnału w SLOPE. Diagram klas UML. 96
6.3 Ontologie niskopoziomowe w SLOPE. Diagram klas UML. 97
6.4 Ontologie wysokopoziomowe w SLOPE. Diagram klas UML. 98
6.5 Grafowa reprezentacja anotacji w SLOPE. Diagram klas UML. 99
6.6 Architektura tablicowa w SLOPE. Diagram klas UML. 100
6.7 Hierarchia komunikatów w SLOPE. Diagram klas UML. 100

7.1 Ekstraktor F_0 w trybie analizy. Diagram współpracy UML. 102
7.2 Przebieg sygnałów na wyjściu komponentów (kolejno od góry): LSU, MKU oraz TPU/LEU. Fraza intonacyjna: mawa/pochodzenie--01-011, mówca: „mawa" (kobieta). 105
7.3 Przebieg sygnałów na wyjściu komponentów (kolejno od góry): LSU, MKU oraz TPU/LEU. Fraza intonacyjna: pano/pochodzenie-026, mówca: „pano" (mężczyzna). 105
7.4 Ekstraktor F_0 w trybie uczenia. Diagram współpracy UML. 106
7.5 Uczenie periodogramu w iteracji pierwszej. Gradient $\nabla E(\widehat{A}_0)$ (strona lewa) oraz macierz $\widehat{A}_1$ (strona prawa). Współczynnik $\eta = 1$. 108
7.6 Uczenie periodogramu w iteracji drugiej. Gradient $\nabla E(\widehat{A}_1)$ (strona lewa) oraz macierz $\widehat{A}_2$ (strona prawa). Współczynnik $\eta = 2/3$. 108
7.7 Uczenie periodogramu w iteracji szóstej. Gradient $\nabla E(\widehat{A}_5)$ (strona lewa) oraz macierz $\widehat{A}_6$ (strona prawa). Współczynnik $\eta = 1/4$. 109

8.1 Podukład analizy ortograficznej w układzie fonetycznej analizy tonalnej. Diagram komponentów UML. 112
8.2 Podukład analizy akustycznej w układzie fonetycznej analizy tonalnej. Diagram komponentów UML. 122
8.3 Podukład wizualizacji metodą ITT. Diagram komponentów UML. 130
8.4 Empiryczne rozkłady jednowymiarowe ontologii NormalizedPitchExcursion na korpusach PoS1 oraz BaS1. 138
8.5 Empiryczne rozkłady jednowymiarowe ontologii NormalizedPitchExcursion na korpusach BaS1F oraz BaS1M. 139

9.1 Rozkłady empiryczne oraz teoretyczne (geometryczne i Poissona) długości melodii w korpusie PoS1. Wykresy dla poszczególnych melodii. 147
9.2 Rozkłady empiryczne oraz teoretyczne (geometryczne i Poissona) długości melodii w korpusie PoS1. Wykresy zbiorcze dla melodii nieakcentowanych oraz akcentowanych. 148
9.3 Wyniki metody MCLUST dla melodii rdzennych rosnących (zbiór uczący $PoS1^h_{nuri}$) 149
9.4 Wyniki metody MCLUST dla melodii rdzennych opadających (zbiór uczący $PoS1^h_{nufa}$) 150
9.5 Wyniki metody MCLUST dla melodii rdzennych opadających (zbiór uczący $PoS1N^h_{nufa}$) 151

10.1 Podukład analizy ortograficznej w układzie fonologicznej analizy tonalnej. Diagram komponentów UML. 153
10.2 Podukład analizy akustycznej w układzie fonologicznej analizy tonalnej. Diagram komponentów UML. 155
10.3 Topologia modelu grafowego pojedynczej melodii. 155

Spis tabel

1.1 Wybrane skale czasu trwania segmentów. 6

3.1 Wybrane algorytmy heurystyczne wstępnej analizy sygnałowej. 32

5.1 Etykiety (ikony) tonalne BT dla języka angielskiego brytyjskiego (O'Connor i Arnold 1973). 59
5.2 Grupy etykiet (ikon) tonalnych BT dla języka angielskiego brytyjskiego. . . . 60
5.3 Etykiety tonalne IPO dla języka niderlandzkiego. 62
5.4 Etykiety tonalne SAMPROSA (językowo uniwersalne). 64
5.5 Etykiety tonalne ToBI dla języka angielskiego amerykańskiego. 65
5.6 Etykiety tonalne PROLAB dla języka niemieckiego. 67
5.7 Etykiety tonalne INTSINT (językowo uniwersalne). 69
5.8 Etykiety tonalne Jassema dla języka polskiego. 74
5.9 Nowe systemy uczące się w fonologicznej analizie tonalnej. 83

6.1 Architektury oraz środowiska programowe układów przetwarzania mowy. . . 88

7.1 Odsetek grubych błędów ekstrakcji F_0 w korpusie Keele. Głosy męskie. . . . 109
7.2 Odsetek grubych błędów ekstrakcji F_0 w korpusie Keele. Głosy kobiece. . . . 109

8.1 Segmentalny system fonetyczny i fonologiczny języka polskiego. Cz. 1. 118
8.2 Segmentalny system fonetyczny i fonologiczny języka polskiego. Cz. 2. 119
8.3 Korpusy do badania rozkładów ontologii StandarizedPitchExcursion. 135
8.4 Macierz korelacji ontologii StandarizedPitchExcursion. Korpus PoS1. 136
8.5 Macierz korelacji ontologii StandarizedPitchExcursion. Korpus BaS1. 136
8.6 Odchylenia standardowe w analizie głównych składowych ontologii StandarizedPitchExcursion. Korpusy PoS1 oraz BaS1. 136
8.7 Wektory własne w analizie głównych składowych ontologii StandarizedPitchExcursion (pominięto wartości <0.1). Korpus PoS1. 137
8.8 Wektory własne w analizie głównych składowych ontologii StandarizedPitchExcursion (pominięto wartości <0.1). Korpusi BaS1. 137

9.1 Rozkład empiryczny etykiet intonacyjnych Jassema w korpusie PoS1. 142
9.2 Dopasowanie rozkładów liczby sylab obejmowanych przez najczęstsze etykiety intonacyjne Jassema. Test χ^2 na korpusie PoS1. 142
9.3 Etykiety melodii rdzennych otrzymanych metodą MCLUST. 145

10.1 Skuteczność procentowa układu rozpoznającego struktury intonacyjne. Wyniki uczenia pod nadzorem na korpusie PoS1. 163
10.2 Skuteczność procentowa układu rozpoznającego struktury intonacyjne opartego na modelu SVMRBF/CRF. Uczenie pod częściowym nadzorem na podzbiorach korpusu PoS2. 166
10.3 Skuteczność procentowa układu rozpoznającego struktury intonacyjne opartego na modelu SVMRBF/CRF. Uczenie pod częściowym nadzorem na podzbiorach korpusu BaS1. 167
10.4 Proponowane ikony melodii. Literę „a” dodano dla pokazania lokalizacji ikony względem tekstu. 168

Spis algorytmów

6.1 Wytyczanie ścieżki minimalnej w anotacji kratowej (środowisko SLOPE). . . 94
8.1 Translacja reguł transkrypcji fonematycznej (fonetycznej) z tabel i wyjątków do kodu źródłowego w języku C. 116
8.2 Wyznaczanie granic części sylaby fonetycznej na podstawie sylabizacji fonologicznej oraz harmoniczności. 124
8.3 Wyznaczanie przebiegu wysokości tonu w granicach sylaby fonologicznej. . . 125
8.4 Detekcja i korekta ontologii StylizedPitchExcursion mających niską istotność percepcyjną. 127
8.5 Detekcja i korekta ontologii StylizedPitchExcurison zawierających grube błędy przebiegu wysokości tonu. 128
8.6 Wyznaczanie segmentu oraz ontologii LowerTadpolePitchExcursion. Algorytm indukcyjny. 131
8.7 Wyznaczanie ścieżki etykietowanej ontologiami UpperTadpolePitchExcursion. Algorytm indukcyjno–dedukcyjny. 132
A.1 Fonetyczna analiza tonalna IPO (Hermes 2006, 36). 172
A.2 Redukcja mikrointonacji w fonetycznej analizie tonalnej MOMEL (Hirst i inni 2000, 63). 173
A.3 Wyznaczenie etykiet kandydujących w fonetycznej analizie tonalnej MOMEL (Hirst i inni 2000, 63). 174
A.4 Wyznaczanie segmentacji w fonetycznej analizie tonalnej MOMEL Hirst i inni (2000, 64). 175
A.5 Fonetyczna analiza tonalna Fujisaki (Mixdorff 2000). 176
A.6 Inicjalizacja komend akcentowych w fonetycznej analizie tonalnej Fujisaki (Mixdorff 2000). 176
A.7 Inicjalizacja komend frazowych w fonetycznej analizie tonalnej Fujisaki (Mixdorff 2000). 177
A.8 Fonetyczna analiza tonalna RFC (Taylor 1995a, 7). 178
A.9 Fonetyczna analiza tonalna Prosogram (Mertens 2009). 179
A.10 Tworzenie segmentacji w fonetycznej analizie tonalnej Prosogram (Mertens 2009). 180
A.11 Tworzenie fonetycznego systemu tonalnego metodą UHCF0C (Lolive i inni 2007). 181

Oznaczenia ogólne

$\varnothing$ wartość nieokreślona

i jednostka urojona

e liczba Eulera

$|a|$ wartość bezwzględna (moduł) liczby a

$\lfloor a \rfloor$ największa liczba całkowita, niewiększa od liczby a (podłoga)

$\lceil a \rceil$ najmniejsza liczba całkowita, niemniejsza od liczby a (sufit)

a^* liczba sprzężona do liczby a

$\approx$ wartość przybliżona

$\ll$ wartość znacznie mniejsza

$\gg$ wartość znacznie większa

$\propto$ proporcjonalność

$\emptyset$ zbiór pusty

$\mathbb{N}$ zbiór liczb naturalnych (włączając 0)

$\mathbb{N}_+$ zbiór liczb naturalnych (wyłączając 0)

$\mathbb{Z}$ zbiór liczb całkowitych

$\mathbb{R}$ zbiór liczb rzeczywistych

$\mathbb{C}$ zbiór liczb zespolonych

$\subset$ podzbiór ostry

$\supset$ nadzbiór ostry

$|w|$ liczba elementów ciągu, wektora lub zbioru w, liczba znaków w napisie w

$u \oplus w$ konkatenacja ciągów, wektorów lub napisów u i w

$w[i]$ i-ty element ciągu, wektora lub n-tki uporządkowanej w, i-ty znak w napisie w; $i \in \{0, 1, \ldots, |w| - 1\}$

$w[i..j]$ podciąg, podwektor (podnapis) w złożony z kolejnych elementów (znaków): $w[i], w[i+1], \ldots, w[j]$

$\mathrm{ind}(w)$ zbiór $\{0, 1, \ldots, |w| - 1\}$

X^* domknięcie Kleene'ego zbioru X

f^{-1} funkcja odwrotna do f

$x * y$ splot sygnałów x i y

$X \leftrightsquigarrow x$ X jest transformatą x

μ wartość oczekiwana

σ odchylenie standardowe

ρ korelacja

$E(X)$ wartość oczekiwana X

$Var(X)$ wariancja X

$w \sim v$ w oraz v są sąsiadami w grafie

T_0 okres podstawowy

F_0 częstotliwość podstawowa

F_1, F_2, ... częstotliwość formantu 1, 2, ...

Oznaczenia specyficzne

ζ skala (funkcja)

$\mathbb{R}_{\varnothing}$ zbiór $\mathbb{R} \cup \{\varnothing\}$

$\mathbb{A}$ zbiór wszystkich kotwic

$\mathbb{S}$ zbiór wszystkich segmentów

$S[i]$ i-ty segment segmentacji sekwencyjnej S

$S(t)$ segment segmentacji prostej obejmujący punkt czasowy t

$\#u$ identyfikator kotwicy lub segmentu s

$\overleftarrow{s}$ lewa kotwica segmentu s

$\overrightarrow{s}$ prawa kotwica segmentu s

$\overline{s}$ etykieta segmentu s

$\boxplus(x, s)$ okno segmentu s w sygnale x

$\overleftarrow{S}$ zb. lewych kotwic segmentów ze zb. S

$\overrightarrow{S}$ zb. prawych kotwic segmentów ze zb. S

$\overleftrightarrow{S}$ zbiór $\overleftarrow{S} \cup \overrightarrow{S}$

$\bowtie S$ zbiór ścieżek segmentów ze zb. S

$\rhd$ relacja ograniczania ścieżkowego segmentów

$\blacktriangleright$ relacja ograniczania czasowego segmentów

$_{(-)}$ melodia słaba równa kończąca się pod

$^{(-)}$ melodia słaba równa kończąca się nad

$_{(/)}$ melodia słaba rosnąca kończąca się pod

$\_$ melodia silna równa kończąca się pod

$^-$ melodia silna równa kończąca się nad

$_\backslash$ melodia silna opadająca kończąca się pod

$^\backslash$ melodia silna opadająca kończąca się nad

$_/$ melodia silna rosnąca kończąca się pod

$^/$ melodia silna rosnąca kończąca się nad

$=$ melodia rdzenna równa

$/\!\!/$ melodia rdzenna rosnąca pełna

$_{/\!\!/}$ melodia rdzenna rosnąca niska

$^{/\!\!/}$ melodia rdzenna rosnąca wysoka

$\backslash\!\!\backslash$ melodia rdzenna opadająca pełna

$_{\backslash\!\backslash}$ melodia rdzenna opadająca niska

$^{\backslash\!\backslash}$ melodia rdzenna opadająca wysoka

$/\!\!/\backslash\!\!\backslash$ melodia rdzenna rosnąco–opadająca

$\backslash\!\!\backslash/\!\!/$ melodia rdzenna opadająco–rosnąca

$\mathcal{M}$ zbiór melodii

$\mathcal{O}$ zbiór stanów modelu frazy intonacyjnej

$\mathcal{S}$ zbiór stanów modelu melodii

Skróty

ACF *Auto–Correlation Function*

AG *Annotation Graph*

AGC *Automatic Gain Control*

AM *Acoustic Model*

AMDF *Average Magnitude Difference Function*

ANN *Artificial Neural Network*

API *Application Programming Interface*

ASCII *American Standard Code for Information Interchange*

ASR *Automatic Speech Recognition*,

BFGS Broyden-Fletcher-Goldfarb-Shanno

BIC *Bayesian Information Criterion*

BT *British Tradition*

bps *bits per second*

CART *Classification And Regression Tree*

CDHMM *Continous Density Hidden Markov Model*

CDGMM *Continous Density Gaussian Mixture Model*

CELP *Codebook Excited Linear Prediction*

CFT *Continuous Fourier Transform*

CRF *Conditional Random Field*

CSTR *The Centre for Speech Technology Research*

dB decybel

DAO *Data Access Object*

DC *Direct Current*

DFT *Discrete Fourier Transform*

DSP *Digital Signal Processing*

DTW *Dynamic Time Warping*

EGG elektroglotograf/elektroglotogram

ERB *Equivalent Rectangular Bandwidth*

EM *Expectation Maximization*

FFT *Fast Fourier Transform*

FIR *Finite Impulse Response*

FS *Feature Structure*

FSA *Finite–State Automaton*

FST *Finite–State Transducer*

FWR *Full–Wave Rectifier*

GM *Gaussian Mixture*

Hz Herc

HMM *Hidden Markov Model*

HTK *Hidden Markov Model Toolkit*

HWR *Half–Wave Rectifier*

IIR *Infinite Impulse Response*

IIS *Improved Iterative Scaling*

INRIA *Institut National de Recherche en Informatique et Automatique*

INTSINT *INternational Transcription System for INTonation*

IPA *International Phonetic Alphabet*

IPP *Intel Performance Primitives*

ITT *Interlinear Tonetic Transcription*

JND *Just Noticeable Difference*

KIM *Kiel Intonation Model*

K–S Kołmogorowa–Smirnowa (test)

LPC *Linear Predictive Coding*

LBFGS *Limited–memory BFGS*

LBG Linde–Buzo–Gray

LM *Language Model*

LTASS *Long–Term Average Speech Spectrum*

LTS *Letter–To–Sound*

LTI *Linear Time–Invariant*

MDS *Multi–Dimensional Scaling*

Mel jednostka wysokości tonu (*Melody*)

MFCC *Mel–Frequency Cepstral Coefficient*

MIT *Massachusetts Institute of Technology*

MLDS *Multi–Level Data Structure*

MLE *Maximum Likelihood Estimate*

MLP *Multi–Layer Perceptron*

MOMEL *MELodic MOdelisation*

MOS *Mean Opinion Score*

MSE *Mean Squared Error*

ms milisekunda

NLP *Natural Language Processing*

NP *Nondeterministic Polynomial*

NT *Nuclear Tune*

OCR *Optical Character Recognition*

PaIntE *Parametric Intonation Event*

PLP *Perceptual Linear Prediction*

PMVDR *Perceptual Minimum Variance Distortionless Response*

POSIX *Portable Operating System Interface*

PSOLA *Pitch–Synchronous Overlap and Add*

RBF *Radial Basis Function*

RMS *Root Mean Square*

RMSE *Root Mean Squared Error*

RNN *Recurrent Neural Network*

RSCI *Reversible Symbolic Coding of Intonation*

SAM *Speech Assessment Methods*

SAMPA *SAM Phonetic Alphabet*

SAMPROSA *SAM Prosodic Transcription*

SFS *Speech Filing System*

SGD *Stochastic Gradient Descent*

SIL *Sound Intensity Level*

SIMD *Single Instruction Multiple Data*

SLOPE *Spoken Language Open Processing Environment*

SLP *Single Layer Perceptron*

SNR *Signal–to–Noise Ratio*

SOM *Self Organizing Maps*

SPL *Sound Pressure Level*

sPT *strong Prenuclear Tune*

SSB *Single Side Band*

SSE *Sum of Squared Error*

ST *Semi–Tone*

SVM *Support Vector Machines*

TDNN *Time–Delay Neural Network*

TTS *Text–To–Speech*

ToBI *Tones and Break Indices*

UCL *University College London*

UHC *Unsupervised HMM Classification*

UML *Unified Modeling Language*

VAD *Voice Activity Detection*

VQ *Vector Quantization*

WFSM *Weighted Finite State Machine*

wPT *weak Prenuclear Tune*

Konwencje typograficzne

Pierwsze wystąpienie terminu, który jest definiowany w danym akapicie wyróżniono pismem pogrubionym, np. **częstotliwość podstawowa**. Terminy lingwistyczne, które nie zostały zdefiniowane w pracy i są rozumiane w znaczniu ogólnym wyróżniono kursywą, np. *struktura składniowa*. Tłumaczenia terminów na język obcy wyróżniono pismem pochylonym, np. *fundamental frequency*.

Transkrypcje ortograficzne sygnału mowy zapisano w cudzysłowach ostrokątnych, np. «Widzę ciemność.». Transkrypcje fonematyczne sygnału mowy zapisano w nawiasach kwadratowych stosując symbole IPA (2005), np. [vidze tɕemnoɕtɕ]. Sylaby akcentowane wyróżniono podkreśleniem. Granice sylab, leksów (wyrazów) oraz fraz intonacyjnych oznaczono odpowiednio krótką pionową kreską, spacją oraz podwójną pionową kreską, np. «Wi|dzę | ciem|ność. ‖ Ciem|ność | wi|dzę!», [vi|dze | tɕem|noɕtɕ ‖ tɕem|nozʲdzʲ | vi|dze].

W zapisie liczb za separator miejsca dziesiętnego przyjęto kropkę, np. ≈ 3.141. Zmienne skalarne oznaczono małymi literami alfabetu łacińskiego pismem pochylonym, np. a, b. Zmienne wektorowe oznaczono małymi literami alfabetu łacińskiego pismem pochylonym pogrubionym, np. $\boldsymbol{a}$, $\boldsymbol{b}$. Zmienne macierzowe oznaczono wielkimi literami alfabetu łacińskiego pismem pochylonym pogrubionym, np. $\boldsymbol{A}$, $\boldsymbol{B}$.

Nazwy typów danych oraz klas zapisano krojem bezszeryfowym, np. integer, Set. Literały napisowe oraz znakowe zapisano krojem maszynowym między apostrofami, np. `'y'`, `'hello, world'`. Wyrażenia regularne zapisano krojem maszynowym między ukośnikami zgodnie ze standardem POSIX (2004), np. `/[Hh]el{2}o,?\s+[Ww]orld!?/`. Diagramy modeli informatycznych zapisano w języku UML 2.x (OMG 2010).

Wstęp

Termin „intonacja" jest rozumiany różnie w zależności od dyscypliny naukowej. W publikacjach lingwistycznych przez intonację rozumie się m.in. „to, jak mówcy używają zmian wysokości tonu aby wyrazić znaczenia językowe oraz pragmatyczne" (Wells 2006, 1) oraz „kontrastywne użycie zmienności tonu w celu wyrażenia znaczenia dyskursywnego lub frazowania" (Gussenhoven 2004, 22). W publikacjach technicznych przez intonację rozumie się m.in. „zmiany częstotliwości podstawowej podyktowane składnią oraz semantyką" (Hess 1983, 4-5) oraz „ślad procesu, w którym pewne rodzaje informacji pochodzące od mówcy są wyrażane w przebiegu częstotliwości podstawowej" (Fujisaki 2000).

W zależności od języka intonacja może być jedynym źródłem informacji umożliwiającym odróżnienie trybu zdania (np. stwierdzenia od zapytania w języku polskim), części mowy (np. rzeczownika od czasownika w języku angielskim) lub znaczenia leksykalnego (np. denotatu w języku szwedzkim). Intonacja należy do głównych źródeł informacji w komunikacji niewerbalnej. Zgodnie z niedawnymi badaniami, odsetek niewerbalnych komunikatów językowych w komunikacji osobowej sięga 49% (Campbell 2006).

Intonacja jest analizowana w kontekście cech tonalnych sygnału mowy, tj. akustycznych obrazów aktywności fałdów głosowych. Przyjmuje się, że pierwszą (w czasach nowożytnych) pracę na temat analizy cech tonalnych opublikował Steele (1775). Znaczny wzrost zainteresowania tym tematem nastąpił w latach dwudziestych i trzydziestych ubiegłego wieku wraz z dostrzeżeniem perspektyw aplikacyjnych w dydaktyce (Klinghardt i Klemm 1920; Palmer 1922) oraz telekomunikacji (Dudley 1935; Grützmacher i Lottermoser 1937). Kolejnym stymulatorem badań nad analizą cech tonalnych były liczne projekty dotyczące syntezy oraz rozpoznawania mowy. Jassem (1973, 344) napisał, że wyczerpująca lista wcześniejszych publikacji na temat analizy cech tonalnych w mowie obejmuje ok. 1500 pozycji. Dziesięć lat później Hess (1983, 17) zamieścił bibliografię z tego samego zakresu zawierającą około 2000 pozycji. Zgodnie z niedawną publikacją Hessa (2008) liczbę publikacji na temat sygnałowej analizy cech tonalnych szacuje się obecnie na 3000. Liczba podana przez Hessa nie obejmuje coraz częściej publikowanych prac z zakresu fonetycznej oraz fonologicznej analizy cech tonalnych, których liczbę autor niniejszej pracy szacuje na co najmniej 500. Współcześnie cech tonalnych dotyczy znacząca liczba referatów na cyklicznych konferencjach naukowych Interspeech oraz ICPhS. Cechy tonalne są tematem przewodnim dwuletniej konferencji naukowej Speech Prosody.

W Polsce pierwsze prace nad układami sygnałowej analizy cech tonalnych (ekstrakcji częstotliwości podstawowej) w mowie wykonali Kubzdela (1976) oraz Gubrynowicz i inni (1980). W ostatnich kilku latach prace nad algorytmami fonologicznej analizy cech tonalnych w mowie polskiej prowadziły m.in. Oliver (2008) oraz Wagner (2008).

Celem niniejszej pracy było stworzenie układu (oprogramowania) rozpoznającego struktury intonacyjne w sygnale mowy polskiej. Przez strukturę intonacyjną rozumie się kategorialną reprezentację przebiegu wysokości tonu, która podlega zależnej od języka gramatyce. Zakłada się, że na wejściu układu dany jest cyfrowy sygnał mowy zawierający pojedynczą frazę intonacyjną, jego transkrypcja ortograficzna oraz identyfikator mówcy. Układ daje na wyjściu strukturę intonacyjną zgodną z gramatyką frazy intonacyjnej Jassema (2003a).

Główne tematy badawcze podejmowane w pracy:

- analiza częstotliwości podstawowej w oparciu o model maskowania tonalnego oraz filtr grzebieniowy w dziedzinie częstotliwości (rozdział 7),
- algorytm integracji czasowej oraz korekcji przebiegów częstotliwości podstawowej oparty na segmentach półsylabowych oraz miarach istotności percepcyjnej (rozdział 8),
- algorytm analizy skupień melodii rdzennych oparty na maksymalizacji wartości oczekiwanej z bayesowskim kryterium informacyjnym (EM/BIC) (rozdział 9),
- statystyczny model frazy intonacyjnej łączący metodę wektorów nośnych (SVM) oraz metodę warunkowych pól losowych (CRF) (rozdział 10),

Główne tezy sprawdzane w pracy:

- połączenie modelu maskowania tonalnego z uczonym metodą gradientową filtrem grzebieniowym pozwala otrzymać skuteczny układ sygnałowej analizy tonalnej (rozdział 7),
- algorytm analizy skupień EM/BIC pozwala, przyjmując statystyczne kryterium dystynktywności, określić liczbę kategorii monotonicznych melodii nuklearnych w spontanicznej mowie polskiej (rozdział 9),
- układ analizy struktur intonacyjny oparty na modelu SVM/CRF przewyższa skutecznością taki sam układ oparty na modelu SVM (rozdział 10).

Oprócz tego w rozdziałach 1 i 2 przedstawiono system terminologiczny, który ma w zamierzeniach ułatwić informatyczną interpretację szeregu pojęć fonetycznych stosowanych w niniejszej pracy. W końcowej części pracy zaprezentowano wstępne wyniki eksperymentu z zastsowaniem uczenia modelu SVM/CRF w trybie pod częściowym nadzorem. W pracy wykorzystano fragmenty korpusów mowy polskiej PoInt (Karpiński 2002) oraz Babel (Gubrynowicz 1999) zawierające łącznie ponad 12 tysięcy fraz intonacyjnych pochodzących od 70 mówców. Integrację modułów programowych wykonanego układu zrealizowano w autorskim środowisku programowym opartym na architekturze tablicowej (rozdział 6).

Praca składa się ze wstępu, dziesięciu numerowanych rozdziałów, podsumowania oraz dwóch dodatków. W rozdziale **pierwszym** przedstawiono proponowany system pojęć fonetyczno–informatycznych oparty na grafowo–porządkowej reprezentacji wiedzy. W rozdziale **drugim** opisano proponowany model głosowego systemu komunikacyjnego z uwzględnieniem cech

tonalnych wzorowany m.in. na klasycznym modelu komunikacyjnym Shannona. W proponowanym modelu wyróżniono trzy poziomy: 1) sygnałowy, 2) fonetyczny oraz 3) fonologiczny. Na uwagę zasługuje równorzędne traktowanie segmentalnych oraz suprasegmentalnych cech sygnału mowy. W rozdziałach **trzecim**, **czwartym** oraz **piątym** zamieszczono przeglądy metod analizy cech tonalnych odpowiednio na poziomie sygnałowym, fonetycznym oraz fonologicznym. W rozdziałach przeglądowych opisano ponad 80 reprezentacji, algorytmów oraz układów analizy cech tonalnych przy wykorzystaniu terminologii wprowadzonej w dwóch początkowych rozdziałach pracy. W rozdziale **szóstym** opisano środowisko programowe użyte do zintegrowania układu rozpoznającego struktury intonacyjne. W rozdziałach **siódmym** oraz **ósmym** opisano proponowane układy analizy cech tonalnych na poziomach: sygnałowym oraz fonetycznym. W rozdziale **dziewiątym** opisano prace nad zbiorem wzorcowych struktur intonacyjnych stosowanych przy uczeniu proponowanego układu rozpoznającego struktury intonacyjne. W rozdziale **dziesiątym** opisano proponowany układ analizy cech tonalnych na poziomie fonologicznym, który (wraz z układami analizy na niższych poziomach) rozpoznaje struktury intonacyjne w sygnale mowy. W dodaktu A przedstawiono fonetyczno–informatyczną interpretację wybranych algorytmów analizy cech tonalnych. W dodatku B przedstawiono wizualizację struktur intonacyjnych wybranych fraz intonacyjnych z korpusów mowy PoInt oraz Babel.

Autor dziękuje prof. drowi hab. Maciejowi Karpińskiemu za udostępnienie korpusu mowy PoInt (Karpiński 2002), prof. drowi hab. Ryszardowi Gubrynowiczowi za udostępnienie korpusu mowy Babel (Gubrynowicz 1999) oraz drowi Georgowi Meyerowi za udostępnienie korpusu mowy Keele (Plante i inni 1995).

Część I

Terminologia fonetyczno–informatyczna

ROZDZIAŁ 1

Analiza anotacyjna

1.1 Anotacja

W pracach z zakresu przetwarzania mowy zaproponowano szereg reprezentacji wiedzy, których podstawą jest powiązanie danych numerycznych lub symbolicznych z punktami lub odcinkami na osi czasu sygnału cyfrowego. Do reprezentacji tych należą m.in.: *wielowarstwowa struktura danych* (MLDS) (Hertz 1988), *struktura cech* (FS) (Taylor i inni 1998a), *graf anotacyjny* (AG) (Bird i Liberman 2001), *anotacja wielowarstwowa* (Cassidy i Harrington 2001) oraz *mapy czasowe* (Gibbon 2006). W niniejszej sekcji wprowadzamy reprezentację wiedzy, która jest częściowo oparta na postulatach Birda i Libermana (2001).

Definicja 1.1. Kotwicą nazywamy parę uporządkowaną $(i, \boldsymbol{t}) \in \mathbb{N} \times (\mathbb{R}_{\varnothing})^n$, gdzie $n \in \mathbb{N}_+$. Wartości n, i oraz $\boldsymbol{t}$ nazywamy odpowiednio **wymiarem**, **identyfikatorem** oraz **czasem** kotwicy $(i, \boldsymbol{t})$.

Kotwica jest to identyfikowalny punkt w przestrzeni, której wymiary utożsamia się z osiami czasowymi n sygnałów. Przez $\#a$ będziemy oznaczać identyfikator kotwicy a. Dopuszcza się, by w pewnych wymiarach czas kotwicy był nieokreślony (wartość $\varnothing$).

Przez $\mathbb{A}^n$ będziemy oznaczać zbiór wszystkich kotwic n-wymiarowych. Dla uproszczania zapisu będziemy pomijać indeks n wszędzie tam, gdzie liczba wymiarów nie ma wpływu na poprawność stwierdzeń.

Niech $(i_0, \boldsymbol{t_0}), (i_1, \boldsymbol{t_1}) \in \mathbb{A}^n$. Relację ostrego porządku częściowego $<^k \subset \mathbb{A}^n \times \mathbb{A}^n$, gdzie $k \in \{0, 1, \ldots, n$ określamy następująco:

$$(i_0, \boldsymbol{t_0}) <^k (i_1, \boldsymbol{t_1}) \iff \boldsymbol{t_0}[k] < \boldsymbol{t_1}[k]. \tag{1.1}$$

Zakładamy, że do relacji $< \subset \mathbb{R}_{\varnothing} \times \mathbb{R}_{\varnothing}$ nie należy żadna para elementów, z których przynajmniej jeden ma wartość $\varnothing$. Analogicznie jak we wzorze 1.1 określamy relacje $>^k$, $\leqslant^k$ oraz $\geqslant^k$. Jeśli $n = 1$, to w oznaczeniach relacji będziemy pomijać indeks k.

Tabela 1.1: Wybrane skale czasu trwania segmentów.

Definicja	Rodzaj
$\zeta_L^{\text{linear}} = Ld,$ gdzie $L > 0$ jest dowolną wartością liczbową.	liniowa
$\zeta_{F,M}^{\text{Klatt}}(d) = \frac{d-F}{M-F} \cdot 100\%,$ gdzie F oraz M są odpowiednio minimalną oraz średnią wartością d w pewnym zbiorze segmentów (Klatt 1979).	ilorazowa
$\zeta_{M,D}^{\text{Campbell}}(d) = \frac{d-M}{D},$ gdzie M oraz S są odpowiednio wartością średnią oraz odchyleniem standardowym wartości d w pewnym zbiorze segmentów (Campbell 1992).	interwałowa

Operator różnicy $- : \mathbb{A}^n \times \mathbb{A}^n \mapsto (\mathbb{R}_\varnothing)^n$ określamy następująco:

$$(i_0, \boldsymbol{t_0}) - (i_1, \boldsymbol{t_1}) = \boldsymbol{t_0} - \boldsymbol{t_1}. \tag{1.2}$$

W szczególności:

$$((i_0, \boldsymbol{t_0}) - (i_1, \boldsymbol{t_1}))[k] = \varnothing \iff \boldsymbol{t_0}[k] = \varnothing \vee \boldsymbol{t_1}[k] = \varnothing, \tag{1.3}$$

przy czym zakładamy, że do relacji $=$ na zbiorze $\mathbb{R}_\varnothing$ należy para $(\varnothing, \varnothing)$.

Definicja 1.2. Segmentem nazywamy trójkę uporządkowaną $(i, b, f) \in \mathbb{N} \times \mathbb{A}^n \times \mathbb{A}^n$, która spełnia warunek:

$$\nexists_k\, b >^k f. \tag{1.4}$$

Wartość i nazywamy **identyfikatorem segmentu**. Wartość b nazywamy **lewą kotwicą** segmentu. Wartość f nazywamy **prawą kotwicą** segmentu.

Zbiór wszystkich segmentów o kotwicach n-wymiarowych będziemy oznaczać przez $\mathbb{S}^n$, tj. $\mathbb{S}^n = \mathbb{N} \times \mathbb{A}^n \times \mathbb{A}^n$. Dla uproszczenia zapisu pomijamy indeks n w $\mathbb{S}^n$ wszędzie tam, gdzie liczba wymiarów nie ma wpływu na poprawność stwierdzeń.

Niech $s = (i, b, f) \in \mathbb{S}$. Przyjmujemy następujące oznaczenia pomocnicze: $\#s = i$, $\overleftarrow{s} = b$ oraz $\overrightarrow{s} = f$.

Definicja 1.3. Czasem trwania segmentu s w wymiarze k nazywamy wartość $\zeta(\boldsymbol{d}[k])$, gdzie $\boldsymbol{d} = \overrightarrow{s} - \overleftarrow{s}$ a ζ jest przyjętą skalą pomiarową.

Tabela 1.1 zawiera przykładowe skale czasu trwania segmentów.

Niech będzie dany zbiór $S \subset \mathbb{S}$. Przez $\overleftarrow{S}$ oznaczamy zbiór lewych kotwic segmentów należących do S, tj. $\overleftarrow{S} = \{a \in \mathbb{A} : \exists_{s \in S} \overleftarrow{s} = a\}$. Analogicznie, przez $\overrightarrow{S}$ oznaczamy zbiór prawych kotwic segmentów a przez $\#S$ oznaczamy zbiór identyfikatorów segmentów należących do segmentacji S.

Definicja 1.4. Niech p oznacza dowolny skończony ciąg segmentów bez powtórzeń. Ciąg p nazywamy **ścieżką** wtedy i tylko wtedy, gdy:

$$\forall_{0<i<|p|} \overrightarrow{p[i-1]} = \overleftarrow{p[i]}. \tag{1.5}$$

Jak wynika z definicji 1.4 każdy jednoelementowy ciąg segmentów jest ścieżką. Przez $\bowtie S$ będziemy oznaczać zbiór wyszystkich ścieżek, które można utworzyć z elementów zbioru $S \subset \mathbb{S}$.

Definicja 1.5. Dowolną ścieżkę p nazywamy **cyklem** wtedy i tylko wtedy, gdy:

$$\overleftarrow{p[0]} = \overrightarrow{p[|p|-1]}. \tag{1.6}$$

Definicja 1.6. Dowolny zbiór $S \subset \mathbb{S}$ nazywamy **zbiorem acyklicznym** wtedy i tylko wtedy, gdy $\bowtie S$ nie zawiera cykli.

Definicja 1.7. Zbiór segmentów $S \subset \mathbb{S}$ nazywamy **segmentacją** wtedy i tylko wtedy, gdy są spełnione następujące warunki:

1. acykliczność zbioru S,
2. unikalność identyfikatorów:

$$\forall_{a_0,a_1 \in \overleftrightarrow{S}} \#a_0 = \#a_1 \Longrightarrow a_0 = a_1, \tag{1.7}$$

$$\forall_{s_0,s_1 \in S} \#s_0 = \#s_1 \Longrightarrow s_0 = s_1, \tag{1.8}$$

3. zgodność czasu kotwic z przebiegiem ścieżek:

$$\forall_{p \in \bowtie S} \nexists_{i<j,k} \overrightarrow{p[j]} <^k \overleftarrow{p[i]}. \tag{1.9}$$

Definicja 1.8. Segmentację S nazywamy segmentacją **zakotwiczoną** wtedy i tylko wtedy, gdy:

$$\forall_{(i,\boldsymbol{t}) \in \overleftrightarrow{S}} \forall_k \boldsymbol{t}[k] \neq \varnothing. \tag{1.10}$$

Definicja 1.9. Relację **ograniczania czasowego** $\blacktriangleright^k \subset \mathbb{S}^n \times \mathbb{S}^n$, gdzie $k \in \{0, 1, \ldots, n-1\}$ definiujemy następująco:

$$s_0 \blacktriangleright^k s_1 \iff \overleftarrow{s_1} \geqslant^k \overleftarrow{s_0} \wedge \overrightarrow{s_1} \leqslant^k \overrightarrow{s_0}. \tag{1.11}$$

Definicja 1.10. Niech będzie dana segmentacja S. Relację **ograniczania ścieżkowego** $\rhd \subset S \times \bowtie S$ definiujemy następująco:

$$s \rhd p \iff \overleftarrow{p[0]} = \overleftarrow{s} \wedge \overrightarrow{p[|p|-1]} = \overrightarrow{s}. \tag{1.12}$$

Definicja 1.11. Niech będzie dana segmentacja S. Relację **ograniczania ścieżkowego** $\rhd \subset S \times S$ definiujemy następująco:

$$s_0 \rhd s_1 \iff \exists_{p \in \bowtie S, i} \; s_0 \rhd p \wedge p[i] = s_1. \tag{1.13}$$

Z każdym zbiorem $S \subset \mathbb{S}$ wiążemy ostry porządek częściowy $\overleftarrow{<}^S \subset S \times S$ taki, że:

$$s_0 \overleftarrow{<}^S s_1 \iff (\exists_k \overleftarrow{s_0} <^k \overleftarrow{s_1} \wedge \nexists_k \overleftarrow{s_0} >^k \overleftarrow{s_1}) \vee \exists_{p \in \bowtie S} \; p[0] = s_0 \wedge p[m] = s_1, \tag{1.14}$$

gdzie $m = |p| - 1 > 0$. Analogicznie określamy na $S \times S$ ostry porządek częściowy $\overrightarrow{<}^S$. Przez $\overleftarrow{\prec}^S$ oraz $\overrightarrow{\prec}^S$ oznaczamy relacje bycia poprzednikiem wynikające z porządków odpowiednio $\overleftarrow{<}^S$ oraz $\overrightarrow{<}^S$.

Definicja 1.12. Segmentację S nazywamy **segmentacją sekwencyjną** wtedy i tylko wtedy, gdy zachodzi:

$$\overleftarrow{<}^S = \overrightarrow{<}^S. \tag{1.15}$$

Jeśli S jest segmentacją sekwencyjną, to przez $S[i]$ będziemy oznaczać i-ty segment w porządku $\overleftarrow{<}^S$ ($\overrightarrow{<}^S$).

Definicja 1.13. Segmentację S nazywamy **rozłączną** w wymiarze k wtedy i tylko wtedy, gdy:

$$\nexists_{s_0, s_1 \in S} \; \overleftarrow{s_0} \leqslant^k \overleftarrow{s_1} <^k \overrightarrow{s_0}. \tag{1.16}$$

Jeśli S jest segmentacją sekwencyjną i rozłączną, to przez $S(t)$ oznaczamy segment $s \in S$ taki, że $\overleftarrow{s} \leqslant^k t <^k \overrightarrow{s}$ dla ustalonego k.

Definicja 1.14. Segmentację S nazywamy **segmentacją ciągłą** w wymiarze k wtedy i tylko wtedy, gdy istnieje segmentacja sekwencyjna $S_0 \subset S$ taka, że

$$\forall_{0<i<|S_0|} \; \overrightarrow{S_0[i-1]} \geqslant^k \overleftarrow{S_0[i]}. \tag{1.17}$$

Definicja 1.15. Segmentację ciągłą i rozłączną nazywamy **segmentacją prostą**.

Definicja 1.16. Segmentacją ramkową o rozmiarze $\boldsymbol{a} \in \mathbb{R}^n$ oraz kroku $\boldsymbol{b} \in \mathbb{R}^n$, gdzie $\forall_k \boldsymbol{b}[k] < \boldsymbol{a}[k]$ nazywamy segmentację sekwencyjną S spełniającą następujące warunki:

1. $\underset{s \in S}{\forall} \overrightarrow{s} - \overleftarrow{s} = \boldsymbol{a}$,
2. $\underset{s_0, s_1 \in S}{\forall} s_0 \overleftarrow{\prec}^S s_1 \Longrightarrow \overleftarrow{s_1} - \overleftarrow{s_0} = \boldsymbol{b}$.

Segment segmentacji ramkowej nazywamy **ramką**.

Definicja 1.17. Niech będzie dany ciąg segmentacji prostych S_i, gdzie $i \in \{0, 1, \ldots, n\}$ taki, że:

$$\underset{i}{\forall} \min(\overleftarrow{S_i}) = b \wedge \max(\overrightarrow{S_i}) = f, \tag{1.18}$$

dla ustalonych $b, f \in \mathbb{A}$. Segmentację S, gdzie:

$$S = \bigcup_{i=0,1,\ldots,n} S_i, \tag{1.19}$$

nazywamy **segmentacją kratową**.

Definicja 1.18. Segmentacja warstwowa jest to segmentacja kratowa, której ścieżki nie mają segmentów wspólnych. Ścieżkę segmentacji warstwowej biegnącą od kotwicy minimalnej do kotwicy maksymalnej nazywamy **warstwą**.

Definicja 1.19. Niech u oraz w będą warstwami należącymi do segmentacji S. Mówimy, że warstwa u jest **wyższa** od warstwy w (oraz jednocześnie, że warstwa w jest **niższa** od warstwy u) wtedy i tylko wtedy, gdy:

$$\underset{0 \leqslant i < |u|}{\forall} \underset{0 \leqslant j < |w|}{\exists} u[i] \rhd w[j] \tag{1.20}$$

Definicja 1.20. Niech będzie dana segmentacja S oraz dowolny zbiór E zawierający element $\varnothing$. **Anotacją** nazywamy parę (S, a), gdzie $a : \#S \mapsto E$ nazywamy **funkcją etykietującą**.

Niech $A = (S, a)$ będzie anotacją. Wartość $a(\#s)$, gdzie $s \in S$ nazywamy **etykietą segmentu** s oraz oznaczamy $\overline{s}$. Ponadto przez $\overline{A}$ oznaczamy zbiór etykiet wszystkich segmentów należących do S.

Definicja 1.21. Oznaczmy przez $A_0 = (S_0, a_0)$ i $A_1 = (S_1, a_1)$ dowolne anotacje. Mówimy, że anotacja A_0 jest **zawarta porządkowo** w anotacji A_1 wtedy i tylko wtedy, gdy istnieje odwzorowanie $\tau : \#S_0 \mapsto \#S_1$ takie, że dla każdych $s_{00}, s_{01} \in S_0$ i $s_{10}, s_{11} \in S_1$ jeśli $\tau(\#s_{00}) = \#s_{10}$ i $\tau(\#s_{01}) = \#s_{11}$, to zachodzą następujące warunki:

1. $a_0(\#s_{00}) = a_1(\#s_{10})$,
2. $a_0(\#s_{01}) = a_1(\#s_{11})$,
3. $s_{00} \overleftarrow{<} s_{01} \Rightarrow s_{10} \overleftarrow{<} s_{11}$,
4. $s_{00} \overrightarrow{<} s_{01} \Rightarrow s_{10} \overrightarrow{<} s_{11}$.

Przez $A_0 \overleftrightarrow{\subseteq} A_1$ będziemy oznaczać, że anotacja A_0 jest zawarta porządkowo w A_1.

Definicja 1.22. Odległością[1] na dowolnym zbiorze B nazywamy w niniejszej pracy dowolną funkcję $d : B \times B \mapsto \mathbb{R}$, która spełnia następujące warunki:

1. $d(b_0, b_1) \geqslant 0$,
2. $d(b_0, b_0) = 0$,
3. $d(b_0, b_1) = d(b_1, b_0)$.

Definicja 1.23. Niech E będzie dowolnym zbiorem zawierającym element $\varnothing$. Odległość określoną na zbiorze $\mathbb{S} \times E$ nazywamy **cząstkową odległością anotacyjną**.

W przypadku, gdy E jest zbiorem liczbowym, do najczęściej stosowanych cząstowych odległości anotacyjnych należą:

$$d_1(b_0, b_1) = |b_0[1] - b_1[1]|, \tag{1.21}$$

oraz

$$d_2(b_0, b_1) = (b_0[1] - b_1[1])^2. \tag{1.22}$$

Definicja 1.24. Odległością anotacyjną nazywamy odległość określoną na dowolnym zbiorze anotacji.

Przyjmijmy, że są dane anotacje $A_0 = (S, a_0)$ oraz $A_1 = (S, a_1)$, gdzie S jest pewną segmentacją sekwencyjną. Wartość odległości anotacyjnej SSE (*Sum of Squared Error*) dla pary A_0, A_1 jest określamy następująco:

$$SSE(A_0, A_1) = \sum_{i=0}^{n-1} d_2\left((S_0[i], \overline{S_0[i]}), (S_1[i], \overline{S_1[i]})\right), \tag{1.23}$$

gdzie dla $(S_j, a_j) = A_j$ dla $j \in \{0, 1\}$. Podobnie określamy odległości anotacyjne MSE (*Mean Squared Error*):

$$MSE(A_0, A_1) = \frac{1}{n} SSE(A_0, A_1), \tag{1.24}$$

gdzie $n = |S_0| = |S_1|$ oraz RMSE (*Root Mean Squared Error*):

$$RMSE(A_0, A_1) = \sqrt{MSE(A_0, A_1)}. \tag{1.25}$$

Tadeusiewicz i Lula (2000, 556) podają szereg dalszych miar (nie)podobieństwa szeregów czasowych, na podstawie których można konstruować definicje odległości anotacyjnych.

Odległości anotacyjne dla anotacji sekwencyjnych o różnej liczbie segmentów wyznaczamy z zastosowaniem metod LTW (*Linear Time Warping*) lub DTW (*Dynamic Time Warping*) z cząstkową odległością anotacyjną jako funkcją kosztu (por. np. Gold 1999, 326).

[1] W publikacjach matematycznych funkcja określona zgodnie z definicją 1.22 nazywana jest pseudosemimetryką. Pseudosemimetrykę od metryki odróżnia zastąpienie warunku $d(b_0, b_1) = 0 \iff b_0 = b_1$ warunkiem 2 (pseudo-) oraz brak warunku nierówności trójkąta (semi-).

1.2 Analiza anotacyjna

Definicja 1.25. Algorytmem analizy anotacyjnej nazywamy taki algorytm $\mathcal{A}$, który dla sygnału x oraz anotacji A_0 tworzy anotację A_1, co zapisujemy:

$$A_1 = \mathcal{A}(x, A_0). \tag{1.26}$$

Proces wykonywania algorytmu analizy anotacyjnej nazywamy **analizą anotacyjną**. Dla uproszczenia przyjmiemy oznaczenie:

$$\mathcal{A}(x) = \mathcal{A}(x, (\emptyset, \emptyset)), \tag{1.27}$$

gdzie $\emptyset$ oznacza zbiór pusty.

W dalszej części sekcji definiujemy szereg przymiotników charakteryzujących algorytmy analizy anotacyjnej. Każdy z wprowadzonych przymiotników będzie stosowany nie tylko w odniesieniu do algorytmu ale i powiązanych z nim: segmentacji, anotacji oraz analizy anotacyjnej.

Definicja 1.26. Algorytm analizy anotacyjnej $\mathcal{A}$ nazywamy **kotwiczącym** wtedy i tylko wtedy, gdy:

$$\overleftrightarrow{S_1} \not\subset \overleftrightarrow{S_0}, \tag{1.28}$$

gdzie $(S_1, a_1) = \mathcal{A}(x, (S_0, a_0))$.

Definicja 1.27. Dla danego sygnału x **oknem segmentu** s nazywamy sygnał będący wartością funkcji $\boxplus$ określonej następująco:

$$\boxplus(x, s)[i] = x[i + \overleftarrow{s}]w^N[i], \tag{1.29}$$

gdzie w^N oznacza dowolne skończone okno sygnałowe (np. okno prostokątne lub okno Hamminga).

Definicja 1.28. Algorytm analizy anotacyjnej nazywamy **sygnałowym** i mówimy, że należy do **sygnałowego poziomu analizy**, gdy dla każdej anotacji wynikowej (S_1, a_1): 1) przyjmuje się, że sygnał wejściowy w granicach okna każdego segmentu $s_1 \in S_1$ jest realizacją stochastycznego sygnału stacjonarnego, 2) przeciwdziedzina a_1 jest związana ze skalą interwałową, 3) wszystkie kotwice ze zbioru $\overleftrightarrow{S_1}$ mają czasy określone.

Definicja 1.29. Algorytm analizy anotacyjnej nazywamy **fonetycznym** i mówimy, że należy do **fonetycznego poziomu analizy**, gdy dla każdej anotacji wynkowej (S_1, a_1) przyjmuje się, że: 1) sygnał w granicach okna każdego segmentu $s_1 \in S_1$ jest realizacją niestacjonarnego sygnału stochastycznego, 2) przeciwdziedzina funkcji a_1 jest związana z przynajmniej jedną skalą interwałową.

Definicja 1.30. Algorytm analizy anotacyjnej nazywamy **fonologicznym** i mówimy, że należy do **fonologicznego poziomu analizy**, gdy dla każdej anotacji wynikowej (S_1, a_1) przyjmuje się, że: 1) sygnał w granicach okna każdego segmentu $s_1 \in S_1$ jest realizacją niestacjonarnego sygnału stochastycznego, 2) przeciwdziedzina a_1 jest związana wyłącznie ze skalami nominalnymi.

Wygodnie jest ustawić poziomy w porządku od sygnałowego (najniższego) do fonologicznego (najwyższego spośród zdefiniowanych w bieżącej sekcji).

Definicja 1.31. Algorytm analizy anotacyjnej $\mathcal{A}$ nazywamy **indukcyjnym**, gdy anotacja wejściowa należy do niższego poziomu, niż anotacja wyjściowa.

Definicja 1.32. Algorytm analizy nazywamy **dedukcyjnym**, gdy anotacja wejściowa należy do wyższego poziomu niż anotacja wyjściowa.

Definicja 1.33. Algorytm analizy anotacyjnej $\mathcal{A}$ nazywamy **segmentalnym**, gdy istnieje funkcja g taka, że dla każdego $(S_1, a_1) = \mathcal{A}(x, (S_0, a_0))$ zachodzi:

$$\underset{s_1 \in S_1}{\forall} \overline{s_1} = g(\boxplus(x, s_1)). \tag{1.30}$$

Etykieta segmentu s otrzymana w wyniku segmentalnej analizy anotacyjnej sygnału x nie zależy od przebiegu sygnału x poza przedziałem czasowym $[\overleftarrow{s}; \overrightarrow{s})$.

Definicja 1.34. Jeśli algorytm analizy anotacyjnej $\mathcal{A}$ nie jest segmentalny, to algorytm $\mathcal{A}$ nazywamy **suprasegmentalnym**.

Przedstawione definicje analizy segmentalnej oraz suprasegmentalnej nawiązują do pracy Lehiste (1976).

Algorytm analizy anotacyjnej nazywamy **subiektywnym**, gdy odnosi się do *wrażeń słuchowych* jednej lub kilku osób. Algorytm analizy anotacyjnej nazywamy **intersubiektywnym**, gdy jest oparty na *wnioskowaniu statystycznym* z wyników wielokrotnego wykonania algorytmu subiektywnego na tych samych danych wejściowych. Algorytm analizy anotacyjnej nazywamy **obiektywnym**, gdy wszystkie jego kroki można wykonać za pomocą *urządzenia obliczeniowego*.

Definicja 1.35. Algorytm analizy anotacyjnej $\mathcal{A}$ nazywamy **strumieniowym**, gdy spełnia następujący warunek:

$$\underset{x_0, x_1, A_0}{\forall} \mathcal{A}(x_0 \oplus x_1, A_0) = \mathcal{A}(x_1, \mathcal{A}(x_0, A_0)), \tag{1.31}$$

gdzie $\oplus$ jest oparatorem konkatenacji sygnałów.

Niech będzie dany (potencjalnie) nieskończony sygnał y oraz dowolny ciąg sygnałów (x_0, x_1, ...) taki, że:

$$y = x_0 \oplus x_1 \oplus \ldots \tag{1.32}$$

Jeśli dany jest strumieniowy algorytm analizy anotacyjnej $\mathcal{A}$, to anotację $\mathcal{A}(y)$ można otrzymać korzystając z następującej rekurencji:

1. warunek początkowy:
$$A_0 = (\emptyset, \emptyset), \tag{1.33}$$
2. krok rekurencyjny:
$$A_{i+1} = \mathcal{A}(x_i, A_i). \tag{1.34}$$

Strumieniowe algorytmy analizy anotacyjnej są wykorzystywane m.in. w *systemach czasu rzeczywistego*.

Definicja 1.36. Algorytmem **syntezy anotacyjnej** nazywamy algorytm tworzenia sygnału x na podstawie anotacji A zgodnie z algorytmem $\mathcal{A}'$, co zapisujemy jako:

$$x = \mathcal{A}'(A). \tag{1.35}$$

Jeśli anotacja A we wzorze 1.35 powstała w wyniku analizy anotacyjnej, to algorytm $\mathcal{A}'$ nazywamy algorytmem **resyntezy anotacyjnej**.

Definicja 1.37. Algorytm analizy anotacyjnej $\mathcal{A}$ nazywamy **odwracalnym** na zbiorze sygnałów $\{x_0, x_1, \ldots, x_n\}$ przy zadanej odległości anotacyjnej d oraz zadanym algorytmie analizy anotacyjnej $\mathcal{D}$ wtedy i tylko wtedy, gdy jest znany algorytm syntezy anotacyjnej $\mathcal{A}'$ spełniający warunek:

$$\frac{1}{N}\sum_{i=0}^{n} d(\mathcal{D}(x_i), \mathcal{D}(y_i)) < \epsilon, \tag{1.36}$$

gdzie

$$y_i = \mathcal{A}'(\mathcal{A}(x_i)), \tag{1.37}$$

N jest sumą liczb segmentów w anotacjach $\mathcal{D}(x_i)$ po $i \in \{0, 1, \ldots, n\}$ a ϵ jest przyjętym progiem.

1.3 System fonologiczny

Definicja 1.38. Niech będzie dany zbiór sygnałów $\{x_0, x_1, \ldots, x_n\}$ oraz zbiór segmentacji sekwencyjnych $\{S_0, S_1, \ldots, S_n\}$. Algorytm analizy fonologicznej $\mathcal{F}$ nazywamy **systemem fonologicznym** pary wyżej wymienionych ciągów wtedy i tylko wtedy, gdy są spełnione następujące warunki:

1. odwracalność: algorytm $\mathcal{F}$ jest odwracalny w sensie definicji 1.37,

2. zachowanie segmentacji:
$$\underset{i}{\forall}\,\underset{a}{\exists}\,\mathcal{F}(x_i, (S_i, \emptyset)) = (S_i, a), \tag{1.38}$$

3. minimalność (lokalna): nie istnieje zbiór etykiet E_1 oraz algorytm $\mathcal{F}_1$ spełniający warunki 1 i 2 takie, że
$$\underset{i}{\forall}\,\overline{\mathcal{F}_1(x_i)} = e(\overline{\mathcal{F}(x_i)}), \tag{1.39}$$
gdzie $e : E \mapsto E_1$, przy czym $E_1 \subset E = \bigcup_i \overline{\mathcal{F}(x_i)}$.

Definicja 1.38 nawiązuje do strukturalistycznych definicji systemów fonologicznych oraz ich formalizacji (por. np. Bloch 1948; Batóg 1967, 1994). Do najbardziej zaawansowanych prób algorytmizacji problemu konstrukcji systemu fonologicznego można zaliczyć prace Batoga (1994) oraz Boersmy (1998) oparte odpowiednio na logice formalnej oraz teorii optymalności (*Optimality Theory*). Niestety, algorytmy Batoga oraz Boersmy są trudne do zastosowania m.in. ze względu na wysoką złożoność obliczeniową. Częściowym rozwiązaniem problemu algorytmizacji konstrukcji systemu fonologicznego są heurystyki opisane w dalszej części niniejszej sekcji (Jassem 1956, 2007).

Definicja 1.39. Przyjmijmy oznaczenia jak w definicji 1.38. Określmy zbiór Ω następująco:

$$\Omega = \bigcup_i \{(i, s) : s \in S_i\}. \tag{1.40}$$

Przez **kryterium dystynktywności** rozumiemy dowolną relację równoważności na zbiorze Ω.

Przyjmuje się, że odwzorowanie ilorazowe związane z kryterium dystynktywności jest prototypem systemu fonologicznego. Określmy zbiory pomocnicze:

$$\Omega_x = \{(i, s, S_i, x_i) : (i, s) \in \Omega\}, \tag{1.41}$$

$$\Omega_a = \{(i, s, S_i, a_i) : (i, s) \in \Omega\}, \tag{1.42}$$

gdzie $(S_i, a_i) = \mathcal{A}(x_i, (S_i, \emptyset))$ dla ustalonego algorytmu analizy fonologicznej $\mathcal{A}$. Zauważmy, że istnieją bijekcje pomiędzy zbiorem Ω a zbiorami Ω_x oraz Ω_a. Na podstawie tych bijekcji dowolną relację równoważności na Ω_x oraz Ω_a można przenieść na zbiór Ω. W związku z powyższym relacje równoważności na zbiorach Ω_x oraz Ω_a będziemy także nazywać **kryteriami dystynktywności**.

Kryteria dystynktywności na zbiorze Ω_x nazywamy **fonetycznymi** kryteriami dystynktywności. Kryteria dystynktywności na zbiorze Ω_a nazywamy **fonologicznymi** kryteriami dystynktywności.

Podobnie jak w przypadku algorytmów analizy anotacyjnej wyróżniamy **intersubiektywne** kryteria dystynktywności oraz **obiektywne** kryteria dystynktywności.

Jassem (2007) proponuje, by konstruując system fonologiczny wziąć pod uwagę następujące kryteria dystynktywności[2]:

- **percepcyjne**: intersubiektywne fonetyczne kryterium dystynktywności konstruowane w oparciu o *testy psychoakustyczne*,
- **statystyczne**: obiektywne fonetyczne kryterium dystynktywności konstruowane w oparciu o algorytmy grupowania pojęciowego (por. np. Krzyśko i inni 2008),
- **pragmatyczne**: intersubiektywne fonologiczne kryterium dystynktywności konstruowane w oparciu o *testy psycholingwistyczne*,
- **dystrybucyjne**: obiektywne fonologiczne kryterium dystynktywności konstruowane w oparciu o postulaty Blocha (1948).

Mówimy, że algorytm $\mathcal{A}$ **spełnia kryterium dystynktywności** $\equiv$ na zbiorze Ω wtedy i tylko wtedy, gdy:

$$\underset{i,j}{\forall}\ \overline{s_i} = \overline{s_j} \Rightarrow \omega_i \equiv \omega_j, \tag{1.43}$$

gdzie $s_i \in S_i$ oraz $(S_i, a_i) = \mathcal{A}(x_i, (S_i, \emptyset))$.

[2] W wypunktowaniu podajemy własne, fonetyczno–informatyczne interpretacje kryteriów opisannych przez Jassema.

Definicja 1.40. Jeśli segmentalny system fonologiczny spełnia fonetyczne kryteria dystynktywności, to etykiety anotacji wyjściowych nazywamy **głoskami**.

Definicja 1.41. Jeśli segmentalny system fonologiczny spełnia fonologiczne kryteria dystynktywności, to etykiety anotacji wyjściowych nazywamy **fonemami**.

Definicja 1.42. Niech będą ustalone: system fonologiczny $\mathcal{F}$ oraz dowolna etykieta e występująca w anotacjach wyjściowych systemu $\mathcal{F}$. **Cechą sygnałową** nazywamy dowolną weryfikowalną statystycznie hipotezę, która dotyczy przebiegu sygnału w granicach lub w bezpośrednim sąsiedztwie segmentów mających etykietę e.

Definicja 1.43. Niech będą ustalone: system fonologiczny $\mathcal{F}$ oraz dowolna etykieta e występująca w anotacjach wyjściowych systemu $\mathcal{F}$. Ponadto niech będzie ustalony segmentalny sygnałowy algorytm analizy anotacyjnej $\mathcal{A}$. **Monosegmentalną cechą sygnałową** nazywamy taką cechę sygnałową, która jest weryfikowana na zbiorze etykiet anotacji otrzymanych przy użyciu algorytmu $\mathcal{A}$ z sygnałów segmentów mających etykietę e w anotacjach otrzymanych przy użyciu algorytmu $\mathcal{F}$.

Definicja 1.44. Niech będą dane dwie anotacje otrzymane za pomocą systemów fonologicznych: segmentalna $A_0 = (S_0, a_0)$ oraz suprasegmentalna $A_1 = (S_1, a_1)$, gdzie S_0 jest segmentacją prostą a S_1 jest segmentacją kratową. Jeśli spełnione są warunki:

1. $\overrightarrow{\overleftarrow{A_1}} \subset \overrightarrow{\overleftarrow{A_0}}$,
2. $\max(\overrightarrow{S_0}) = \max(\overrightarrow{S_1})$,
3. $\min(\overleftarrow{S_0}) = \min(\overleftarrow{S_1})$,

to anotację $(S_0 \cup S_1, a_0 \cup a_1)$ nazywamy **komunikatem językowym**.

Jak wynika z definicji 1.17 komunikat językowy jest anotacją kratową. Dowolny ciąg komunikatów językowych nadanych kolejno po sobie będziemy nazywać **wypowiedzią**.

1.4 Korpus i leksykon

Definicja 1.45. Niech będą ustalone anotacja fonologiczna (S, a) oraz ścieżka $p \in \bowtie S$. **Tekstem** ścieżki p nazywamy ciąg etykiet:

$$(\overline{p[0]}, \overline{p[1]}, \ldots, \overline{p[|p|-1]}), \tag{1.44}$$

który oznaczamy przez $\overline{p}$.

Tekstem fonologicznym nazywamy tekst ścieżki powstałej w wyniku działania dowolnego systemu fonologicznego. Tekst fonologiczny złożony z głosek nazywamy **tekstem fonetycznym**. Tekst fonologiczny złożony z fonemów nazywamy **tekstem fonematycznym**. W zależności od rodzaju systemu fonologicznego tekst fonologiczny nazywamy **segmentalnym** lub **suprasegmentalnym**. Jak wynika z przyjętej terminologii, komunikat jest wzbogaconą synchroniczną reprezentacją tekstu fonologicznego segmentalnego oraz tekstu fonologicznego suprasegmentalnego.

Sygnałem ortograficznym nazywamy w niniejszej pracy sygnał cyfrowy, którego skalę wartości utożsamia się z punktami kodowymi alfabetów naturalnych[3] zamieszczonych w tablicy Unicode (2008). Sygnał ortograficzny jest naturalną (będącą wytworem kulturowym) reprezentacją komunikatu językowego. Próbki sygnału ortograficznego nazywamy **znakami**.

Definicja 1.46. Korpusem mowy nazywamy zbiór trójek uporządkowanych (x, o, A), gdzie x jest cyfrowym sygnałem mowy, o jest sygnałem ortograficznym oraz A jest anotacją zakotwiczoną w sygnałach x oraz o.

W szczególności język naturalny utożsamiamy się z (nieskończonym) korpusem mowy obejmującym wszystkie dopuszczalne w ustalonym języku naturalnym trójki postaci (x, o, K), gdzie K jest komunikatem językowym.

Definicja 1.47. Przez **leks** rozumiemy w niniejszej pracy parę uporządkowaną (o, K), gdzie o jest sygnałem ortograficznym a $K = (S, a)$ jest komunikatem językowym takim, że $\min(\overleftarrow{S}) = 0$ oraz $\max(\overrightarrow{S}) = |o|$.

Leks określa związek komunikatu językowego z sygnałem ortograficznym. Jeśli (o, K) jest leksem, to sygnał o nazywamy **wyrazem**.

Mówimy, że leks $L = (o, K_0)$ **występuje w komunikacie** K_1 wtedy i tylko wtedy, gdy zachodzi $K_0 \overrightarrow{\subseteq} K_1$ (por. definicja 1.21 na stronie 9).

Definicja 1.48. Leksykonem korpusu mowy R nazywamy taki zbiór leksów M, że dla każdego $(x, o, A) \in R$ istnieje ciąg leksów $((o_0, K_0), (o_1, K_1), \ldots, (o_n, K_n))$ ze zbioru M taki, że tekst segmentalny anotacji A jest równy konkatenacji tekstów segmentalnych ciągu komunikatów $(K_0, K_1, \ldots, K_n)$.

[3] W definicji sygnału ortograficznego wyłączamy alfabety sztuczne, np. IPA (2005).

ROZDZIAŁ 2

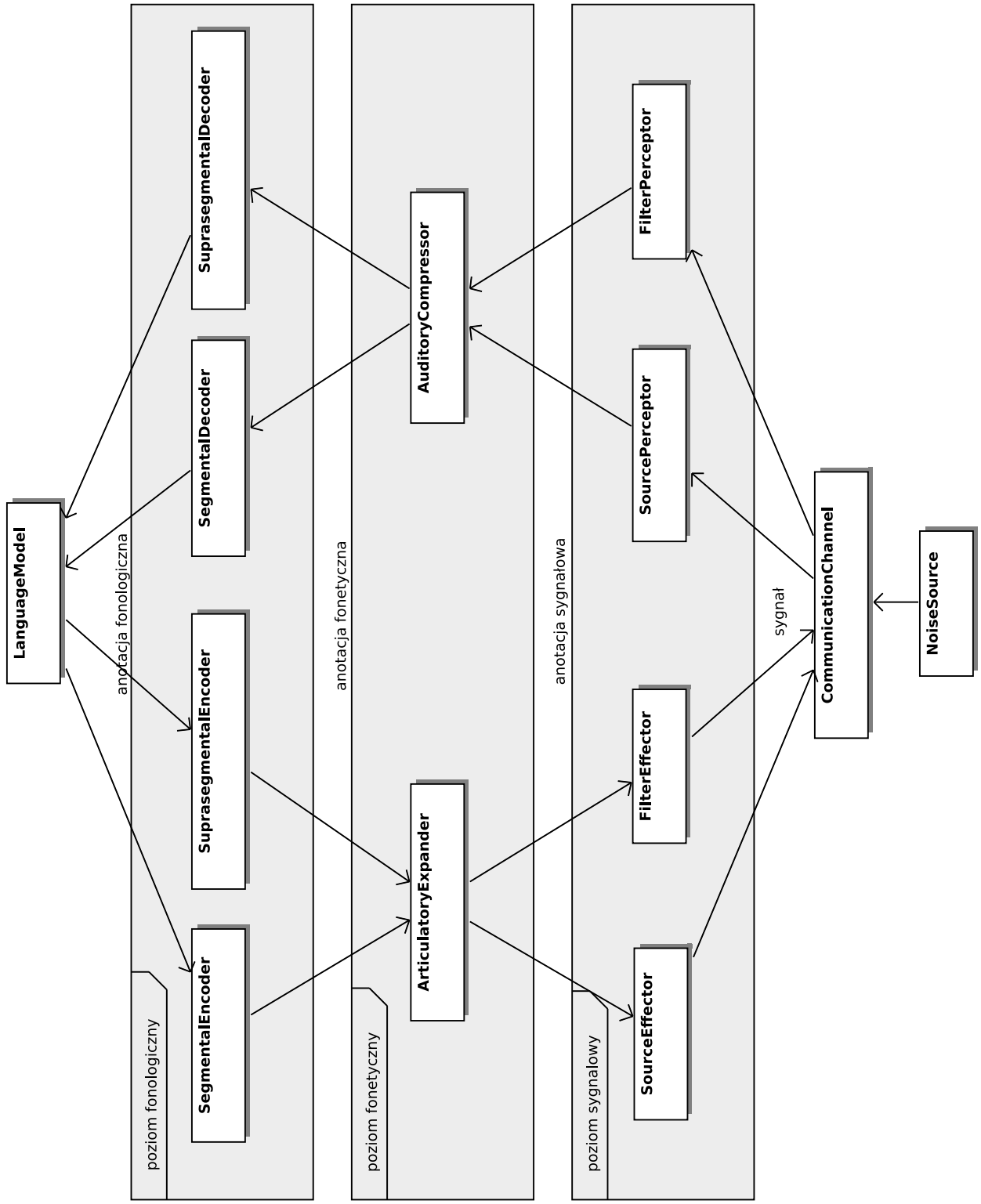
Komunikacja głosowa

2.1 Głosowy system komunikacyjny

Procesem mówienia nazywamy proces przekazywania komunikatów językowych w postaci sygnału akustycznego między człowiekiem a człowiekiem lub maszyną (por. np. Frydrychowicz 1999, 19). Sygnał reprezentujący komunikat językowy w procesie mówienia nazywamy **sygnałem mowy**.

System realizujący proces mówienia nazywamy **głosowym systemem komunikacyjnym**. Na rycinie 2.1 na stronie 18 przedstawiono proponowany model głosowego systemu komunikacyjnego w postaci diagramu klas UML. (Strzałki asocjacji wskazują jednocześnie kierunek przepływu informacji.) Proponowany model został oparty na klasycznym modelu Shannona (1948) z uwzględnieniem szeregu późniejszych prac, które opublikowali: Jassem (1974, 34), Hirst i inni (2000, 53), Fujisaki (2004), Levinson (2005, 233) oraz Fastl i Zwicker (2007, 361). Wyróżnikami proponowanego modelu są: 1) grupowanie klas w pakietach reprezentujących poziomy analizy lingwistycznej, 2) dwutorowość przepływu informacji na poziomie sygnałowym oraz fonologicznym oraz 3) przyjęcie anotacyjnej reprezentacji wiedzy dla informacji przepływających między poziomami analizy.

Pakiety proponowanego modelu są uporządkowane od sygnałowego (najniższego) do fonologicznego (najwyższego) zgodnie z pionową lokalizacją na diagramie. **Poziom sygnałowy** to pakiet klas, których instancje wykonują obustronną konwersję między sygnałem a anotacją sygnałową. **Poziom fonetyczny** to pakiet klas, których instancje wykonują obustronną konwersję między anotacją sygnałową a anotacją fonetyczną. Na poziomie fonetycznym są wprowadzane (podczas syntezy) lub eliminowane (podczas analizy) te własności sygnału mowy, które wynikają z cech osobniczych aparatu produkcji mowy (por. Hirst i inni 2000, 56). Poziom fonetyczny jest (w znacznym stopniu) językowo uniwersalny. **Poziom fonologiczny** to pakiet klas, których instancje wykonują obustronną konwersję między anotacją fonetyczną a anotacją fonologiczną. Poziom fonologiczny jest językowo specyficzny. W dalszej części sekcji omówiono kolejno klasy proponowanego modelu komunikacji głosowej. Przyjęto kolejność zgodną z kierunkiem przepływu informacji zaczynając od klasy LanguageModel umieszczonej na samej górze ryciny 2.1.



Rycina 2.1: Głosowy system komunikacyjny. Diagram klas UML.

Klasa LanguageModel (model językowy) reprezentuje źródło i jednocześnie cel komunikatów językowych. Zakłada się, że komunikaty pojawiające się na wyjściu obiektów klasy LanguageModel spełniają szereg dodatkowych (nieobjętych w proponowanym modelu) ograniczeń. Usytuowanie klasy LanguageModel w kontekście tradycyjnie rozpatrywanych ponadfonologicznych poziomów analizy mowy i języka (tj. morfo-syntaktycznego, semantycznego i pragmatycznego) wykracza poza zakres proponowanego modelu.

Obiekty klasy SegmentalEncoder oraz SuprasegmentalEncoder tworzą anotacje fonetyczne na podstawie segmentalnych oraz suprasegmentalnych ścieżek komunikatów językowych. Obiekty klasy ArticulatoryExpander tworzą anotacje sygnałowe na podstawie anotacji fonetycznych. Obiekty klas SourceEffector oraz FilterEffector tworzą sygnał mowy na podstawie anotacji sygnałowych w oparciu o *model źródło-filtr* (por. Fant 1960; Levinson 2005).

Podsytem złożony z klas SegmentalEncoder, SuprasegmentalEncoder, ArticulatoryExpander, SourceEffector oraz FilterEffector nazywamy **podsystemem nadawczym**. Wyróżniamy podsystemy nadawcze **naturalne** (człowiek) oraz **sztuczne** (maszyna). Naturalny podsystem nadawczy nazywamy **mówcą**.

Przyjmujemy[1], że w przypadku mówcy etykiety anotacji fonetycznej reprezentują *komendy motoryczne* narządów artykulacyjnych (por. Fujisaki 2000), natomiast etykiety anotacji sygnałowej reprezentują lokalizacje narządów artykulacyjnych. Do klasy SourceEffector u mówcy należą *fałdy głosowe* (Hess 1983, 38-62) oraz zwężenia *toru głosowego* wywołujące turbulentny przepływ powietrza (Stevens 1998, 127-130).

Prawieokresowy sygnał generowany przez fałdy głosowe jest nazywany **tonem krtaniowym**. **Ciśnienie podkrtaniowe** jest to różnica pomiedzy ciśnieniem potwierza w płucach a ciśnieniem atmosferycznym. Obszar między fałdami głosowymi nazywany jest **głośnią**. Źródłem tonu krtaniowego jest **fonacja**, tj. drganie fałdów głosowych wywoływane ciśnieniem podkrtaniowym. W trakcie fonacji następują cykliczne zmiany powierzchni głośni powodowane przez zjawisko aerodynamiczne znane jako efekt Bernoulliego (Hess 1983, 38). Wyróżnia się dwie fazy cyklu fonacji: 1) **fazę głośni zamkniętej** oraz 2) **fazę głośni otwartej**. W fazie głośni zamkniętej następuje wzrost ciśnienia podkrtaniowego prowadzący, przy osiągnięciu pewnej wartości progowej, do otwarcia głośni. W fazie głośni otwartej przepływ powietrza między fałdami głosowymi wytwarza podciśnienie, które prowadzi do zamknięcia głośni. Zjawisko fonacji opisali m.in. Hess (1983, 38-50) oraz Stevens (1998, 55-97).

Ze względu na przebieg oraz częstość cykli fonacji wyróżnia się trzy **rodzaje fonacji**: **modalną**, **chrypliwą** oraz **falsetową**. Jeśli dla określonego głosu oznaczymy przez μ wartość średnią oraz przez σ odchylenie standardowe liczby cykli fonacji w jednostce czasu, fonacja modalna z dobrym przybliżeniem zawrze się w przedziale $[\mu - 2\sigma; \mu + 2\sigma]$ cykli w jednostce czasu. Przedziały fonacji chrypliwej oraz falsetowej znajdują się odpowiednio powyżej oraz poniżej przedziału fonacji modalnej.

Jedną z technik obrazowania fonacji w fałdach głosowych jest elektroglotografia (EGG) (Fourcin 1974). W trakcie fonacji następuje prawieokresowa zmiana powierzchni zetknięcia fałdów głosowych a tym samym oporności elektrycznej krtani (Marasek 1997). Urządzenie elektroniczne o nazwie elektroglotograf przetwarza zmiany oporności elektrycznej krtani na sygnał elektryczny (sygnał EGG).

Do klasy FilterEffector u mówcy należy *tor głosowy* (por. np. Stevens 1998, 127-130). Zgodnie z modelem Fanta (1960) tor głosowy odpowiada zespołowi połączonych ze sobą rezonatorów oraz antyrezonatorów. Częstotliwości rezonansowe toru głosowego są nazywane **formantami**. Wartości kolejnych formantów są oznaczane przez: F_1, F_2, itd. Częstotliwości

[1]Ze względu na szereg niejasności związanych ze strukturą oraz funkcjonowaniem wyższych warstw kory mózgowej, w pracy prezentujemy interpretację biologiczną wyłącznie wybranych, niskopoziomowych elementów modelu komunikacji głosowej. Zaznaczamy przy tym, że proponowany model nie aspiruje do miana modelu kognitywnego.

antyrezonansowe toru głosowego są nazywane **antyformantami**. Źródłem antyformantów w torze głosowym jest komora nosowa.

W sztucznych podukładach nadawczych (np. w formantowych lub statystycznych systemach syntezy mowy) etykiety anotacji fonetycznej reprezentują wartości docelowe lub statystyki przebiegu parametrów modelu, natomiast etykiety anotacji sygnałowej reprezentują wektory wszystkich zmieniających się w czasie parametrów modelu (por. np. Allen i inni 1987; Taylor 2009, 435). Do klasy SourceEffector w sztucznych podukładach nadawczych należą m. in.: model źródła sygnału w formantowej syntezie mowy (por. Klatt i Klatt 1990) oraz model źródła sygnału w statystycznej syntezie mowy (por. Maia i inni 2007). Do klasy FilterEffector w sztucznych podukładach nadawczych należą m.in.: rezonatory formantowego syntezatora mowy (por. Allen i inni 1987, 123) oraz statystyczne modele widma sygnału mowy (por. Yoshimura i inni 1999).

CommunicationChannel reprezentuje **kanał komunikacyjny**, tj. obiekt lub ośrodek fizyczny, z którym lub w obrębie którego sygnały fizyczne zmieniają lokalizację czasowo–przestrzenną. Do klasy CommunicationChannel należą m.in.: powietrze, przewodniki elektryczne, sieci telekomunikacyjne oraz nośniki danych. Zgodnie z klasycznym modelem systemu komunikacyjnego Shannona (1948), sygnał podczas pobytu w kanale komunikacyjnym może ulec niepożądanej modyfikacji (zakłóceniu). Klasa NoiseSource reprezentuje **źródła zakłóceń**, tj. obiekty, które modyfikują sygnał w trakcie pobytu w kanale komunikacyjnym. Do klasy NoiseSource należą m.in.: fizyczne uszkodzenia nośników danych, sygnały akustyczne towarzyszące pracy urządzeń mechanicznych oraz podukłady nadawcze innych procesów komunikacji głosowej.

Obiekty klas SourcePerceptor oraz FilterPerceptor zapisują w etykietach anotacji sygnałowej reprezentacje odpowiednio częstotliwości podstawowej oraz obwiedni widma. Obiekty klasy AuditoryCompressor zapisują w etykietach anotacji fonetycznej reprezentację niepodzielnych przebiegów monotonicznych lub ekstremów lokalnych w reprezentacji sygnałowej. Obiekty klasy SegmentalDecoder tworzą, na podstawie anotacji fonetycznej, prostą lub kratową segmentalną anotację fonologiczną. Obiekty klasy SuprasegmentalDecoder tworzą, na podstawie anotacji fonetycznej, prostą lub kratową suprasegmentalną anotację fonologiczną. Zakłada się, że klasy SegmentalDecoder oraz SuprasegmentalDecoder działają w oparciu o systemy fonologiczne.

Podsystem złożony z klas SourcePerceptor, FilterPerceptor, AuditoryCompressor, SegmentalDecoder oraz SuprasegmentalDecoder nazywamy **podsystemem odbiorczym**. Analogicznie jak w przypadku systemów nadawczych, wyróżniamy podsystemy odbiorcze **naturalne** (człowiek) oraz **sztuczne** (maszyna). Naturalny podsystem odbiorczy nazywamy **słuchaczem**.

Odstęp sygnału od zakłóceń[2] (SNR, *Signal–to–Noise Ratio*) jest to stosunek mocy sygnału pochodzącego z podsystemu nadawczego do mocy sygnału pochodzącego ze źródła zakłóceń mierzony na wejściu podsystemu odbiorczego.

Przyjmujemy, że etykiety anotacji sygnałowej u słuchacza reprezentują stany niższych warstw *kory słuchowej* (*auditory cortex*), w których następuje wstępne przetworzenie informacji pochodzących z *peryferyjnego układu słuchowego* (*peripherial auditory system*) (por.

[2] Ze względu na ścisłe rozumienie szumu jako własności sygnału stochastycznego (por. strona **??**) w bieżącej pracy używamy terminu „odstęp sygnału od zakłóceń" zamiast częściej spotykanego „odstęp sygnału od szumu".

np. Matthews 2000, 464). Przyjmujemy, że etykiety anotacji fonetycznej u słuchacza reprezentują *złożone wrażenia słuchowe* (*complex auditory sensations*) powstające w wyższych warstwach kory słuchowej (por. np. Fastl i Zwicker 2007, 361). Do klasy SourcePerceptor oraz FilterPerceptor należą *ucho środkowe* (*inner ear*) oraz specjalizowane struktury *przodomózgowia* (*forebrain*) (por. np. Gold 1999, 214, 228).

W sztucznych podukładach odbiorczych etykiety anotacji sygnałowych zawierają zdekorelowane reprezentacje obwiedni widma oraz wysokości tonu. W sztucznych podukładach odbiorczych anotacje fonetyczne[3] reprezentują proste (jedno lub dwukierunkowe) przebiegi parametrów akustycznych (np. częstotliwość podstawową oraz częstotoliwości formantów). Realizacjami klasy FilterPerceptor w sztucznych podukładach odbiorczych są m. in.: algorytmy MFCC (Mermelstein 1976), PLP (Hermansky 1990) oraz PMVDR (Yapanel i Hansen 2008). Realizacje klas SourcePerceptor, AuditoryCompressor oraz SuprasegmentalDecoder w sztucznych podukładach odbiorczych opisano w niniejszej pracy w rozdziałach odpowiednio 3, 4 oraz 5. Przykładami instancji SegmentalDecoder jest statystyczny układ rozpoznawania mowy oparty na HMM (por. np. Huang i inni 2001, 377).

2.2 Cechy tonalne i intonacja

Obrazy fonacji (definicja fonacji na stronie 19) w sygnale mowy nazywamy **cechami tonalnymi** (Fujisaki 2004). **Anotacją tonalną** nazywamy dowolną anotację, która reprezentuje cechy tonalne. Analogicznie przymiotnika „tonalny" używamy z wcześniej wprowadzonymi pojęciami takimi jak analiza, algorytm oraz system fonologiczny.

Cechy tonalne nazywamy **specyficznymi** jeśli ich zinterpretowanie wymaga znajomości języka naturalnego, którym posłużył się mówca (Gussenhoven 2002). Te cechy tonalne, które nie są specyficzne nazywamy cechami tonalnymi **uniwersalnymi**. Interpretacja uniwersalnych cech tonalnych nie wymaga znajomości języka, natomiast wymaga znajomości cech osobniczych mówcy, np. zakresu F_0 (Jassem i Kudela-Dobrogowska 1980).

Nie wszystkie cechy tonalne sygnału mowy wynikają z etykiet suprasegmentalnej anotacji fonologicznej danej na wejściu obiektu klasy SuprasegmentalEncoder (por. rycina 2.2 nas stronie 23). Cechy tonalne sygnału mowy, które nie wynikają z etykiet suprasegmentalnych nazywamy **wariancją tonalną**. **Mikrointonacją** nazywamy taką wariancję tonalną, która jest przewidywalna na podstawie etykiet segmentalnej anotacji fonologicznej danej na wejściu obiektu klasy SegmentalEncoder (por. Hirst i Espesser 1993). Do mikrointonacji zalicza się m.in.: 1) nieokreśloność F_0 w sygnale *głosek bezdźwięcznych*, 2) obniżenie F_0 (zwykle $<5\%$) w sygnale *samogłosek zamkniętych*, 3) raptowne podwyższenie F_0 w sygnale *głosek zwarto-wybuchowych*.

Pozajęzykowe cechy tonalne to rodzaj wariancji tonalnej, która jest uniwersalna oraz nie jest kontrolowana przez mówcę. Pozajęzykowe cechy tonalne przenoszą informacje identyfikujące mówcę (układ nadawczy). Przykładem pozajęzykowych cech tonalnych jest zakres F_0 mówcy (Carlson i inni 2004, por. np.) oraz mikrointonacja.

[3]Należy zaznaczyć, że w dominującym obecnie statystycznym paradygmacie rozpoznawania mowy anotacje fonetyczne nie są stosowane.

Parajęzykowe cechy tonalne to rodzaj wariancji tonalnej, która jest uniwersalna oraz pozostaje pod kontrolą mówcy. Parajęzykowe cechy tonalne przekazują słuchaczowi informacje m.in. dotyczące zainteresowania oraz stanu emocjonalnego mówcy. Przykładem parajęzykowej cechy tonalnej jest rodzaj fonacji. Przyjmuje się w niniejszej pracy, że parajęzykowe cechy tonalne nie wpływają na komunikat językowy (tj. fonologiczną anotację suprasegmentalną).

Językowymi cechami tonalnymi nazywamy te specyficzne cechy tonalne, które nie są wariancją tonalną (por. np. Tench 1996, 152).

Intonacją nazywamy w niniejszej pracy takie językowe cechy tonalne, które podlegają analizie anotacyjnej z zastosowaniem suprasegmentalnego systemu fonologicznego. Rycina 2.2) przedstawia związany z intonacją podsystem głosowego systemu komunikacyjnego, który obejmuje następujące klasy: LanguageModel, SuprasegmentalEncoder, ArticulatoryExpander, SourceEffector, CommunicationChannel, NoiseSource, SourcePerceptor, AuditoryCompressor oraz SuprasegmentalDecoder (opisy wymienionych klas zamieszczono w sekcji 2.1).

Analizę językowych cech tonalnych wykonywaną w oparciu o suprasegmentalny system fonologiczny nazywamy **analizą intonacyjną**. Analogicznie przymiotnika „intonacyjny" będziemy używać w odniesieniu do powiązanych z znalizą intonacyjną: algorytmów, anotacji oraz systemów fonologicznych. Wreszcie **strukturą intonacyjną** nazywamy anotację intonacyjną, w której porządek etykiet podlega ustalonej gramatyce (formalnej lub stochastycznej).

Definicja 2.1. Jeśli intonacyjny system fonologiczny spełnia fonetyczne kryteria dystynktywności, to etykiety anotacji wyjściowych algorytmu nazywamy **melodiami**.

Definicja 2.2. Jeśli intonacyjny system fonologiczny spełnia fonologiczne kryteria dystynktywności, to etykiety anotacji wyjściowych algorytmu nazywamy **intonemami**[4].

2.3 Akcent potencjalny i realny

Niech będzie dany sygnał x oraz ramkowa anotacja sygnałowa $(S_\mathcal{E}, a_\mathcal{E})$ taka, że dla każdego $s_\mathcal{E} \in S_\mathcal{E}$ zachodzi:

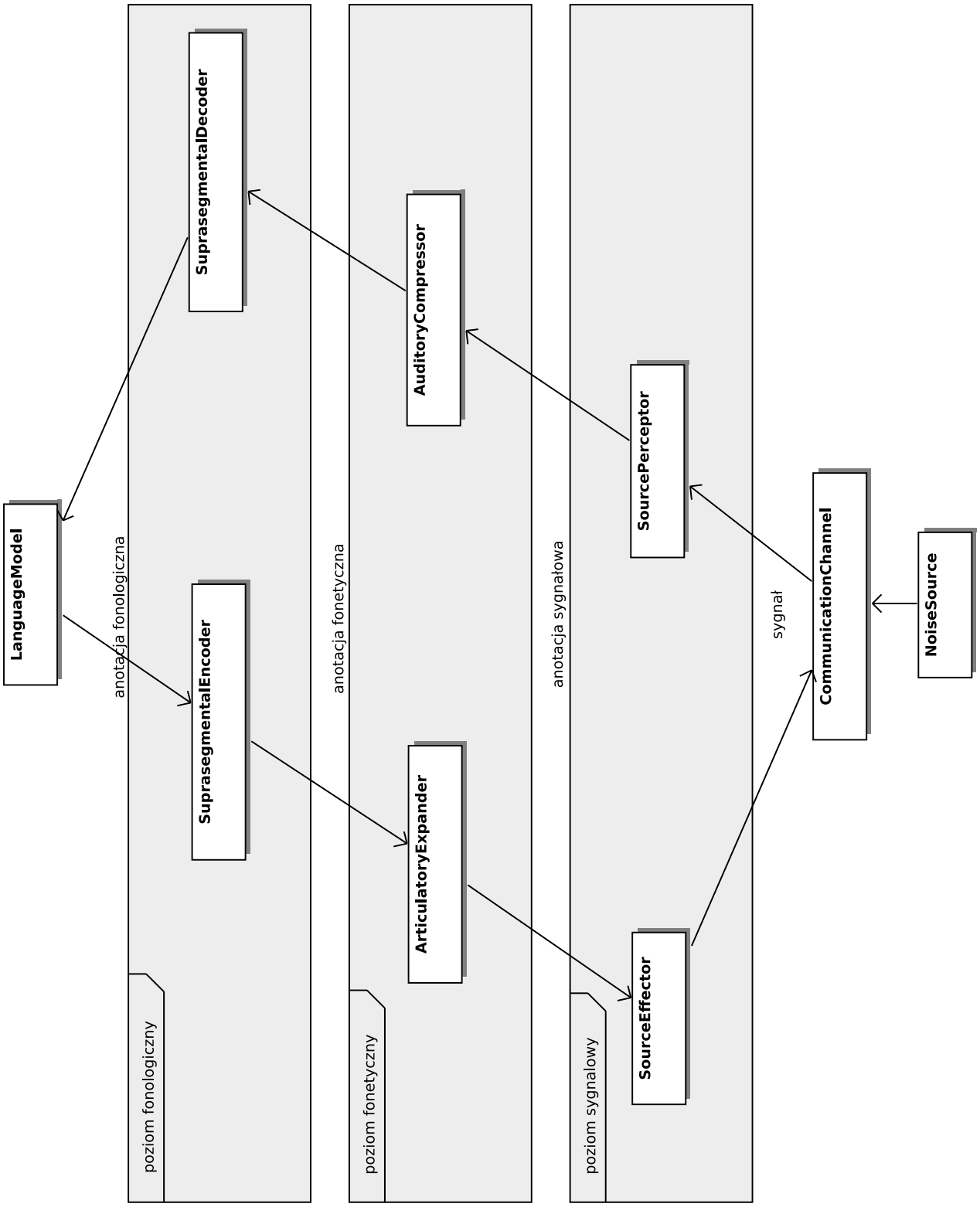
$$\overline{s_\mathcal{E}} = \frac{1}{\overrightarrow{s_\mathcal{E}} - \overleftarrow{s_\mathcal{E}}} \sum_{n=-\infty}^{\infty} |\boxplus(x, s)[n]|^2. \tag{2.1}$$

Anotacja $(S_\mathcal{E}, a_\mathcal{E})$ reprezentuje przebieg **mocy chwilowej** sygnału x. Jeśli x jest sygnałem mowy, to główny korelat percepcyjny mocy chwilowej nazywamy **donośnością**.

Definicja 2.3. Sylabizacją fonetyczną nazywamy segmentaję prostą, w której każdy segment obejmuje dokładnie jedno *niezależnie percypowane* lokalne maksimum donośności.

Segmenty sylabizacji fonetycznej nazywamy **sylabami fonetycznymi**. Przyjmuje się, że sylabizacja fonetyczna jest uniwersalna (nie zależy od języka naturalnego).

[4] Terminu „intonem" w literaturze polskiej użyła wcześniej Steffen-Batogowa (1996).



Rycina 2.2: Intonacyjny system komunikacyjny. Diagram klas UML.

Definicja 2.4. Niech S będzie sylabizacją fonetyczną taką, że $\overleftrightarrow{S} \subset \overleftrightarrow{S_{\mathcal{E}}}$. **Ośrodkiem fonetycznym** sylaby $s \in S$ nazywamy segment $u \blacktriangleleft s$ taki, że:

1. $(\max \overline{e} : e \in S_{\mathcal{E}} \wedge s \blacktriangleright e) = (\max \overline{e} : e \in S_{\mathcal{E}} \wedge u \blacktriangleright e)$,

2. $\underset{e,e' \in S_\mathcal{E}}{\forall} e' < e \rightarrow \overline{e} - \overline{e'} \leqslant \alpha$,

3. $\underset{e,e' \in S_\mathcal{E}}{\forall} e' > e \rightarrow \overline{e} - \overline{e'} \leqslant \beta$,

gdzie zwykle przyjmuje się $\alpha = 3\text{dB}$ oraz $\beta = 9\text{dB}$.

Definicja 2.5. Ścisłą dźwięcznością nazywamy monosegmentalną cechę sygnałową (por. definicja 1.43 na stronie 15) stwierdzającą, że ma widmo sygnałów segmentów jest *harmoniczne* oraz nie zawiera składowych *szumowych*.

Głoskę, dla której zachodzi ścisła dźwięczność (tj. nie ma podstaw do odrzucenia hipotezy o ścisłej dźwięczności) nazywamy **głoską ściśle dźwięczną**. Analogicznie określamy pojęcie **fonemu ściśle dźwięcznego**. Głównym korelatem artykulacyjnym ścisłej dźwięczności jest fonacja modalna połączona z nieturbulentnym przepływem powietrza w torze głosowym.

Definicja 2.6. Nosowością nazywamy monosegmentalną cechę sygnałową stwierdzającą, że w widmie sygnałów segmentów znajduje się *antyformant* w paśmie występowania pierwszego *formantu* (por. Pickett 1999, 117).

Głoskę, dla której zachodzi nosowość (tj. nie ma podstaw do odrzucenia hipotezy o nosowości) nazywamy **głoską nosową**. Analogicznie określamy pojęcie **fonemu nosowego**. Głównym korelatem artykulacyjnym nosowości jest otwarcie komory nosowej.

Definicja 2.7. Wokoidem[5] nazywamy głoskę, która jest ściśle dźwięczna i jednocześnie nie jest nosowa.

Definicja 2.8. Kontoidem nazywamy głoskę, która nie jest wokoidem, tj. głoskę, która jest nosowa lub nie jest ściśle dźwięczna.

Niech będzie ustalony leksykon L, w którym segmentalny tekst fonologiczny dowolnego elementu pasuje do wyrażenia regularnego:

```
/(K*)(W+)(K*W?)/,
```
(2.2)

gdzie `'K'` dopasowuje dowolny kontoid natomiast `'W'` dopasowuje dowolny wokoid. Elementy leksykonu L nazywamy **sylabami fonologicznymi** natomiast zbiór L **leksykonem sylab fonologicznych**. Części sylaby fonologicznej odpowiadające kolejnym parom nawiasów w wyrażeniu 2.2 nazywane są odpowiednio: **nagłosem** (*onset*), **ośrodkiem** (*nucleus*) oraz **wygłosem** (*coda*).

Definicja 2.9. Niech będzie ustalony leksykon sylab fonologicznych L. **Sylabizacją fonologiczną** dowolnej anotacji A nazywamy segmentację prostą, w której każdy segment jest oparty na tych samych kotwicach, co wystąpienie pewnego komunikatu ze zbioru L w anotacji A.

W dalszej części pracy pomijamy przymiotniki „fonetyczna” oraz „fonologiczna” przy terminach sylaba oraz sylabizacja jeśli rozróżnienie między definicją fonetyczną a fonologiczną wynika z kontekstu lub nie ma wpływu na poprawność stwierdzenia.

[5] Terminów „wokoid” oraz „kontoid” jako pierwszy użył Pike (1943).

Jest możliwe, by warstwy sylab fonetycznych oraz fonologicznych w tym samym komunikacie miały różną liczbę segmentów. Przykładowo leksy «rytm» oraz «umysł» mają jednosylabowe warstwy sylab fonologicznych oraz dwusylabowe warstwy sylab fonetycznych.

Definicja 2.10. Dla ustalonego leksykonu sylab fonologicznych L oraz zbioru wokoidów W **samogłoską** nazywamy każdy element maksymalnego podzbioru $V \subset W$, takiego że dowolna sylaba z leksykonu L zawiera co najwyżej jeden wokoid ze zbioru V.

Definicja 2.11. Spółgłoską nazywamy każdą głoskę, która nie jest samogłoską.

Sylabę fonetyczną, która nie zawiera samogłoski nazywamy **sylabą epentetyczną**.

Definicja 2.12. Niech będzie dany komunikat językowy K, leksykon sylab fonologicznych L oraz niech $\mathrm{syl}(K, L)$ oznacza zbiór wszystkich możliwych sylabizacji fonologicznych komunikatu językowego K za pomocą sylab należących do L. **Sylabizacją morfologiczną** komunikatu językowego K nazywamy sylabizację fonologiczną $S \in \mathrm{syl}(K, L)$ mającą maksymalną liczbę wspólnych kotwic segmentów sylabicznych oraz *morfemów*.

Np. w języku polskim można wyróżnić co najmniej dwie sylabizacje fonologiczne leksu «nadmarza»: 1) «na|dma|rza» oraz 2) «nad|mar|za». Wyłącznie w wariancie 2 pierwszy segment sylabiczny ma wspólną prawą kotwicę z morfemem «nad», stąd wariant 2 jest sylabizacją morfologiczną (przy założeniu, że warianty 1 oraz 2 stanowią wszystkie możliwe sylabizacje fonologiczne leksu «nadmarza»).

Uwydatnienie segmentu sylabicznego w ramach komunikatu za pomocą cech tonalnych nazywamy **akcentem melodycznym**. Akcent melodyczny jest warunkowany przez strukturę intonacyjną. Mówi się, że **język naturalny ma akcent melodyczny** wtedy i tylko wtedy, gdy w języku tym akcent melodyczny jest *głównym* sposobem uwydatniania segmentów sylabicznych. Przyjmujemy w niniejszej pracy, że język polski ma akcent melodyczny (Jassem 1962; Dogil 1995).

Definicja 2.13. Niech będą dane: korpus mowy M, leks $L = (o_L, K_L)$ oraz niech S_L będzie sylabizacją komunikatu językowego K_L. Mówimy, że segment sylabiczny $s_L \in S_L$ ma **potencjalny akcent melodyczny** wtedy i tylko wtedy, gdy istnieje taka trójka uporządkowana $(o_M, x_M, A_M) \in M$, że $\overleftarrow{s_M} = t + \overleftarrow{s_L}$, gdzie s_M jest segmentem pewnej melodii w anotacji A_M oraz $t \in \mathbb{Z}$ jest punktem czasowym w sygnale o_M takim, że $o_L = o_M[t..t + |o_L| - 1]$.

2.4 Struktura intonacyjna i interpretacja komunikatu

W szeregu języków naturalnych struktura intonacyjna komunikatu językowego jest powiązana z jego *strukturą składniową*. W języku angielskim istnieje ponad sto par leksów homograficznych, w których językowe cechy tonalne decydują o funkcji składniowej. Np. leksy «compress», «object», «transfer» wymawiane jako «com̲press», «o̲bject», «trans̲fer» mogą pełnić funkcję podmiotu ale nie orzeczenia a wymawiane jako «compr̲ess», «obj̲ect»,

«transfer» mogą pełnić funkcję orzeczenia ale nie podmiotu. Wells (2006, 188) podaje następujący przykład na zależność struktury składniowej komunikatu językowego od językowych cech tonalnych w języku angielskim: «What's that in the road ahead?», «What's that in the road? ‖ A head?». Podobne przykłady dla języka polskiego można zbudować w oparciu o pary: «pora dnia» oraz «poradnia», «ja jem» oraz «jajem», «zbiera liście» oraz «zbieraliście», «na wóz» oraz «nawóz» (Demenko 1999, 189), np. «Czy to najlesza poradnia?», «Czy to najlepsza pora dnia?».

W niektórych językach językowe cechy tonalne determinują denotat struktury składniowej. W języku angielskim komunikat językowy: «I didn't come because of the \\rain» oznacza, że mówca nie przyszedł z powodu padającego deszczu, podczas gdy komunikat językowy «I didn't come because of the \\//rain» oznacza, że mówca przyszedł ale z innego powodu niż deszcz (Jassem 2003a). Podobnie w komunikacie: «I don't lend my books to \\anybody» mówca stwierdza, że nikomu nie pożycza książek, natomiast w komunikacie «I don't lend my books to \\//anybody» mówca stwierdza, że nie pożycza byle komu (Jassem 2003a). Przyjmujemy, że w języku polskim cechy tonalne nie wpływają na wybór denotatu leksu ani denotatu struktury składniowej.

W wielu językach naturalnych językowe cechy tonalne (współ)określają tryb komunikatu językowego. W mowie polskiej same językowe cechy tonalne pozwalają odróżnić stwierdzenia od zapytania (por. np. Durand i inni 2002). Np. komunikat językowy «idziemy do \\kina» jest stwierdzeniem, natomiast komunikat językowy «idziemy do //kina» jest zapytaniem. (Znakiem \\ oznaczamy początek ściśle monotonicznego spadku F_0, znakiem // oznaczamy początek ściśle monotonicznego wzrostu F_0). Do funkcji językowych cech tonalnych na poziomie pragmatycznym zalicza się określanie statusu informacyjnego leksów (von Heusinger 1999; Ito i inni 2004). Cechy tonalne parajęzykowe analizowane na poziomie pragmatycznym umożliwiają rozpoznanie złożonych sytuacji komunikacyjnych takich, jak stosunek mówcę do słuchacza oraz rozpoznanie kłótni lub wydawania rozkazów. Model głosowego procesu komunikacyjnego proponowany w niniejszej pracy (rozdział 1) z założenia nie obejmuje cech tonalnych pozajęzykowych na poziomie pragmatycznym. Cechy tonalne pozajęzykowe analizowane na poziomie pragmatycznym umożliwiają m.in. określenie płci, przybliżonego wieku oraz stanu zdrowia mówcy.

Część II

Przegląd metod analizy cech tonalnych

ROZDZIAŁ 3

Sygnałowa analiza tonalna

Obiektywna sygnałowa analiza tonalna nazywana jest **ekstrakcją** F_0 lub **wyznaczaniem tonu podstawowego** (*Pitch Determination*). W niniejszej pracy będziemy używać terminu ekstrakcja F_0. Układ ekstrakcji F_0 będziemy nazywać **ekstraktorem** F_0.

Historia ekstraktorów F_0 sięga początków dwudziestego wieku, kiedy to zaadaptowano do potrzeb analizy sygnału mowy kimograf oraz stroboskop (Mehnert i Hoffmann 2006). Pierwsze elektroniczne ekstraktory F_0 zbudowali w latach 30-tych Dudley (1935) oraz Grützmacher i Lottermoser (1937). Pierwsze w polsce elektroniczne ekstraktory F_0 wykonali Kubzdela (1976) oraz Gubrynowicz i inni (1980).

Hess (2008, 182) szacuje, że do roku 2008 opublikowano ok. 3000 prac przedstawiających kilkaset algorytmów ekstrakcji F_0. Najobszerniejszą dotąd monografię na temat ekstrakcji F_0 opublikował Hess (1983). Kolejne lata prac nad ekstrakcją F_0 podsumowali w swoich pracach Gerhard (2003) oraz Hess (2008). W ostatnich dwudziestu latach, na skutek rozwoju najpierw komercyjnych[1] a następnie otwartych[2] aplikacji do analizy mowy, ekstraktory F_0 stały się dostępne dla szerokiego grona użytkowników komputerów osobistych.

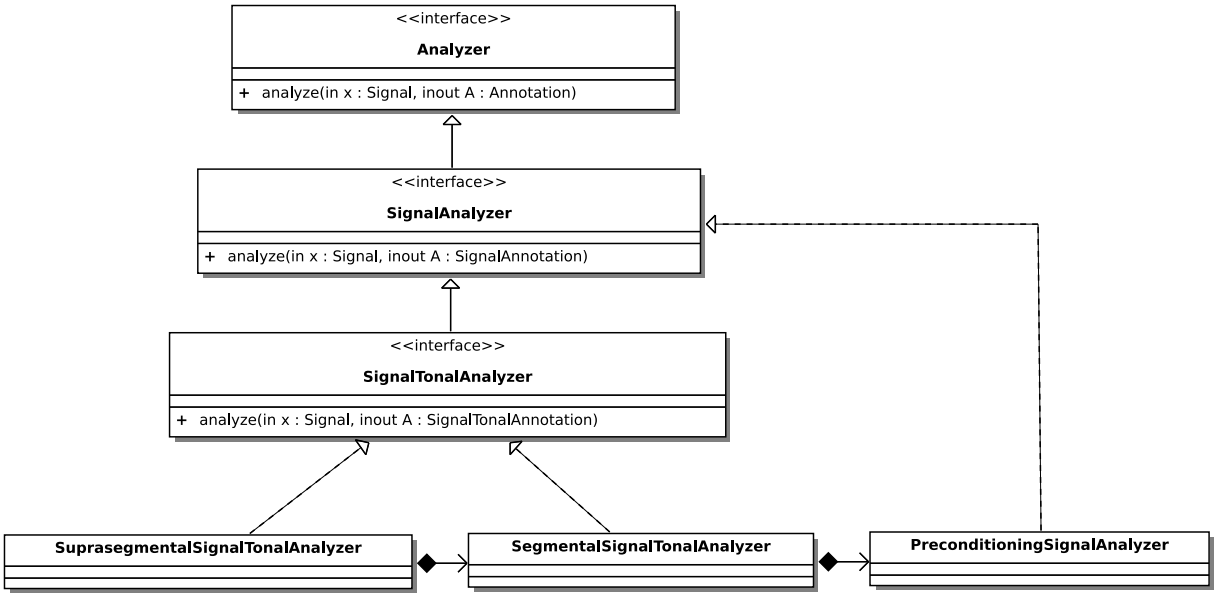
Ekstrakcję F_0 wykonuje się na: 1) sygnale pochodzącym ze Źródła sygnału (np. sygnał fonacji lub EGG[3]) albo 2) sygnale pochodzącym z Kanału komunikacyjnego.

Ze względu na pochodzenie kotwic w segmentacji wynikowej wyróżniamy dwa rodzaje ekstrakcji F_0: 1) analizę kotwiczącą z segmentacją prostą (**analiza w dziedzinie czasowej**, *time-domain analysis*) oraz 2) analizę niekotwiczącą z segmentacją ramkową (**analiza krótkookresowa**, *short-term analysis*). Segmentacje powstałe w wyniku analizy w dziedzinie czasowej wskazują lokalizacje prawieokresów w sygnale mowy. W przypadku analizy krótkookresowej segmentacja wynikowa jest ustalona przed rozpoczęciem analizy. W analizie krótkookresowej występuje ograniczenie będące analogią zasady nieoznaczoności Heisenberga:

[1] Wśród komercyjnych aplikacji do analizy mowy rozwijanych w ostatnich dwudziestu latach wymienić można m.in. CSL firmy Kay Elemetric, SpeechStation firmy Sensimetrics oraz xwaves firmy Entropic.

[2] Wśród rozwijanych obecnie otwartych aplikacji do analizy mowy można wymienić m.in. Speech Filing System (Huckvale i inni 1987), Praat (Boersma i Weenink 1996) oraz Emu (Cassidy i Harrington 2001)

[3] Znaczne podobieństwa w przebiegach sygnału fonacji oraz związanego z nią sygnału EGG pozwalają traktować te sygnały zamiennie (Marasek 1997).



Rycina 3.1: Krótkookresowy algorytm ekstrakcji F_0. Diagram klas UML.

dokładniejszy pomiar F_0 wymaga stosowania dłuższych segmentów, co zmniejsza dokładność lokalizacji pomiaru w wymiarze czasowym; problem ten analizuje m.in. Mallat (1999, 31). W dalszej części rozdziału ograniczamy się do przedstawienia algorytmów krótko-okresowej ekstrakcji F_0 ze względu na ich wiodącą rolę w realizacji fonetycznych oraz fonologicznych układów analizy tonalnej (por. rozdziały 4 oraz 5).

Funkcję $p : F \mapsto \mathbb{R}_+$, gdzie F jest dowolnym skończonym podzbiorem pasma cech tonalnych $[\gamma_L; \gamma_H]$ (por. str. 31), nazywamy **rozkładem** F_0. Przyjmujemy, że etykietą segmentu sygnałowej anotacji tonalnej jest trójka uporządkowana (e, h, p), gdzie e oraz h reprezentują odpowiednio energię oraz harmoniczność sygnału segmentu natomiast p jest rozkładem F_0 sygnału segmentu. Elementy e, h oraz p mogą być związane ze skalami ciągłymi lub dyskretnymi. W przypadku deterministycznej ekstrakcji F_0 rozkład p jest jednopunktowy.

Na rycinie 3.1 przedstawiono diagram klas UML uogólnionego algorytmu krótkookresowej ekstrakcji F_0. Na diagramie zamieszczono trzy klasy algorytmów stosowanych w algorytmie ekstrakcji F_0:

1. wstępną analizę sygnałową (PreconditioningSignalAnalyzer); zastosowanie: segmentacja sygnału, okienkowanie, zmniejszenie energii cech nietonalnych oraz wariancji tonalnej występujących w sygnale wejściowym.

2. segmentalną sygnałową analizę tonalną (SegmentalSignalTonalAnalyzer); zastosowanie: określenie rozkładów F_0 dla segmentów rozpatrywanych w izolacji.

3. suprasegmentalną sygnałową analizę tonalną (SuprasegmentalSignalTonalAnalyzer); zastosowanie: ponowne określenie rozkładów F_0 dla segmentów z uwzględnieniem rozkładów F_0 pozostałych segmentów należących do anotacji.

W dalszych sekcjach przedstawiono wybrane obiektywne algorytmy stosowane w implementacji poszczególnych klas modelu ekstraktora F_0.

Sygnałową anotację tonalną A_x^R nazywamy **referencyjną** dla sygnału x, wtedy i tylko wtedy, gdy pożądane jest, by dla dowolnego algorytmu sygnałowej analizy tonalnej f zachodziło:

$$D(f(x), A_x^R) = 0, \tag{3.1}$$

gdzie D jest ustaloną odległością anotacyjną. Trzy podstawowe metody tworzenia par uporządkowanych (x, A_x^R) to:

1. intersubiektywna sygnałowa analiza tonalna sygnału x,
2. obiektywna sygnałowa analiza tonalna sygnału EGG związanego z sygnałem x,
3. synteza sygnału x z zadanej anotacji A_x^R.

Wartość $D(f(x), A_x^R)$ w równaniu 3.1 nazywamy **błędem ekstrakcji** F_0. W definicjach błędu ekstrakcji F_0 stosuje się głównie odległości anotacyjne SSE, MSE oraz RMSE opisane na stronie 10.

Wprowadźmy funkcję pomocniczą:

$$f_{\max}(p) = \operatorname*{argmax}_f p(f), \tag{3.2}$$

gdzie p jest funkcją rozkładu F_0. Przyjmijmy ponadto oznaczenia $c_i = (s_i, a_i)$ oraz $a_i = (e_i, h_i, p_i)$. Rabiner (1977) zaproponował podział błędów ekstrakcji F_0 na **grube** (*gross*) oraz **drobne** (*fine*) zgodnie z którym definicje cząstkowych odległości anotacyjnej przyjmują postać:

$$d_{\text{fine}}^{\alpha}(c_1, c_2) = \begin{cases} |f_{\max}(p_1) - f_{\max}(p_2)| & \text{dla } \dfrac{2|f_{\max}(p_1) - f_{\max}(p_2)|}{f_{\max}(p_1) + f_{\max}(p_2)} < \alpha \\ 0 & \text{w przec. przyp.,} \end{cases} \tag{3.3}$$

$$d_{\text{gross}}^{\alpha}(c_1, c_2) = \begin{cases} 1 & \text{dla } \dfrac{2|f_{\max}(p_1) - f_{\max}(p_2)|}{f_{\max}(p_1) + f_{\max}(p_2)} > \alpha \\ 0 & \text{w przec. przyp.,} \end{cases} \tag{3.4}$$

dla $\alpha \in [0.02; 0.2]$. Cząstkową odległość anotacyjną dla błędów harmoniczności określamy następująco:

$$d_{\text{harmo}}^{\beta}(c_1, c_2) = \begin{cases} 1 & \text{dla } |h_1 - h_2| > \beta \\ 0 & \text{w przec. przyp.,} \end{cases} \tag{3.5}$$

dla pewnego $\beta > 0$. Ahmadi i Spanias (1999) proponuje ważony błąd F_0 z uwzględnieniem energii okna segmentu, którego sformułowanie w kategoriach cząstkowej odległości anotacyjnej ma postać:

$$d_{\text{GPE}}(c_1, c_2) = \left(\frac{e_1}{e_m ax}\right)^2 \left|\frac{2|f_{\max}(p_1) - f_{\max}(p_2)|}{f_{\max}(p_1) + f_{\max}(p_2)}\right|. \tag{3.6}$$

Średnia wartość $d_{\text{gross}}^{0,2}$ dla współczesnych ekstraktorów F_0 wyliczona na czterech znacznie różniących się korpusach mowy zawiera się w przedziale $[0.01; 0.1]$ (de Cheveigné i Kawahara 2002).

Do najczęstszych błędów o niezerowej wartości $d_{\text{gross}}^{\alpha}$ należą: połowienie F_0 (*pitch halving*) oraz podwajanie F_0 (*pitch doubling*). Połowienie F_0 występuje m.in. na skutek fonacji chrypliwej. Podwajanie F_0 występuje m.in. w sytuacji, gdy druga harmoniczna zostaje uwydatniona przez F_1. Błędy o niezerowej wartości d_{fine}^{α} wynikają m.in. ze zbyt niskiej rozdzielczości częstotliwościowej algorytmu analizy.

Szereg algorytmów stosowanych w ekstrakcji F_0 należy to technik cyfrowego przetwarzania sygnałów (DSP, *Digital Signal Processing*). Obszerne monografie na temat analizy mowy obejmujące DSP opublikowali m.in. Gold (1999), Huang i inni (2001), Schroeder (2004) oraz Benesty i inni (2008).

3.1 Wstępna analiza sygnałowa

W wyniku wstępnej analizy sygnałowej powstaje anotacja sygnałowa ramkowa (zgodnie z zadanymi parametrami), gdzie etykieta $e(i)$ jest wektorem (w szczególności sygnałem) lub macierzą reprezentującą sygnał segmentu i. W dalszych sekcjach opisano pięć rodzajów algorytmów stosowanych w analizatorach wstępnych: 1) algorytmy redukcji pasma, 2) algorytmy heurystyczne w dziedzinie czasowej, 3) algorytmy transformacyjne, 4) algorytmy artykulacyjne oraz 5) algorytmy percepcyjne.

3.1.1 Algorytmy redukcji pasma

Pasmem cech tonalnych nazywamy pasmo $[\gamma_L; \gamma_H]$, poza którym cechy tonalne w sygnale mowy nie występują lub ich występowanie nie ma *znaczącego* wpływu na wyniki sygnałowej analizy tonalnej. Wartości γ_L oraz γ_H ustala się przy uwzględnieniu własności Źródła sygnału oraz algorytmu sygnałowej analizy tonalanej. Przez algorytm redukcji pasma w kontekście sygnałowej analizy wstępnej rozumiemy algorytm przetwarzania sygnału, który zwiększa stosunek energii sygnału w paśmie cech tonalnych do energii sygnału poza pasmem cech tonalnych. Dwie podstawowe korzyści, których oczekuje się w związku z zastosowaniem algorytmów redukcji pasma to: 1) zmniejszenie udziału cech nietonalnych oraz zakłóceń w sygnale mowy, 2) możliwość ograniczenia częstotliwości próbkowania sygnału do nieco ponad $2\gamma_H$.

Za ograniczenie dolne γ_L przyjmuje się wartość minimalną F_0 w mowie (ok. 40Hz w fonacji chrypliwej dla głosów męskich). Przyjmując dodatkowe założenia dla Źródła sygnału (np. płeć, rodzaj fonacji) wartość γ_L można zwiększyć, np. do 160Hz (wysoki głos kobiecy w fonacji modalnej).

Dla algorytmów ekstrakcji F_0, w których analizuje się strukturę harmoniczną sygnału mowy górnym ograniczeniem γ_H jest częstotliwość maksymalna formantu czwartego, tj. ok. 4kHz (Stevens 1998, 278). W przypadku szeregu algorytmów ekstrakcji F_0, γ_H mieści się w przedziale od 500Hz do 2kHz.

Do najczęstszych źródeł zakłóceń sygnału mowy występujących poza pasmem cech tonalnych należą:

- Napięcie niezrównoważenia (*DC offset*): przesunięcie napięciowe w elektronicznych układach wzmacniania sygnału (przebieg stały lub wolnozmienny w paśmie $[0; 5]$ Hz).
- Wibracje akustyczne: ruch membrany wywołany wibracjami przenoszonymi na korpus mikrofonu (przebieg periodyczny lub tłumiony w paśmie $[15; 45]$ Hz).
- Zasilanie układów analogowych prądem przemiennym: niedoskonałości układów konwersji prądu przemiennego na prąd stały; indukcja elektromagnetyczna przewodników sygnału elektrycznego przez przewodniki prądu przemiennego (przebieg periodyczny tętniący w paśmie $[50; 60]$ Hz).

Tabela 3.1: Wybrane algorytmy heurystyczne wstępnej analizy sygnałowej.

Nazwa	Formuła	Przykładowe wartości parametrów
pre-emfaza	$y[n] = x[n] - ax[n-1]$	$a \in [0.97; 0.99]$
prostownik dwupołówkowy	$y[n] = \lvert x[n]\rvert^a$	$a = 0.33, 0.5, 1, 2$
prostownik jednopołówkowy	$y[n] = max(x[n], 0)$	
center-clipping	$y[n] = \begin{cases} 0 & \text{dla} \lvert x[n]\rvert < cA \\ x[n] & \text{w przec. przyp.} \end{cases}$	$c = 0.1, 0.2$, $A = \max_{n\in\mathbb{Z}}(x[n])$
peak-clipping	y[n]=min(|x[n]|, cA)*sign(x[n])	$c = 0.2, 0.4$, $A = \max_{n\in\mathbb{Z}}(x[n])$
Single Side Band	$x[n] = \sqrt{\max(SSB(x), 0)}$	

- Niewystarczająca częstotliwość próbkowania (*aliasing*): niedoskonałości lub brak filtrów dolnoprzepustowych przed konwersją analogowo–cyfrową.

Pożądane jest, by oprócz zakłóceń, poza pasmem cech tonalnych znalazła się jak największa część energii przebiegów szumowych sygnału mowy.

W celu redukcji energi sygnału mowy poza pasmem $[\gamma_L; \gamma_H]$ stosuje się filtry cyfrowe środkowoprzepustowe o częstotliwościach odcięcia (γ_L, γ_H) lub łączone w szereg filtry cyfrowe dolno oraz górnoprzepustowe o częstotliwościach odcięcia odpowiednio γ_H oraz γ_L.

3.1.2 Algorytmy heurystyczne w dziedzinie czasowej

W tabeli 3.1 przedstawiono wybrane algorytmy heurystyczne w dziedzinie czasowej (por. Hess 1983). Oznaczenia użyte w tabeli: x — sygnał cyfrowy wejściowy, y — sygnał cyfrowy wyjściowy, SSB — modulacja wąskopasmowa (*Single Side Band*) (por. Szabatin 2000, 410). Algorytmy heurystyczne pozwalają m.in. na skompensowanie spadku długookresowego uśrednionego widma sygnału mowy (LTASS — *Long–Term Average Speech Spectrum*), uwydatnienie pierwszej harmonicznej oraz zmniejszenie energii formantów.

Algorytmy heurystyczne a także ich implementacje w formie układów elektronicznych były szczególnie popularne w okresie rozwoju analogowych elektronicznych ekstraktorów F_0 (od lat 30-tych do lat 70-tych XX wieku). Jak wykazały późniejsze badania porównawcze, skuteczność przetworników heurystycznych podlega znacznym wahaniom w zależności od własności kanału komunikacyjnego, cech pozajęzykowych sygnału mowy oraz rodzaju segmentalnej analizy tonalnej (Hess 1983, 313-322). Współcześnie powszechnie stosowanym przetwornikiem heurystycznym pozostaje pre-emfaza (*pre-emphasis*), która pozwala kompensować spadek LTASS (por. Huckvale i inni 1987; Boersma i Weenink 2008)).

3.1.3 Algorytmy transformacji częstotliwościowej

Niech $x_s = \boxplus((,x),s)$, gdzie x jest sygnałem mowy oraz s jest segmentem anotacji ramkowej.

Dyskretna transformata Fouriera (DFT — *Discrete Fourier Transform*) sygnału x_s jest zdefiniowana następująco:

$$X_s[k] = \sum_{n=0}^{N-1} x_s[n + \overleftarrow{s}]e^{-i2\pi nk/N}, \tag{3.7}$$

gdzie N jest liczbą próbek sygnału x znajdujących się w granicach segmentu s oraz $0 \leqslant k < N$. Jeśli X_s jest dyskretną transformatą Fouriera sygnału x_s, to piszemy:

$$X_s \leftrightsquigarrow x_s. \tag{3.8}$$

Szereg zastosowań w algorytmach segmentalnej sygnałowej analizy tonalnej znajduje **widmo amplitudowe** sygnału x_s zdefiniowane następująco:

$$A_s[k] = |X_s[k]|, \tag{3.9}$$

gdzie $X_s \leftrightsquigarrow x_s$.

Jeśli N jest liczbą B-gładką[4], gdzie $B \ll N$, to do wyznaczenia X_N stosuje się algorytm FFT (*Fast Fourier Transform*). Poniżej przytaczamy definicję algorytmu FFT dla przypadku $N = 2^k$, gdzie $k \in \mathbb{N}$ (Huang i inni 2001, 223). Niech $x[n] = x_s[n + \overleftarrow{s}]$ oraz $W_N = e^{-i2\pi/N}$. Zgodnie z przyjętymi oznaczeniami DFT przyjmuje postać:

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk}, \tag{3.10}$$

gdzie $0 \leqslant k < N$. Oznaczmy przez f oraz g sygnały cyfrowe takie, że $f[n] = x[2n]$ oraz $g[n] = x[2n+1]$. Algorytm FFT opiera się na rekursywnym wykorzystaniu następującego spostrzeżenia:

$$X[k] = \sum_{n=0}^{N/2-1} f[n]W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} g[n]W_{N/2}^{nk} = F[k] + W_N^k G[k], \tag{3.11}$$

gdzie $F \leftrightsquigarrow f$ oraz $G \leftrightsquigarrow g$ są transformatami $N/2$ punktowymi. Wartości $X[k]$ dla $N/2 \leqslant k < N$ oblicza się korzystając z równości $F[k + N/2] = F[k]$ oraz $G[k + N/2] = G[k]$. Złożoność obliczeniowa algorytmu FFT wynosi $\mathrm{O}(N \log N)$, co wykazali Cooley i Tukey (1965).

Transformata falkowa sygnału analogowego[5] x ma postać:

$$Wx(t,f) = \int_{-\infty}^{+\infty} x(u)\frac{1}{\sqrt{f}}\psi^*\left(\frac{u-t}{f}\right)\mathrm{d}u, \tag{3.12}$$

[4] Liczba B-gładka to liczba, której wszystkie dzielniki pierwsze są mniejsze lub równe B.

[5] Równania z zakresu analizy falkowej dla uproszczenia przedstawiamy w wersji analogowej, co jest przyjęte w publikacjach z zakresu teorii falek (por. Białasiewicz 2000).

gdzie funkcja ψ spełniająca:

$$\int_{-\infty}^{+\infty} \psi(u)\mathrm{d}u = 0 \tag{3.13}$$

nazywana jest **falką**. Wyznaczenie zbioru wartości (dyskretnej) transformaty falkowej dla zbioru argumentów tworzących *bazę ortonormalną* wymaga $\mathrm{O}(N)$ operacji, gdzie N jest długością okna sygnałowego; stosowany algorytm o nazwie FWT (*Fast Wavelet Transform*). Więcej o transformatach falkowych piszą m.in. Mallat (1999) oraz Białasiewicz (2000).

Chisaki i inni (2003) proponuje **harmoniczną transformatę falkową** (*Harmonic Wavelet Transform*, HWT), w której falka ψ jest określona następująco:

$$\frac{1}{\sqrt{f}}\psi^*\left(\frac{u-t}{f}\right) = \frac{\sqrt{f}}{||h||^2}h(t,f,u), \tag{3.14}$$

gdzie $||h||^2$ oznacza normę L^2 funkcji h, przy czym

$$h(t,f,u) = w(u)\sum_{k=1}^{n}\alpha_k e^{i(2\pi f k(u-t)+\phi_k)}, \tag{3.15}$$

dla gaussowskiego okna sygnałowego w, $-N/2f \leqslant u \leqslant N/2f$ oraz ustalonych parametrów n, α_k oraz ϕ_k. W stosunku do transformat opartych na analizie Fourierowskiej, transformaty falkowe są bardzej odporne na łamanie założeń o stacjonarności sygnału w obrębie segmentu analizy.

3.1.4 Algorytmy artykulacyjne

Zgodnie z modelem produkcji mowy źródło-filtr (por. strona 19 oraz rycina 2.1) sygnał mowy x powstaje w wyniku niezależnego działania generatora sygnału g (źródło) oraz niezmiennego w czasie systemu liniowego (LTI) o odpowiedzi impulsowej h (filtr):

$$x = g * h, \tag{3.16}$$

gdzie gwiazdka oznacza splot sygnałów. Model źródło-filtr opiera się na założeniu o stacjonarności sygnału x. Z własności splotu sygnałów wynika, że:

$$X(z) = G(z)H(z), \tag{3.17}$$

gdzie $X \leftrightsquigarrow x$, $G \leftrightsquigarrow g$ oraz $H \leftrightsquigarrow h$.

Filtrowanie odwrócone (*inverse filtering*) jest artykulacyjnym algorytmem analizy wstępnej, pozwalającym wyznaczyć sygnał g z sygnału x przy znajomości transformaty H:

$$g \leftrightsquigarrow X(z)\frac{1}{H(z)}. \tag{3.18}$$

W praktyce zamiast transformaty H używa się estymaty $\widehat{H}$ wyznaczonej z sygnału x.

Dla danego sygnału cyfrowego x **sygnałem predykcji liniowej** rzędu p jest sygnał $\tilde{x}$, którego wartość w chwili n jest liniową kombinacją próbek $(x[n-1], x[n-2], .., x[n-p])$:

$$\tilde{x}[n] = \sum_{k=1}^{p} a_k x[n-k], \tag{3.19}$$

gdzie parametry $a_k \in \mathbb{R}$ nazywane są **współczynnikami predykcji liniowej** (Atal i Hanauer 1971). Sygnałem **błędu predykcji liniowej** nazywamy sygnał $\tilde{e}$ zdefiniowany następująco:

$$\tilde{e}[n] = x[n] - \tilde{x}[n]. \tag{3.20}$$

Współczynniki predykcji liniowej dla sygnału x wyznaczane są poprzez minimalizację średniokwadratowego błędu predykcji liniowej. **Liniowe kodowanie predykcyjne** (*Linear Predictive Coding* — LPC) jest odwracalną sygnałową analizą segmentalną, która opiera się na segmentacji ramkowej, gdzie etykietą dowolnego segmentu s jest wektor współczynników predykcji liniowej oraz sygnał błędu predykcji liniowej sygnału $\boxplus(x, s)$. O sygnale $\boxplus(x, s)$ zakłada się, że jest stacjonarny. Gold (1999, 281-291) oraz Huang i inni (2001, 290-306) zamieszczają szereg informacji praktycznych na temat stosowania LPC.

W filtrowaniu odwróconym z użyciem LPC za estymatę transformaty H przyjmuje się:

$$\widehat{H}(z) = \frac{a}{1 - \sum_{k=1}^{p} a_k z^{-k}}, \tag{3.21}$$

gdzie $a \in \mathbb{R}$ jest stałą a pozostałe parametry zdefiniowane są jak w równaniu 3.19. W związku z 3.21 estymatą sygnału g jest:

$$\widehat{g}[n] = \sum_{i=0}^{|S|} \boxplus(e_i, s_i)[n - \overleftarrow{s_i}], \tag{3.22}$$

gdzie S jest segmentacją LPC, s_i jest i-tym elementem segmetacji S w porządku $\overleftarrow{<}^S$ oraz $\boxplus(e_i, s_i,)$ jest błędem predykcji liniowej sygnału $\boxplus(x, s_i)$. Więcej o filtrowaniu odwróconym metodą LPC i jego rozszerzeniach napisali m.in. Hess (1983, 221-229), Hedelin (1984), Dutoit (1997, 221-226) oraz Ishi (2004).

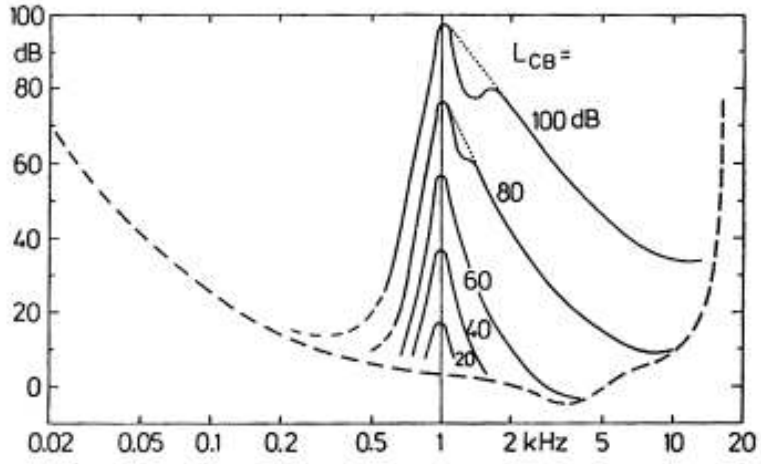
Transformatą cepstralną sygnału cyfrowego x nazywamy ciąg c taki, że:

$$c[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln |X(e^{i\omega})| e^{i\omega n} \mathrm{d}\omega, \tag{3.23}$$

gdzie $X \leftrightsquigarrow x$; por. np. Gold (1999, 271). Transformację C taką, że $C(x) = c$, gdzie x jest sygnałem cyfrowym oraz c jest transformatą cepstralną x, nazywamy **transformacją cepstralną**. Transformacja cepstralna jest **transformacją homomorficzną**, tj. spełniony jest warunek:

$$C(x_0 * x_1) = C(x_0) + C(x_1), \tag{3.24}$$

gdzie x_0 oraz x_1 są sygnałami cyfrowymi (Bogert i inni 1963).



Rycina 3.2: Progi percepcji tonu maskowanego przez przebieg szumowy wąskopasmowy o częstotliwości 1kHz oraz energii L_{CB} (Fastl i Zwicker 2007, 65).

Po podstawieniu 3.16 do 3.24 otrzymujemy:

$$C(x) = C(g) + C(h). \tag{3.25}$$

Z dobrym przybliżeniem można założyć, że istnieje $M \in \mathbb{Z}$ takie, że:

$$\underset{i<M}{\forall} C(g)[i] = 0 \tag{3.26}$$

oraz

$$\underset{i\geqslant M}{\forall} C(h)[i] = 0 \tag{3.27}$$

(por. Oppenheim i inni 1968). Estymatą g w przygotowaniu sygnału metodą cepstralną jest:

$$\widehat{g} = C^{-1}(L_M(C(x)), \tag{3.28}$$

gdzie $C^{-}1$ oznacza odwrotną transformację cepstralną oraz L_M jest funkcją, która dla danego ciągu liczb c zwraca ciąg c' taki, że:

$$c'[n] = \begin{cases} 0, & \text{dla } n < M, \\ c[n], & \text{w przec. przyp..} \end{cases} \tag{3.29}$$

3.1.5 Algorytmy percepcyjne

Maskowanie to zjawisko psychoakustyczne, w wyniku którego nie podlegają percepcji części widma sygnału występujące w otoczeniu silnych maksimów lokalnych obwiedni widma. Przez otoczenie w powyższym stwierdzeniu rozumie się zarówno otoczenie w wymiarze częstotliwości jak i w wymiarze czasu. Od lat 20-tych XX-tego wieku przeprowadzono szereg eksperymentów, w wyniku których wyznaczono progi słyszalności tonów oraz przebiegów szumowych w otoczeniu tonów oraz przebiegów szumowych o większej energii (Stevens 1998, 229-241), (Fastl i Zwicker 2007, 61-110). Rycina 3.2 przedstawia przykładowe progi percepcji tonu prostego przy jednoczesnym występowaniu przebiegu szumowego wąskopasmowego o częstotliwości 1kHz (szum maskuje ton). Huang i inni (2001, 35) podają następującą aproksymację progu percepcji tonu:

$$T_T(b) = E_N - 6.025 - 0.275g + S_m(b - g), \tag{3.30}$$

gdzie

$$S_m(b) = 15.81 + 7.5(b + 0.474) - 17.5\sqrt{1 + (b + 0.474)^2} \tag{3.31}$$

oraz g jest częstotliwością maskującego przebiegu szumowego wyrażoną w Barkach a E_N jest energią maskującego przebiegu szumowego wyrażoną w decybelach (SPL).

Terhardt i inni (1982) zaproponował ekstraktor F_0 zawierający model maskowania. Niech x będzie sygnałem mowy. Danymi wejściowymi modelu maskowania Terhardta jest sygnałowa ramkowa anotacja segmentalna, w której etykietą segmentu s jest widmo amplitudowe segmentu s. Oznaczmy przez $f(i)$ oraz $L(i)$, gdzie $i = 0, 1, \ldots, N$, odpowiednio częstotliwość oraz poziom (dB) i-tego maksimum lokalnego $|X_s|$, gdzie $X_s \rightsquigarrow \boxplus(x, s)$. Niech $L_M(m, k)$ oznacza poziom (dB) powyżej którego następuje percepcja tonu lub przebiegu szumowego o częstotliwości $f(k)$ przy jednoczesnym występowaniu dokładnie jednego tonu lub szumu o częstotliwości $f(m)$ i poziomie $L(m)$. ($L_M(m, k)$ liczone jest analogicznie jak w wyrażeniu 3.30.) Oznaczmy przez:

$$A_M(m, k) = 10^{L_M(m,k)/20\text{dB}}, \tag{3.32}$$

amplitudę odpowiadającą $L_M(m, k)$. Zgodnie z modelem Terhardta amplituda maskowania k-tego maksimum lokalnego przez pozostałe maksima lokalne wynosi:

$$A(k) = \sum_{i \neq k} A_M i, k. \tag{3.33}$$

Ostatecznie przyjmuje się, że k-te maksimum lokalne jest percypowane wtedy i tylko wtedy, gdy:

$$L(k) > 10 \log \left(A_M^2(k) + 10^{T(k)/10\text{dB}} + I \right), \tag{3.34}$$

gdzie $T(k)$ jest bezwzględnym progiem słyszalności (dB) oraz I jest korektą wprowadzaną w przypadku szumowych maksimów lokalnych (Hess 1983, 72). Model Terhardta nie uwzględnia maskowania w wymiarze czasowym.

W modelu percepcji wysokości tonu SPINET wstępna analiza sygnałowa obejmuje następujące cztery etapy przetwarzania sygnału (Cohen i inni 1995):

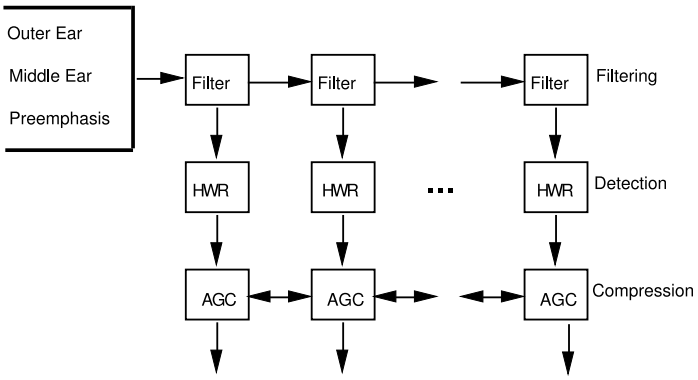
1. ławę 512-tu środkowoprzepustowych filtrów Gamma rozłożonych równomiernie na skali częstotliwości ERB, modelujących błonę podstawną (Cohen i inni 1995, 5,41); por. Stevens (1998, 204),
2. krótkookresowe uśrednianie energii sygnałów wyjściowych ławy w wymiarze czasowym,
3. filtr środkowoprzepustowy w paśmie [500Hz; 2kHz] modelujący m.in. własności kanału słuchowego; por. Stevens (1998, 204).
4. algorytm współpracy konkurencyjnej (*cooperative-competitive interactions*) modelujący m.in. maskowanie tonalne.

Formuła reprezentująca etapy 1-3 przyjmuje następującą postać:

$$Y(f_i, n) = BB(f_i) \sum_{\tau_1=0}^{\Theta} \frac{(1 - \alpha)}{\Theta} \left(\sum_{\tau_2=0}^{n} x^2(f_i, n - (\tau_1 + \tau_2)) \alpha^{\tau_2} \right)^{1/2}, \tag{3.35}$$

gdzie $x(f_i, n)$ reprezentuje n-tą próbkę sygnału wyjściowego z filtra ławy o częstotliwości f_i, $\alpha = 0.9914$, $\Theta = 80$ (częstotliwość próbkowania 16kHz) oraz

$$BB(f_i) = \frac{1}{1000} f_i \exp(-s f_i). \tag{3.36}$$



Rycina 3.3: Model percepcji wysokości tonu według Lyona (za: Slaney 1988, 65).

Niech

$$|H(f_i, f, \kappa)|^2 = [1 + ((f - f_i)/\kappa b(f_i))^2]^{-4}, \tag{3.37}$$

gdzie $b(f_i) = \zeta_{ERB}(f_i)/0.982$. Etap 4 wstępnej analizy sygnałowej w modelu SPINET przyjmuje postać:

$$S(f_i, n) = \sum_{j=1}^{512} Y(f_i, n) \left[\frac{|H(f_i, f_j, \kappa_{\text{ex}})|^2}{A_{\text{ex}}(f_i)} - \frac{|H(f_i, f_j, \kappa_{\text{in}})|^2}{A_{\text{in}}(f_i)} \right], \tag{3.38}$$

gdzie $\kappa_{\text{ex}} = 0.4$, $\kappa_{\text{in}} = 0.6$ oraz $n = \overleftarrow{s} = n$.

Wstępna analiza sygnałowa w modelu percepcji wysokości tonu Lyona (Lyon 1982) obejmuje następujące cztery etapy przetwarzania sygnału (Slaney i Lyon 1993):

1. filtr modelujący podstawowe własności akustyczne kanału słuchowego (Slaney 1988, 24); por. Stevens (1998, 204),
2. kaskada filtrów dolnoprzepustowych (ok. 80) modelująca błonę podstawną (Slaney 1988, 24), por. Stevens (1998, 208),
3. prostownik jednopołówkowy (*Half–Wave Rectifier*, HWR) modelujący jednostronną odpowiedź komórek włoskowatych,
4. automatyczny kontroler wzmocnienia (AGC — *Automatic Gain Control*) modelujący maskowanie oraz kompresję dynamiczną synaps eferentnych na komórkach włoskowatych (Slaney 1988, 39); por. Matthews (2000, 462).

Reprezentacja sygnału otrzymana na wyjściu modelu Lyona nazywana jest **cochleagramem**. Cosi i inni (1998) oraz Ottaviani i Rocchesso (2001) zaproponowali ekstraktory F_0 oparte na cochleagramie. Jedną z wymienianych przez autorów zalet cochleagramu jest odporność na addytywne zakłócenia szumowe.

W ostatnim dziesięcioleciu w badaniach psychoakustycznych coraz szerzej stosuje się drobnoziarniste modele percepcji dźwięku (w tym wysokości tonu), gdzie modelowanymi przetwornikami sygnału są np. poszczególne komórki włoskowate oraz włókna nerwowe (Bleeck i inni 2004; Meddis i O'Mard 2006; O'Mard i inni 2007). Podstawową przeszkodą w zastosowaniu drobnoziarnistych modeli modeli percepcji w przygotowaniu sygnału jest wysoki koszt obliczeniowy przy jednoczesnym braku jednoznacznych przesłanek o skuteczności. Gold (1999, 214-227) oraz de Cheveigné (2005) publikują przeglądy psychoakustycznych modeli wysokości tonu.

3.2 Segmentalna sygnałowa analiza tonalna

Niech y_s będzie etykietą segmentu s należącego do anotacji otrzymanej w wyniku sygnałowej analizy wstępnej; w szczególności y_s może być sygnałem segmentu s. **Periodogramem** nazwiemy funkcję rzeczywistą przypisaną sygnałowi y_s, której dziedzina reprezentuje okres (częstotliwość) a przeciwdziedzina reprezentuje *poziom prawieokresowości* sygnału y_s dla zadanego okresu.[6] Wyróżniamy dwa etapy działania algorytmu segmentalnej sygnałowej analizy tonalnej: 1) tworzenie periodogramu oraz 2) tworzenie rozkładu F_0. Rozkład F_0 tworzony jest poprzez wybór podzbioru maksimów lokalnch periodogramu w granicach pasma cech tonalnych. W dalszych sekcjach zamieszczono przegląd periodogramów.

3.2.1 Periodogramy samopodobieństwa

Periodogramem samopodobieństwa sygnału y_s nazywamy periodogram $AF : \mathbb{Z} \mapsto \mathbb{R}$, którego wartość AF($d$) reprezentuje *podobieństwo* sygnału y_s do sygnału y_s przesuniętego w czasie o d próbek. Oczekuje się, że prawieokresowy sygnał y_s wykazuje (lokalnie) maksymalne podobieństwo do przesuniętego sygnału y_s, gdy przesunięcie równe jest okresowi podstawowemu.

Dwa podstawowe periodogramy samopodobieństwa to ACF (Auto–Correlation Function, funkcja autokorelacyjna) (Rabiner 1977) oraz -AMDF (funkcja przeciwna do Average Magnitude Difference Function) (Sobolev i Baronin 1968) za (Hess 2008).

$$\mathrm{ACF}(d) = \sum_i y_s[i] y_s[i+d]. \tag{3.39}$$

$$-\mathrm{AMDF}(d) = -\sum_i |y_s[i] - y_s[i+d]|. \tag{3.40}$$

(Dla uproszczenia w formułach periodogramów przyjmujemy, że zakres parametru i wynika z kontekstu.)

Do zalet periodogramów samopodobieństwa zalicza się relatywnie niską złożoność obliczeniową (ACF można obliczyć poprzez dwukrotne FFT) oraz odporność na addytywne zakłócenia szumowe. Do wad periodogramów samopodobieństwa zalicza się zależność od struktury formantowej. Z tego względu wraz z periodogramami korelacyjnymi zaleca się stosowanie metod wstępnej analizy sygnałowej niwelujących strukturę formantową, np. *center–clipping* albo filtrowanie odwrócone. Wiarygodność funkcji ACF, jako periodogramu spada, gdy w analizowanym sygnale występują znaczące różnice średnich amplitud kolejnych prawieokresów (Hess 1983, 355); problem ten nie występuje w przypadku periodogramu −AMDF (de Cheveigné i Kawahara 2002). W przypadku ACF występuje konieczność stosowania relatywnie długich segmentów (w zależności od rodzaju okna sygnałowego do ok. czterech maksymalnych przewidywanych T_0).

Boersma (1993) w wyniku kompensacji wpływu funkcji okienkującej sygnał segmentu na periodogram ACF uzyskał periodogram ACNF przewyższający ACF o kilka rzędów wielkości pod względem dokładności oraz odporności na addytywne zakłócenia szumowe.

$$\mathrm{ACNF}(d) = \frac{ACF(d)}{\sum_i w_s[i] w_s[i+d]}, \tag{3.41}$$

[6] Pojęcie „periodogram” używanie jest także literaturze w znaczeniu „widmo amplitudowe”.

gdzie w_s jest funkcją okienkującą użytą do otrzymania sygnału y_s. Shimamura i Kobayashi (2001) w celu zwiększenia odporności na zakłócenia szumowe proponuje autokorelację ważoną. de Cheveigné i Kawahara (2002) proponują algorytm YIN, w ramach którego stosowany jest zarówno ACF jak i AMDF. Ishi (2004) analizuje zastosowanie periodogramu autokorelacyjnego do extrakcji F_0 przy fonacji chrypliwej. Ying i inni (1996) proponuje probabilistyczną interpretację periodogramu AMDF.

3.2.2 Periodogramy cepstralne

Periodogramem cepstralnym nazywamy funkcję $L_M(C(x))$ określoną jak w równaniu 3.28 na stronie 36

Periodogram cepstralny (pomijając intuicje związane z separacją źródła i filtra) opiera się na logarytmicznej kompresji widma, która redukuje strukturę formantową sygnału mowy. Niestety równolegle z redukcją struktury formantowej następuje spadek SNR (Schroeder 2004, 242), (Hess 2008, 188). Zastąpienie w równaniu 3.23 logarytmu naturalnego pierwistkiem stopnia czwartego pozwala ograniczyć spadek SNR (Sreenivas 1982). Periodogram, który proponuje Sreenivas (1982) nie spełnia jednak założeń analizy homomorficznej.

3.2.3 Periodogramy harmoniczne

Periodogramem harmonicznym nazywamy periodogram $\mathrm{HF} : \mathbb{Z} \mapsto \mathbb{R}$, którego wartość $\mathrm{HF}(f)$ reprezentuje zagregowaną amplitudę harmonicznych o częstotliwościach nf, $n \in \mathbb{Z}$. Periodogramy harmoniczne wywodzą się z metody *histrogramu częstotliwościowego* (por. Schroeder 1968).

Podstawowa postać periodogramu harmonicznego określona jest następująco:

$$\mathrm{HPSF}(f) = \sum_i |Y_s|(fi), \tag{3.42}$$

gdzie $Y_s \leftrightsquigarrow y_s$ oraz $fi \leqslant 1250\mathrm{Hz}$ (Hermes 1988).

Sun (2002b) proponuje periodogram harmoniczny agregujący sumaryczną amplitudę podharmonicznych:

$$\mathrm{SSHF}(f) = \sum_i |Y_s|(f(i-1)/2). \tag{3.43}$$

Iloraz:

$$\mathrm{SHR}(f) = \frac{\mathrm{SSHF}(f)}{\mathrm{HPSF}(f)}, \tag{3.44}$$

pozwala na określenie rodzaju fonacji; w szczególności $\mathrm{SHR} < 0.2$ dla fonacji chrypliwej oraz $\mathrm{SHR} > 0.4$ dla fonacji modalnej. W zależności od wykrytego rodzaju fonacji stosowane są odrębne algorytmy wyznaczania rozkładu F_0.

3.2.4 Periodogramy grzebieniowe

Periodogramem grzebieniowym nazywamy periodogram $CBF : \mathbb{Z} \mapsto \mathbb{R}$, którego wartość $\mathrm{CBF}(f)$ reprezentuje *podobieństwo* widma sygnału y_s do wzorcowego widma sygnału złożonego z wielu harmonicznych częstotliwości podstawowej f.

Określmy funkcję pomocniczą:

$$\mathrm{CB}(m,p) = \begin{cases} k^{-1/s} & m = kp, k \in \mathbb{N} \\ 0 & \text{w przec. przyp.}, \end{cases} \tag{3.45}$$

gdzie parametr $s \in \mathbb{N} \wedge s > 1$. Podstawowy rodzaj periodogramu grzebieniowego określony jest następująco:

$$\mathrm{CBF(p)} = \sum_k |Y_s|(kp)\mathrm{CB}(kp,p), \tag{3.46}$$

gdzie $Y_s \leftrightsquigarrow y_s$ Martin (1982).

Periodogram grzebieniowy pozwala poprawnie określić F_0 także wtedy, gdy w sygnale wejściowym brak harmonicznej o częstotliwości F_0 (ton różnicowy). Wadą podstawowej wersji periodogramu grzebieniowego jest niska rozdzielczość częstotliwościowa (ok. 10Hz-30Hz) oraz zależność wyników od struktury formantowej widma sygnału.

Stosując interpolację widma z użyciem funkcji kwadratowych można zwiększyć rozdzielczość częstotliwościową periodogramu grzebieniowego do ok. 1Hz (Martin 1982, 1987; Brown 1992). Paliwal i Rao (1983) opisują sposób zmniejszenia zależności periodogramu grzebieniowego od struktury formantowej poprzez uzależnienie funkcji CB od obwiedni widma uzyskanej metodą LPC.

3.2.5 Periodogramy percepcyjne

Periodogramem percepcyjnym nazywamy periodogram, którego wartości są wyliczane na podstawie aktualnego[7] modelu psychoakustycznego. Przyjmuje się, że skala częstotliwościowa periodogramu percepcyjnego jest związana z wysokością tonu a nie (jak w przypadku innych rodzajów periodogramów) z częstotliwością podstawową.

Duifhuis i inni (1979) proponują **sito harmoniczne**, czyli periodogram percepcyjny oparty na modelu Goldsteina (por. Gold 1999, 221) (por. de Cheveigné 2005, 11). Danymi wejściowymi sita harmonicznego jest funkcja Y będąca wynikiem operacji uproszczonego maskowania na 64-punktowym widmie amplitudowym[8]. Zgodnie z modelem Goldsteina różnice wartości niezerowych miejsc funkcji Y są percepcyjnie nieistotne. Niech będą dane trzy funkcje pomocnicze: $k : \mathbb{N} \mapsto \mathbb{R}$, $q : \mathbb{N} \mapsto \mathbb{R}$ oraz $c : \mathbb{N} \mapsto \mathbb{R}$ określone następująco:

$$k(p) = |i \in \mathbb{N} : Y[i] \neq 0 \wedge \underset{j<K}{\exists} is \in [jp(1-b); jp(1+b)]|, \tag{3.47}$$

$$q(p) = |i \in \mathbb{N} : Y[i] \neq 0 \wedge is < jp(1+b)|, \tag{3.48}$$

$$c(p) = \lfloor \rfloor, \tag{3.49}$$

gdzie $s = 2000/64$, $K = 8$ oraz $b = 1/16$. Ostatecznie, sito harmoniczne określone jest wzorem:

$$HSF(p) = \frac{k(p)}{q(p) + c(p)}. \tag{3.50}$$

[7] Do nieaktualnych (sfalsyfikowanych) psychoakustycznych modeli percepcji częstotliwości podstawowej zalicza się periodogram ACF opisany w sekcji 3.2.1 (por. de Cheveigné 2005, 20).

[8] Zaniedbywalność widma fazowego w modelach percepcyjnych jest przedmiotem kontrowersji. Niedawne badania (Paliwal i Alsteris 2003) wykazały, że widmo fazowe ma istotnie mniejszy wpływ na percepcję mowy od widma amplitudowego, jeżeli czas trwania okna sygnałowego poddanego transformacji wynosi mniej niż 60 ms. Dla okien analizy dłuższych od 200 ms widmo fazowe jest percepcyjnie bardziej istotne niż widmo amplitudowe.

Terhardt i inni (1982) proponują periodogram percepcyjny, którego wartość reprezentuje intensywność **wirtualnego tonu podstawowego**, pojęcia psychoakustycznego wprowadzonego we wcześniejszych pracach Terherdta. Danymi wejściowymi ww. periodogramu są wektory $f[i]$ oraz $L[i]$ określające odpowiednio częstotliwość oraz poziom (SPL) znaczących (nie zamaskowanych, por. str. 37) maksimów lokalnych DFT sygnału y_s. Intensywność wirtualnego tonu podstawowego o częstotliwości f jest wyznaczana na podstawie wartości $f[i]$ takich, że $300Hz \leqslant f[i] \leqslant 3kHz$ oraz $f[i] \approx nf$, gdzie $n \in \mathbb{Z}$ (Fastl i Zwicker 2007, 123). Terhardt postawił dodatkowo hipotezę, że wirtualny ton podstawowy jest pojęciem nabywanym przez człowieka w wyniku percepcji dźwięków o bogatej strukturze harmonicznej (np. sygnału mowy). Shamma i Klein (2000) zaprezentowali wyniki świadczące przeciw tej hipotezie.

W modelu percepcji wysokości tonu SPINET (Cohen i inni 1995) wprowadzono periodogram percepcyjny:

$$SPF(p) = \sum_m [S(mp)]^+ h(m), \tag{3.51}$$

gdzie $S(mp) = S(mp, \overleftarrow{s})$ (definicję dwuargumentowej funkcji S zamieszczono w równaniu 3.38 na stronie 38),

$$[x]^+ = \begin{cases} x & \text{dla} x > 0 \\ 0 & \text{w przec. przyp.,} \end{cases} \tag{3.52}$$

$$h(m) = \begin{cases} 1 - 0.15 log_2(m) & \text{dla} 0.15 log_2(m) < 1 \\ 0 & \text{w przec. przyp..} \end{cases} \tag{3.53}$$

Cosi i inni (1998) zaproponował **korelogram zbiorczy** — periodogram percepcyjny oparty na modelu Lyona (Lyon 1982). Danymi wejściowymi korelogramu zbiorczego jest cochleagram (por. str. 38). Przyjmijmy, że y_s^j oznacza j-ty kanał cochleagramu sygnału y_s. Korelogram zbiorczy określony jest jako:

$$\mathrm{SCF}(d) = \sum_j \frac{\sum_i y_s^j[i] y_s^j[i+d]}{\sum_i w_s[i] w_s[i+d]}, \tag{3.54}$$

gdzie w_s jest przyjętym oknem sygnałowym segmentu s. Opracowano także wariant SCF dający lepsze wyniki w warunkach obniżonego SNR:

$$\mathrm{SCNRF}(d) = \mathrm{SCF}(d) - \mathrm{SCF}_{\mathrm{noise}}(d), \tag{3.55}$$

gdzie $\mathrm{SCF}_{\mathrm{noise}}$ jest uśrednionym korelogramem zbiorczym segmentów zawierających wyłącznie sygnał stacjonarnego źródła zakłóceń Cosi i inni (1998). (Ottaviani i Rocchesso 2001) opisuje zastosowanie interpolacji Lagrange'a do zwiększenia rozdzielczości częstotliwościowej periodogramu percepcyjnego opartego na modelu Lyona.

Chisaki i inni (2003) proponują transformatę periodycznościową opartą na modelu percepcji wysokości tonu, który zaproponowali Meddis i Hewitt (1991). Niech W_s oznacza harmoniczną transformatę falkową (HWT) sygnału x_s zgodnie z równaniem 3.14 na stronie 34. Określamy pomocniczą funkcję autokorelacji transformaty W_s w wymiarze czasowym jak następuje:

$$R(\tau, f) = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} W_s(t, f) W(t + \tau, f) \mathrm{d}t, \tag{3.56}$$

Transformata periodycznościowa WCF (Chisaki i inni 2003) jest otrzymywana przez całkowanie funkcji R w dziedzinie częstotliwościowej:

$$WCF(\tau) = \int_{F_L}^{F_H} R(\tau, f)\mathrm{d}f, \tag{3.57}$$

gdzie F_L oraz F_H określają pasmo cech tonalnych.

Wśród korzyści wynikających z zastosowania periodogramów percepcyjnych wymienić można: 1) symulowanie zjawisk psycholingwistycznych związanych z częstotliwością podstawową, 2) stosowalność anotacji ramkowych o krótkich czasach trwania segmentów (np. $2T_0$), z dobrym przybliżeniem spełniających założenie o stacjonarności sygnału mowy. Periodogramy percepcyjne nie zyskały jak dotąd znaczącej popularności aplikacyjnej m.in. ze względu na złożoność obliczeniową.

3.3 Suprasegmentalna sygnałowa analiza tonalna

Suprasegmentalna sygnałowa analiza tonalna obejmuje dwa etapy: 1) segmentalną sygnałową analizę tonalną oraz 2) redukcję błędu ekstrakcji F_0. Segmentalną sygnałową analizę tonalną opisano w sekcji 3.2. Dla danej sygnałowej anotacji tonalnej $A = (S, a)$ algorytm redukcji błędu ekstrakcji F_0 tworzy sygnałową anotację tonalną $A' = (S, a')$ taką, że *przeciętnie* spełnione jest $D(A', A_x^R) < D(A, A_x^R)$, gdzie D jest odległością anotacyjną a A_x^R jest anotacją referencyjną. W algorytmach redukcji błędu ekstrakcji F_0 wykorzystuje się zależności między etykietami segmentów obserwowane w anotacjach referencyjnych.

3.3.1 Wygładzanie

Niech p_i oraz p'_i oznaczają rozkłady F_0 i-tego segmentu odpowiednio anotacji wejściowej (S, a) oraz anotacji wyjściowej (S, a') w porządku $\overleftarrow{<}^S$. Wygładzanie medianowe jest metodą redukcji błędu ekstrakcji F_0 określoną następująco:

$$p'_k = \{(\mathrm{med}\{f_{max}(p_{k-N}), f_{max}(p_{k-N+1}), \ldots, f_{max}(p_{k+N})\}, 1)\}, \tag{3.58}$$

gdzie med oznacza medianę zbioru oraz $N \in 1, 2, 3$.

Ze względu na niewielkie wymagania co do anotacji wejściowej (wykorzystywana jest wyłącznie dominanta rozkładu F_0) wygładzanie medianowe może być stosowane z dowolnym algorytmem sygnałowej analizy tonalnej. Wygładzanie medianowe pozwala na redukcję błędu ekstrakcji F_0 nawet o jeden rząd wielkości (Hess 2008, 201).

3.3.2 Minimalizacja kosztu

Niech będzie dana sygnałowa anotacja tonalna A otrzymana w wyniku analizy sygnału x. Niech p_i oznacza rozkład F_0 i-tego segmentu anotacji $A = (S, a)$ w porządku $\overleftarrow{<}^S$. Dla dowolnego $A = (S, a)$ zbiór **przebiegów** F_0 określony jest jako:

$$\mathcal{F}^A = \prod_{i=0}^{|S|-1} \mathrm{domain} p_i. \tag{3.59}$$

Niech $f \in \mathcal{F}^A$. Przez A^f oznaczamy anotację uzyskaną z anotacji A poprzez zastąpienie rozkładów p_i rozkładami jednopunktowymi:

$$p'_i = \{(f[i], 1)\}. \tag{3.60}$$

Funkcją kosztu przebiegu F_0 nazywamy funkcję $c : \mathcal{F}^A \mapsto \mathbb{R}$, pozytywnie skorelowaną z błędem ekstrakcji $D(A^f, A^R_x)$, gdzie A^R_x jest anotacją referencyjną. Istotą metody minimalizacji kosztu jest zastosowanie programowania dynamicznego do znalezienia przebiegu $f' \in \mathcal{F}^A$ minimalizującym wartość $c(f')$. Jako jedni z pierwszych minimalizację kosztu przebiegu F_0 zastosowali Secrest i Doddington (1982).

Niech h_i oznacza harmoniczność i-tego segmentu anotacji $A = (S, a)$ w porządku $\overleftarrow{<}^S$. Boersma (1993) określa funkcję kosztu następująco:

$$c(f) = \sum_{i=1}^{|f|-1} c_t(i) - \sum_{i=0}^{|f|-1} p_i(f[i]), \tag{3.61}$$

gdzie

$$c_t(i) = \begin{cases} 0 & \text{dla } h_{i-1} = h_i \\ \alpha & \text{dla } h_{i-1} \neq h_i \\ \beta \left| log_2(f_{i-1}/f_i) \right| & \text{w przec. przyp..} \end{cases} \tag{3.62}$$

Boersma (1993) przyjmuje binarną skalę harmoniczności oraz ustala wartości parametrów $\alpha = \beta = 0.2$.

Talkin (1995) proponuje metodę korekty harmoniczności segmentu. Kasi i Zahorian (2002) określa koszt przejścia między wartościami harmoniczności na skali binarnej (parametr α w równaniu 3.62) w oparciu o etykiety energii segmentu. Wang i inni (2002) proponuje probabilistyczny model funkcji kosztu przebiegu F_0.

ROZDZIAŁ 4

Fonetyczna analiza tonalna

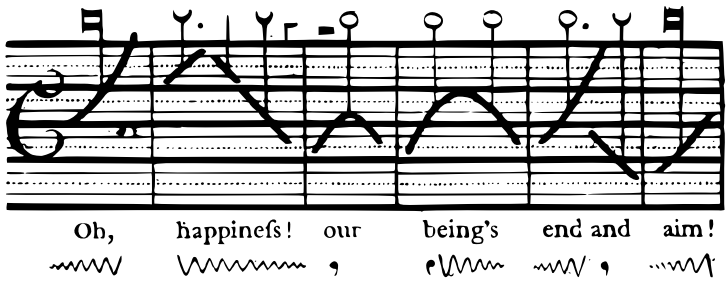
Początki obiektywizacji fonetycznej analizy tonalnej wiążą się z pracami fonetyka brytyjskiego o nazwisku Steele wykonanymi w drugiej połowie XVIII-tego wieku (Steele 1775). Steele współpracował z wybitnym aktorem szekspirowskim, który potrafił wielokrotnie powtarzać rolę z niezmienioną intonacją. Steele za pomocą bezprogowego instrumentu strunowego generował dźwięki o przebiegach wysokości tonu subiektywnie zgodnych z przebiegami wysokości tonu w mowie aktora. Następnie notował punkty początkowe, końcowe oraz zwrotne pozycji palca na gryfie instrumentu korzystając ze zmodyfikowanej pięciolinii.

Rycina 4.1 przedstawia wyniki analizy Steele'a dla przykładowego zdania w języku angielskim (fragment z „Essay on Man" Aleksandra Pope'a). Oprócz etykiet anotacji tonalnej rycina 4.1 zawiera elementy transkrypcji iloczasu (symbole nad pięciolonią, pionowe odcinki na pięciolinii) oraz elementy transkrypcji intensywności (symbole pod transkrypcją ortograficzną) zaproponowane przez Steele'a.

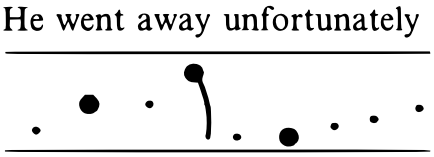
W kategoriach fonetyki informatycznej Steele wykonywał fonetyczną analizę tonalną niekotwiczącą z zadaną segmentacją sylabiczną. Etykietą sylaby w anotacji wynikowej jest wielomian niskiego stopnia (najczęściej prosta lub parabola), którego przebieg reprezentuje przebieg wysokości tonu w granicach sylaby. Punkty początkowe i końcowe przebiegów wysokości tonu reprezentowane są na dyskretnej skali logarytmicznej o rozdzielczości ćwierćtonowej.

W pierwszej połowie dwudziestego wieku, w kręgu badaczy związanych ze Szkołą Brytyjską (por. strona 5.1) upowszechniła się **tonetyczna transkrypcja międzyliniowa** (*interlinear tonetic transcription*). Tonetyczna transkrypcja międzyliniowa jest otrzymywana w wyniku analizy niekotwiczącej z zadaną segmentacją sylabiczną. Etykiety anotacyjne reprezentowane są w postaci graficznej, na którą składają się: 1) dwa poziome odcinki, o tych samych współrzędnych w wymiarze poziomym, reprezentujące dolną oraz górną granicę wysokości tonu danego mówcy w fonacji modalnej, 2) koło o jednej z wartości średnicy; mniejsza średnica dla sylab bez akcentu realnego oraz większa średnica dla sylab z akcentem realnym, lokalizacja pionowa koła względem poziomych odcinków odpowiada wysokości tonu na początku sylaby, 3) (opcjonalnie) krzywa obrazująca przebieg wysokości tonu w granicach sylaby.

Rycina 4.2 zawiera przykładową tonetyczną transkrypcję międzyliniową sygnału mowy angielskiej. Tonetyczną transkrypcję międzyliniową zastosowali w swoich pracach m.in. O'Connor



Rycina 4.1: Przykładowa anotacja fonetyczna Steele'a (1775).



Rycina 4.2: Przykładowa tonetyczna transkrypcja międzyliniowa (Cruttenden 1997, 36).

i Arnold (1973), Cruttenden (1997) i Wells (2006). Transkrypcje zamieszczone w cytowanych pracach zostały otrzymane w wyniku analizy subiektywnej. W sekcji 8.3 zaproponowano obiektywny algorytm tworzenia tonetycznej transkrypcji międzyliniowej z sygnału mowy polskiej.

W latach 70-tych, w ośrodkach IPO (*Institute for Perception Research*, Eindhoven) oraz ERI (*Engineering Research Institute*, Uniwersytet Tokijski) prowadzono niezależnie badania nad obiektywizacją fonetycznej analizy tonalnej (por. 't Hart 1979; Fujisaki i Hirose 1984). W stosunku do wcześniejszych, powstałe wtedy algorytmy analizy wyróżniały się: 1) zastosowaniem układów ekstrakcji F_0, 2) aproksymacją przebiegu F_0 matematycznymi modelami wybranych układów biologicznych (percepcji F_0 w IPO oraz fonacji w ERI), 3) zapisem w etykietach anotacji parametrów funkcji aproksymującej, 4) odwracalnością analizy (por. strona 13), 5) intersubiektyną weryfikacją wyników. W sekcjach 4.1 oraz 4.3 przedstawiono algorytmy analizy wywodzące się z prac IPO oraz ERI.

Po roku 1990 zainteresowanie obiektywnymi algorytmami fonetycznej analizy tonalnej znacznie wzrosło m.in. ze względu na zastosowania w korpusowej syntezie mowy. W dalszej części niniejszego rozdziału opisano obiektywne algorytmy analizy fonetycznej rozwijane po roku 1990.

4.1 IPO

Dane wejściowe sygnał mowy
Anotacja wyjściowa segmentalna, fonetyczna, percepcyjna, prosta, zakotwiczona

't Hart (1976) oraz 't Hart i inni (1990) opisują intersubiektywny, odwracalny algorytm tonalnej analizy fonetycznej znany obecnie jako algorytm/model IPO (algorytm/model *Institute for Perception Research* w Eindhoven). Algorytm analizy IPO aproksymuje przebieg F_0 funkcją przedziałami liniową o minimalnej liczbie przedziałów. Scheffers (1988) przedstawił pierwszy obiektywny algorytm analizy IPO. Bagshaw (1993) oraz Spaai i inni (1996) wprowadzili dalsze ulepszenia a najnowszą wersję algorytmu IPO opisał Hermes (2006).

Anotacja fonetyczna IPO oparta jest na segmentacji prostej z etykietami w postaci par liczb rzeczywisych. Etykieta segmentu s w anotacji IPO określa współczynniki funkcji liniowej, która aproksymuje przebieg F_0 w granicach segmentu s na skali półtonów lub ERB-ów. Bagshaw (1993) wprowadził warunek ciągłości przebiegu wysokości tonu, tj. by dla każdej pary segmentów $s_0, s_1 \in S$ takich, że $\overrightarrow{s_0} = \overleftarrow{s_1} = t$ zachodziło:

$$a_0 + b_0 t = a_1 + b_1 t, \tag{4.1}$$

gdzie (S, a) jest dowolną anotacją IPO, $(a_0, b_0) = \overline{s_0}$ oraz $(a_1, b_1) = \overline{s_1}$.

Dla danej anotacji IPO (S, a) przebieg czasowy F_0 określony jest wzorem:

$$f_{IPO}(t) = \begin{cases} a_0 + b_0 t & \text{dla } \overleftarrow{S[0]} \leqslant t < \overrightarrow{S[0]} \\ a_1 + b_1 t & \text{dla } \overleftarrow{S[1]} \leqslant t < \overrightarrow{S[1]} \\ \dots & \\ a_{|S|-1} + b_{|S|-1} t & \text{dla } \overleftarrow{S[|S|-1]} \leqslant t < \overrightarrow{S[|S|-1]} \end{cases} \tag{4.2}$$

gdzie $(a_i, b_i) = \overline{S[i]}$. Jednostką skali przeciwdziedziny funkcji f_{IPO} jest ERB.

Algorytm analizy IPO obejmuje 4 etapy:

1. ekstrakcję F_0 w oparciu o periodogram harmoniczny SHS (Hermes 1988) oraz algorytm minimalizacji kosztu (por. sekcja 3.2.3 oraz 3.3.2); anotację otrzymaną w wyniku ekstrakcji F_0 oznaczamy przez F,

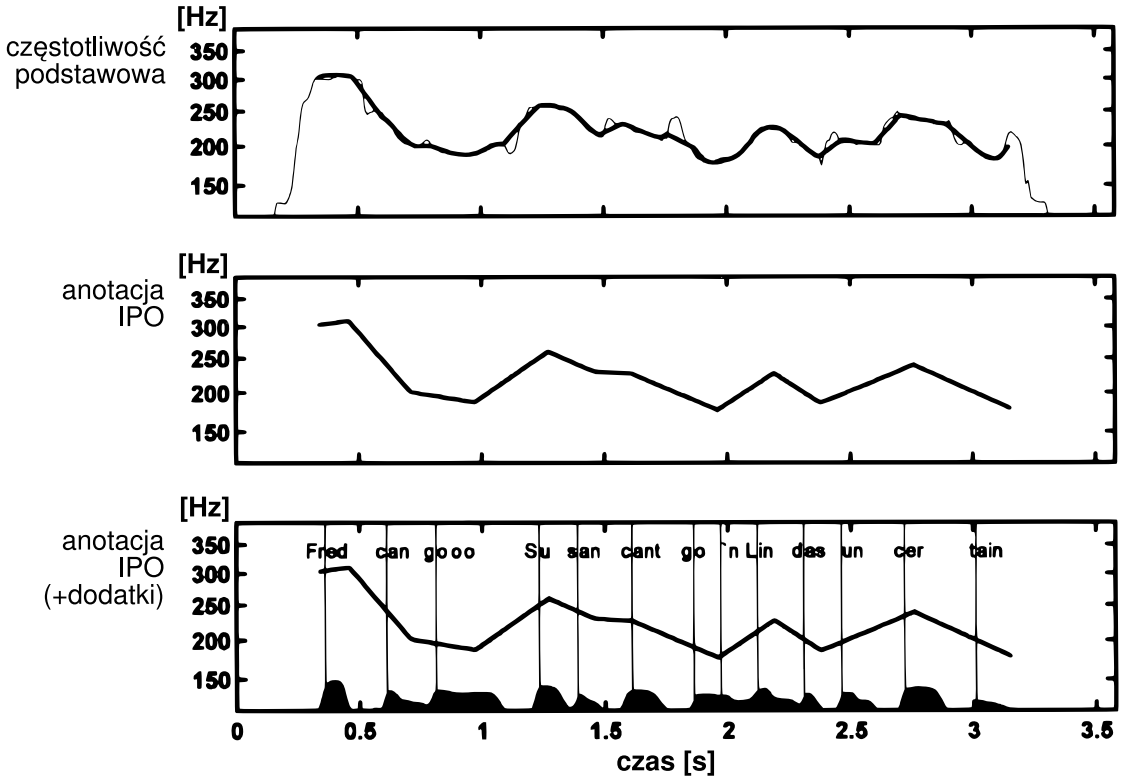
2. ekstrakcję *samogłoskowości* (*vowel strength*), wynikiem której jest anotacja segmentalna V złożona z segmentacji anotacji F etykietowanej energią harmonicznych sygnału segmentu występujących w pobliżu formantów F_1 i F_2.

3. redukcję mikrointonacji poprzez wygładzanie przebiegu etykiet w anotacji F średnią ważoną wartościami etykiet anotacji V z zastosowaniem filtra drugiego rzędu o nieznacznym wzmocnieniu w otoczeniu 3.5Hz.

4. procedurę *stylizacji* IPO.stylize(F, V), tj. algorytm A.1.

Na rys. 4.3 przedstawiono wykresy danych wejściowych oraz wyjściowych algorytmu A.1 dla przykładowej wypowiedzi w języku angielskim («Fred can go, Susan can't go, and Linda's uncertain.»/«Fred może iść, Susan nie może iść a Linda jest niezdecydowana.»). Górny wykres przedstawia dwa przebiegi F_0: 1) z ekstraktora F_0 (cieńsza kreska) oraz 2) wygładzony F_0 (grubsza kreska). Środkowy wykres przedstawia anotację IPO. Dolny wykres przedstawia anotację IPO wraz z transkrypcją ortograficzną wypowiedzi, granicami samogłosek (pionowe odcinki) oraz przebiegiem samogłoskowości (bezpośrednio nad osią czasu).

W przypadku intersubiektywnego algorytmu IPO testowanie odwracalności reprezentacji (*perceptual equality*) jest jednym z etapów analizy ('t Hart i inni 1990, 42). W przypadku algorytmu A.1 trudno jest mówić o odwracalności bez przyjęcia konkretnych wartości progów α, β, γ. Przy odpowiednio niskich progach możliwa jest dowolnie dokładna resynteza kosztem zwiększenia liczby segmentów w anotacji. Nie znaleziono danych na temat wartości parametrów algorytmu A.1, dla których algorytm A.1 tworzyłby anotacje typu *close-copy* zdefiniowane w intersubiektywnej analizie IPO ('t Hart i inni 1990, 43).



Rycina 4.3: Wizualizacja anotacji IPO przykładowego sygnału mowy w języku angielskim (Hermes 2006, 37).

Algorytm IPO zastosowano do analizy cech tonalnych m.in. w językach niderlandzkim, angielskim oraz rosyjskim. Do zalet analizy IPO zaliczamy: 1) uniwersalność językową, 2) prostą interpretację etykiet oraz 3) uwzględnianie wpływu energii i harmoniczności sygnału na percepcję F_0. Do wad analizy IPO zaliczamy: 1) założenie o ciągłości przebiegu F_0, 2) potencjalnie wysoka średnia liczba segmentów w przeliczeniu na jednostkę czasową lub sylabę, 3) dopuszczanie praktycznie dowolnych przebiegów F_0 (przy założeniu swobodnego doboru parametrów α, β, γ).

4.2 MOMEL

Dane wejściowe sygnał mowy
Anotacja wyjściowa segmentalna, fonetyczna, ciągła, matematyczna, zakotwiczona

Hirst i Espesser (1993) przyjmują model, w którym obserwowany przebieg F_0 jest złożeniem przebiegów dwóch funkcji: 1) (makro)intonacyjnej oraz 2) mikrointonacyjnej. Hirst i Espesser (1993) proponują, by funkcję makrointonacyjną aproksymować funkcją sklejaną drugiego stopnia (przedziałami-paraboliczną funkcją różniczkowalną). W związku z zastosowaniem funkcji sklejanej drugiego stopnia przyjęto założenie o ciągłości oraz określoności funkcji makrointonacyjnej w każdym punkcie modelowanego przedziału czasowego. Nieciągłości, nieokreśloność fragmentów przebiegu F_0 oraz część zmian F_0 zachodzących w niewielkiej rozciągłości czasowej ($<$200ms) obejmuje funkcja mikrointonacyjna, której Hirst i Espesser (1993) nie modelują jawnie. Hirst i inni (2000) przedstawili metodę MOMEL (fr. *MELodic MOdelisation*) będącą rozwinięciem modelu opisanego powyżej.

Anotacja fonetyczna MOMEL składa się z segmentacji prostej zakotwiczonej S oraz funkcji etykietującej o przeciwdziedzinie $\mathbb{R} \times \mathbb{R}$ takich, że dla każdego $s \in S$ zachodzi:

$$\overleftarrow{s} \leqslant t \leqslant \overrightarrow{s}, \tag{4.3}$$

gdzie $(t, h) = \overline{s}$. Ponadto dla każdej pary $\{s_0, s_1\} \in S$ takiej, że $s_0 \prec^S s_1$ spełnione jest

$$\overrightarrow{s_0} = \frac{t_0 + t_1}{2}, \tag{4.4}$$

gdzie $(t_0, h_0) = \overline{s_0}$ oraz $(t_1, h_1) = \overline{s_1}$. Etykieta (t, h) reprezentuje odpowiednio czas oraz wartość (skala liniowa) maksimum lokalnego funkcji sklejanej.

Przyjmijmy oznaczenia jak we wzorze 4.4 oraz ponadto niech będzie dany segment $s_2 \in S$ taki, że $s_1 \prec^S s_2$. Dla danej anotacji MOMEL $M = (S, a)$ przebieg czasowy F_0 określony jest następująco:

$$f_M(t) = \begin{cases} h_1 + 2(t - t_1)^2(h_0 - h_1)/(t_0 - t_1)^2 & t \leqslant t_1 \\ h_1 + 2(t - t_1)^2(h_2 - h_1)/(t_2 - t_1)^2 & \text{w przec. przyp.,} \end{cases} \tag{4.5}$$

gdzie $s_1 = S(t)$ oraz $(t_2, h_2) = \overline{s_2}$ (Rolland 2000).

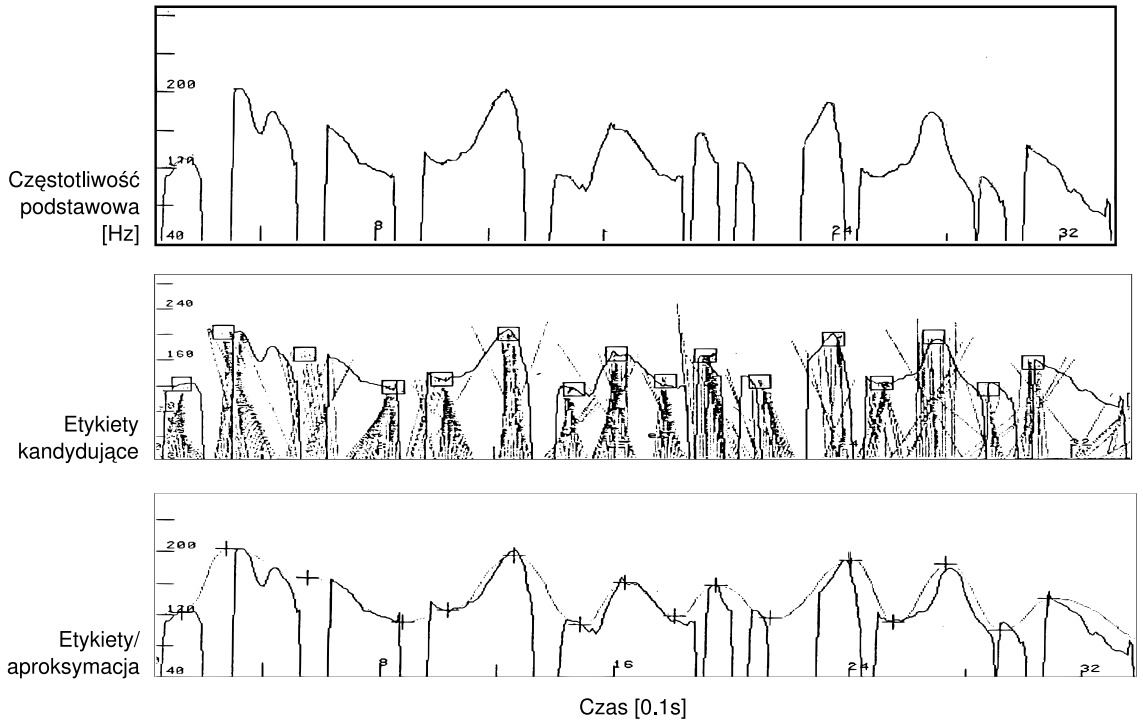
Algorytm analizy MOMEL (Hirst i inni 2000) obejmuje 5 etapów:

1. ekstrakcję F_0 — zaimplementowaną z użyciem periodogramów samopodobieństwa AMDF i ACF (por. sekcja 3.2.1) oraz periodogramu grzebieniowego (por. sekcja 3.2.4) (Hirst i inni 2000, 67),
2. redukcję mikrointonacji w anotacji sygnałowej — por. algorytm A.2,
3. wyznaczenie etykiet kandydujących — por. algorytm A.3,
4. wyznaczenie segmentacji — por. algorytm A.4,
5. wyznaczenie etykiet — odrzucenie etykiet-kandydatów o wartościach dalekich od wartości średniej w obrębie segmentu oraz ustalenie etykiety jako wartości średniej z pozostałych etykiet-kandydatów.

Etap 3 algorytmu MOMEL oparty jest na *regresji modalnej* — technice przypominającej regresję, w której zamiast wartości oczekiwanej stosowana jest moda[1] Hirst i inni (2000, 62).

Na rys. 4.4 przedstawiono na przykładzie wypowiedzi w języku francuskim («La fille de Charles Sablon a voulu un petit chien en guise de cadeau.», tłum.: «Córka Charles'a Sablon chciała dostać małego psa w prezencie.») trzy wykresy danych wykorzystywanych w algorytmie MOMEL. Na górnym wykresie znajduje się przebieg F_0. Na środkowym wykresie przedstawiono dane pośrednie oraz wynikowe algorytmu A.3: odcinki łączące środek okna regresji z parą wynikową regresji (t, h) oraz etykiety kandydujące (wewnątrz prostokątów). Na dolnym wykresie znajdują się pozycje etykiet anotacji wynikowej (znak plus) oraz przebieg funkcji sklejanej drugiego stopnia (linia przerywana).

[1] Przez modę rozkładu Hirst i inni (2000) rozumie się taką wartość r, która maksymalizuje liczbę punktów rozkładu w przedziale $(r - \Delta; r + \Delta)$, gdzie Δ jest ustalonym parametrem. Dla danego rozkładu oraz ustalonego parametru Δ może istnieć więcej niż jedna moda.



Rycina 4.4: Wizualizacja anotacji MOMEL przykładowego sygnału mowy w języku francuskim (Hirst i Espesser 1993).

Ewaluację algorytmu MOMEL przeprowadzono na podzbiorze korpusu mowy EUROM1 (Chan i inni 1995). Zbiór testujący zawierał sygnał mowy 10-ciu mówców (5 kobiet, 5 mężczyzn) dla każdego z trzech języków (angielski, niemiecki oraz francuski), czytających od 10 do 20 tekstów złożonych z pięciu znaczeniowo powiązanych ze sobą zdań (Hirst i inni 2000, 68). Średni dystans jaki otrzymano między przebiegiem F_0 a przebiegiem funkcji sklejanej otrzymanej w wyniku analizy MOMEL wyniósł 5.60% dla języka angielskiego, 4.66% dla języka niemieckiego oraz 6.00% dla języka francuskiego.

Algorytmem MOMEL analizowano sygnał mowy szeregu języków europejskich (m.in. angielskiego, niemieckiego, francuskiego,włoskiego, hiszpańskiego oraz szwedzkiego). Do zalet analizy MOMEL należą: 1) uniwersalność językowa, 2) minimalne wymagania co do danych wejściowych (analizie można poddać dowolny fragment sygnału mowy), 3) eliminowanie wpływu mikrointonacji. Wśród wad analizy MOMEL wymienić można: 1) brak uwzględnienia różnic w *istotności* poszczególnych fragmentów konturu intonacyjnego, 2) nieznormalizowaną skalę F_0. W roku 2000 MOMEL z nieznacznymi modyfikacjami został reimplementowany w środowisku Praat (Rolland 2000).

4.3 Fujisaki

Dane wejściowe sygnał mowy
Anotacja wyjściowa suprasegmentalna, fonetyczna, artykulacyjna, ciągła, zakotwiczona

Fujisaki i Nagashima (1969) proponuje model, w którym przebieg F_0 jest interpretowany w kategoriach *komend motorycznych* dwóch mięśni krtani: 1) *pars obliqua musculi cricothyreoidei* oraz 2) *pars recta musculi cricothyreoidei*. Zgodnie z modelem Fujisaki pierwszy z

mięśni odpowiada za tonalność i sterowany jest **komendami frazowymi** natomiast drugi z mięśni odpowiada za toniczność i sterowany jest **komendami akcentowymi**. Przyjmuje się, że komendy frazowe oraz komendy akcentowe są od siebie niezależne. Komendy frazowe reprezentowane są w postaci impulsów dyskretnych o określonej wielkości; komendy akcentowe w postaci odcinków z przypisaną wielkością skalarną.

Anotacja proponowana dla modelu Fujisaki zawiera 1) ścieżkę segmentów komend frazowych, 2) zbiór nie zachodzących na siebie segmentów komend akcentowych oraz 3) segment linii bazowej (przesunięcia F_0 uwarunkowanego pozajęzykowo) obejmujący czasowo wszystkie pozostałe segmenty. Segmenty należące do ścieżki komend frazowych reprezentują komendę frazową w punkcie lewej kotwicy segmentu. Skalarne etykiety segmentów reprezentują wielkości komend frazowych, akcentowych oraz linii bazowej określone w modelu Fujisaki. Proponowana anotacja jest zakotwiczona oraz ciągła.

Wzór 4.6 definiuje zależność między przebiegiem częstotliwości podstawowej oraz fonetyczną anotacją tonalną modelu Fujisaki $U = (S_U, a_U)$ (Fujisaki 2000, 2004).

$$\ln f_U(t) = \ln \overline{s_b} + \underbrace{\sum_{s \in P} \overline{s} G_p(t - \overleftarrow{s})}_{\text{skladowa frazowa}} + \underbrace{\sum_{s \in A} \overline{s} \left[G_a(t - \overleftarrow{s}) - G_a(t - \overrightarrow{s}) \right]}_{\text{skladowa akcentowa}}, \tag{4.6}$$

gdzie s_b jest segmentem linii bazowej, $P \subset S_U$ jest zbiorem segmentów reprezentujących komendy frazowe, $A \subset S_U$ jest zbiorem segmentów reprezentujących komendy akcentowe,

$$G_p(t) = \begin{cases} \alpha^2 t e^{-\alpha t}, & \text{dla } t \geqslant 0 \\ 0, & \text{w przec. przyp.} \end{cases} \tag{4.7}$$

oraz

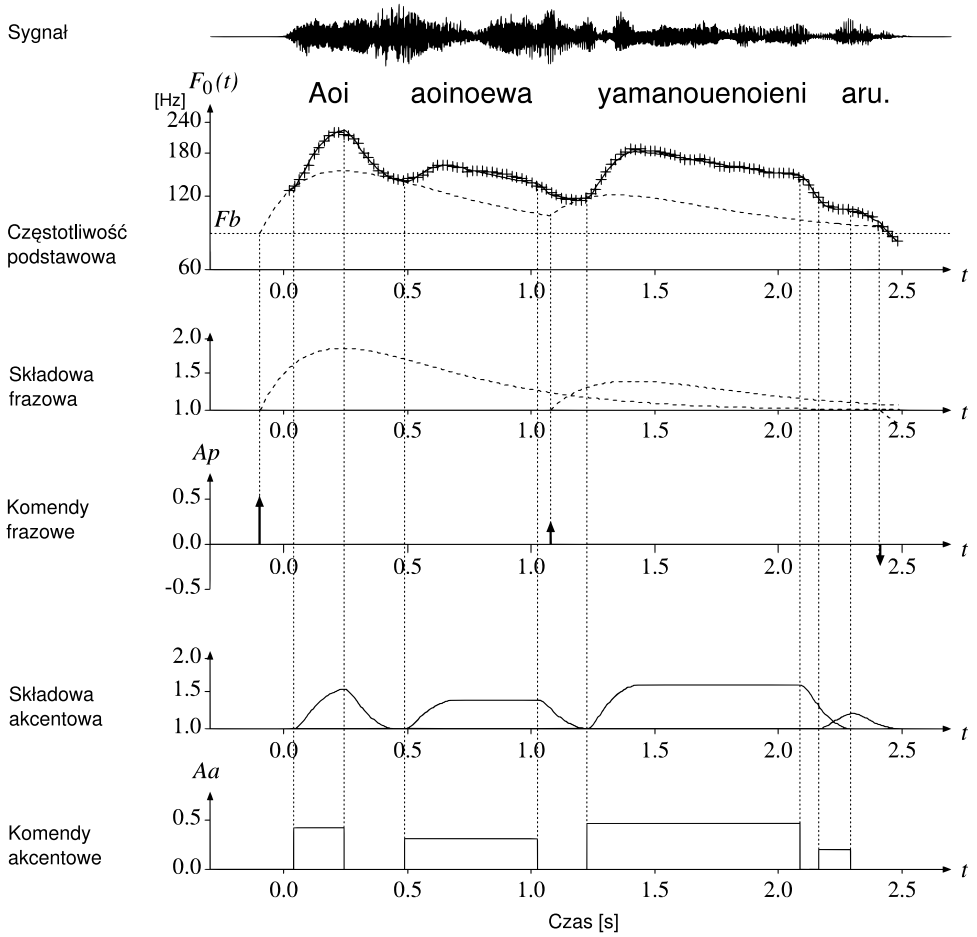
$$G_a(t) = \begin{cases} \min\left[1 - (1 + \beta t) e^{-\beta t}, 0.9\right] & \text{dla } t \geqslant 0 \\ 0 & \text{w przec. przyp.} \end{cases} \tag{4.8}$$

Przyjmuje się, że parametry α oraz β we wzorach 4.7 oraz 4.8 są stałe w obrębie komunikatu językowego (Fujisaki 2000, 5). Zgodnie z intencjami Fujisaki cechy tonalne pozajęzykowe reprezentowane są wyłącznie w postaci parametrów $\overline{s_b}$, α oraz β; komendy frazowe oraz akcentowe reprezentują natomiast cechy lingwistyczne. Zastosowanie skali logarytmicznej dla F_0 Fujisaki uzasadnia anatomią fałdów głosowych Fujisaki (2000, 3).

W początkowym okresie analiza za pomocą modelu Fujisaki wykonywana była pół-automatycznie metodą analizy przez syntezę. Mixdorff (2000) zaproponował obiektywny algorytm fonetycznej analizy tonalnej w oparciu o model Fujisaki (por. algorytm A.5).

Na rycinie 4.5 poddano wizualizacji dane pośrednie analizy za pomocą modelu Fujisaki przykładowej wypowiedzi w języku japońskim («obraz niebieskiej malwy jest w domu na szczycie wzgórza»).

Niech $F = (S_F, a_F)$ jest tonalną anotacją sygnałową należącą do pewnego korpusu mowy. Przyjmując, że $U = \text{MixdorffFujisaki}(F)$ oraz mając określone f_U jak we wzorze 4.6 definiujemy tonalną anotację sygnałową $F' = (S_F, a'_F)$ w taki sposób, że dla każdego $s \in S_F$ zachodzi $a'_F(\#s) = f_U(\overleftarrow{s})$. Mixdorff (2000) podaje, że średni błąd liczony między etykietami anotacji F oraz F' dla korpusu, który zgormadził Rapp (1998b) wynosi 3.1% dla segmentów dźwięcznych, oraz 1.7% dla segmentów dźwięcznych obejmujących stacjonarne widma samogłoskowe.



Rycina 4.5: Wizualizacja anotacji Fujisaki przykładowego sygnału mowy w języku japońskim (Fujisaki 2004).

Model Fujisaki został zastosowany m.in. do języków: angielskiego, japońskiego, niemieckiego, koreańskiego i hiszpańskiego (Fujisaki 2004). Szczególną popularność model Fujisaki znalazł w analizie języków japońskiego oraz niemickiego oraz ich dialektów, np. szwajcarskiego (Leemann i Siebenhaar 2008). Do zalet analizy Fujisaki należą: 1) stosowalność metody do dowolnego języka, 2) niewielkie wymagania dla danych wejściowych. Do wad analizy Fujisaki należą: 1) założenie, że sygnał wejściowy obejmuje pojedynczą frazę intonacyjną, a więc jednostkę pochodzącą z poziomu fonologicznego, 2), etykiety poszczególnych segmentów mają wpływ na duże części konturu (podatność na propagację błędów analizy), 3) niejasny status interpretacji artykulacyjnej. Pod koniec lat 90-tych zaproponowano Ogólny Superpozycyjny Model Intonacji (*General Superpositional Intonation Model*), w którym pomimo wielu podobieństw do modelu Fujisaki odstąpiono od artykulacyjnej interpretacji anotacji (van Santen i inni 1998; Mishra i inni 2006).

4.4 Tilt

Dane wejściowe	sygnał mowy, zakotwiczona sekwencyjna anotacja fonologiczna
Anotacja wyjściowa	segmentalna, fonetyczna, matematyczna, nieciągła, zakotwiczona

Taylor (2000) **zdarzeniem intonacyjm** nazywa przebieg F_0 towarzyszący sylabie akcentowanej melodycznie (zdarzenie nazwane `'a'`) lub końcowi frazy intonacyjnej (zdarzenie

nazwane `'b'`). Jeśli przebieg F_0 jednocześnie spełnia kryteria dla zdarzenia `'a'` oraz zdarzenia `'b'`, to mówi się o wystąpieniu zdarzenia `'ab'`. Algorytm Tilt (Taylor 1995a, 2000) oparty jest na aproksymacji przebiegów F_0 w granicach zdarzeń intonacyjnych za pomocą funkcji sklejanej stopnia drugiego złożonej z dwóch lub czterech parabol. W bieżącej sekcji zakłada się, że zdarzenia intonacyjne dla sygnału poddawanego analizie dane są w postaci nieciągłej sekwencyjnej anotacji fonologicznej (S, a), gdzie $a : \mathbb{Z} \mapsto \{\texttt{'a'}, \texttt{'b'}, \texttt{'ab'}\}$.[2]

Wyjściowa anotacja fonetyczna Tilt oparta jest na danej na wejściu segmentacji S (z nieznacznymi przesunięciami kotwic). Etykiety anotacji wyjściowej mają postać $(h, A, tilt)$, gdzie $h \in \mathbb{R}$ [Hz] reprezentuje F_0 w miejscu lewej kotwicy segmentu, $A \in \mathbb{R}$ [Hz] reprezentuje sumę wzrostu oraz spadku F_0 w granicach segmentu a $tilt \in \mathbb{R}$ określa kształt przebiegu funkcji aproksymującej. Zakłada się przy tym, że przebieg F_0 w obrębie zdarzenia intonacyjnego może być rosnący, opadający albo rosnąco-opadający.

Przyjmijmy, że $T = (S_T, a_T)$ jest anotacją fonetyczną Tilt. Skonstruujemy teraz funkcję f_T przebiegu F_0 reprezentowaną przez anotację T. Weźmy dowolny segment $s \in S_T$ oraz niech $(h, A, tilt) = \overline{s}$. Taylor (2000, 17) proponuje by na podstawie czasu trwania segmentu s oraz pary $(A, tilt)$ liczyć parametry modelu RFC następująco:

$$A_{rise} = A(1 + tilt)/2, \tag{4.9}$$

$$A_{fall} = A(1 - tilt)/2, \tag{4.10}$$

$$D_{rise} = (\overrightarrow{s} - \overleftarrow{s})(1 + tilt)/2, \tag{4.11}$$

$$D_{fall} = (\overrightarrow{s} - \overleftarrow{s})(1 - tilt)/2. \tag{4.12}$$

Parametry A_{rise} oraz A_{fall} oznaczają odpowiednio amplitudę wzrostu oraz amplitudę spadku F_0. Parametry D_{rise} oraz D_{fall} oznaczają czasy trwania odpowiednio wzrostu oraz spadku F_0. Równania 4.9-4.12 zachodzą przy założeniu, że:

$$\frac{|A_{rise}|}{D_{rise}} = \frac{|A_{fall}|}{D_{fall}} \tag{4.13}$$

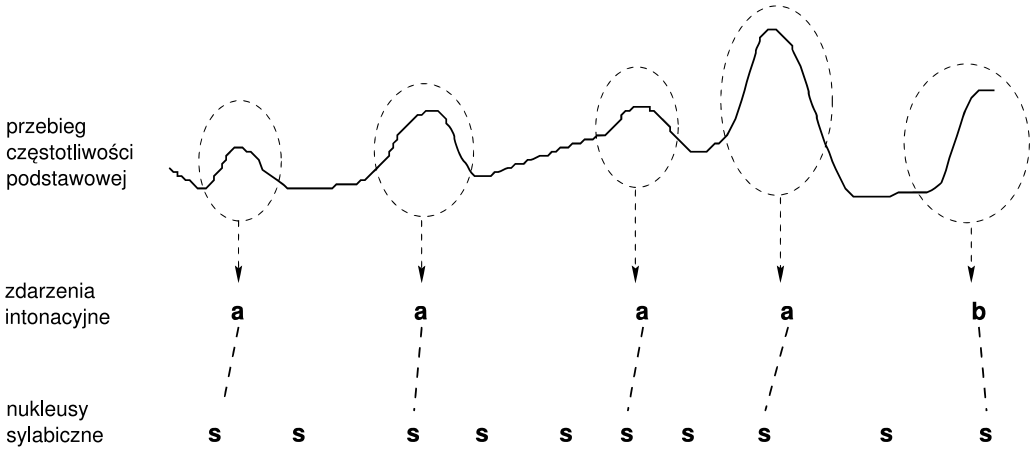
(Taylor 2000, 17). Mając dane A_{rise}, A_{fall}, D_{rise}, D_{fall} oraz h przebieg F_0 w przedziale $[\overleftarrow{s}\,;\overrightarrow{s}]$ określa się następująco:

$$f(t;s) = \begin{cases} h + A_{rise} - 2A_{rise}\left(\frac{t - \overleftarrow{s}}{D_{rise}}\right)^2 & \text{dla } \overleftarrow{s} \leqslant t \leqslant \overleftarrow{s} + \frac{D_{rise}}{2} \\ h + 2A_{rise}\left(1 - \frac{t - \overleftarrow{s}}{D_{rise}}\right)^2 & \text{dla } \overleftarrow{s} + \frac{D_{rise}}{2} < t \leqslant \overleftarrow{s} + D_{rise} \\ f(\overleftarrow{s} + D_{rise}; s) + A_{fall} - 2A_{fall}\left(\frac{t - \overleftarrow{s}}{D_{fall}}\right)^2 & \text{dla } \overleftarrow{s} + D_{rise} < t \leqslant \overrightarrow{s} - \frac{D_{fall}}{2} \\ f(\overleftarrow{s} + D_{rise}; s) + 2A_{fall}\left(1 - \frac{t - \overleftarrow{s}}{D_{fall}}\right)^2 & \text{dla } \overrightarrow{s} - \frac{D_{fall}}{2} < t \leqslant \overrightarrow{s} \end{cases} \tag{4.14}$$

Wartości F_0 poza granicami segmentów należących do S_T uzyskiwane są metodą interpolacji liniowej. Ostatecznie funkcję f_T zapisujemy jako:

$$f_T(t) = \begin{cases} f(t;s) & \text{dla } s \in S_T : \overleftarrow{s} \leq t \leqslant \overrightarrow{s} \\ f(\overrightarrow{s_1}; s_1) + \frac{f(\overleftarrow{s_2}; s_2) - f(\overrightarrow{s_1}; s_1)}{\overleftarrow{s_2} - \overrightarrow{s_1}}(t - \overrightarrow{s_1}) & \text{dla } s_1, s_2 \in S_T : s_1 \prec s_2 \wedge \overrightarrow{s_1} < t < \overleftarrow{s_2} \end{cases} \tag{4.15}$$

[2] W sekcji 5.2.2 na stronie 78 opisany jest algorytm analizy fonologicznej, którego wynikiem jest anotacja wejściowa algorytmu Tilt.



Rycina 4.6: Wizualizacja anotacji Tilt przykładowego sygnału mowy w języku angielskim (Taylor 2000).

Algorytm analizy Tilt obejmuje (Taylor 2000, 9):

1. ekstrakcję F_0 z zastosowaniem periodogramu SRPD, który zaproponowali Medan i inni (1991) oraz Bagshaw i inni (1993),
2. (opcjonalnie) sygnałową suprasegmentalną analizę tonalną metodą, którą opisał Secrest i Doddington (1982),
3. analizę RFC (por. algorytm A.8),
4. wyliczenie etykiet Tilt na podstawie etykiet RFC.

Etykiety Tilt wyliczane są z etykiet RFC przy użyciu następujących zależności:

$$tilt = \frac{|A_{rise}| - |A_{fall}|}{2(|A_{rise}| + |A_{fall}|)} + \frac{D_{rise} - D_{fall}}{2(D_{rise} + D_{fall})}, \tag{4.16}$$

$$A = |A_{rise}| + |A_{fall}|. \tag{4.17}$$

Wstosunku do etykiet RFC, etykiety Tilt odznaczają się niższą korelacją między parametrami (Taylor 2000, 16). Najmocniej skorelowane parametry RFC to A_{rise} z A_{fall} ($\rho = -0.48$) oraz A_{rise} z A_{fall} ($\rho = -0.46$). Najmocniej skorelowanymi parametrami Tilt są A z $\overrightarrow{s} - \overleftarrow{s}$ ($\rho = (0.17)$.

Na rys. 4.6 przedstawiono poglądową anotację Tilt. Na górnym wykresie znajduje się hipotetyczny przebieg F_0 wraz z oznaczeniem granic zdarzeń instonacyjnych. W środkowej i dolnej części ryciny zamieszczone są odpowiednio etykiety fonologiczne zdarzeń intonacyjnych oraz ośrodki sylabiczne.

Odwracalność algorytmu analizy Tilt przetestowano na 1061 wypowiedziach wybranych z korpusu mowy DCIEM (Bard i inni 1996). Za metryki odwracalności przyjeto dystans RMSE oraz korelację między anotacją otrzymaną w wyniku ekstrakcji F_0 (opcjonalnie z wygładzaniem medianowym) a anotacją sygnałową uzyskaną z anotacji Tilt wzorem 4.14 (Taylor 2000, 18). Badania prowadzono przy zastosowaniu obietywnych oraz intersubiektywnych wejściowych anotacji fonologicznych. Najniższy średni dystans RMSE osiągnięto dla wygładzonego

przebiegu F_0 oraz intersubiektywnej anotacji fonologicznej (RMSE=7.14). Najwyższą średnią korelację osiągnięto dla wygładzonego przebiegu F_0 oraz obiektywnej anotacji fonologicznej ($\rho = 0.833$).

Rojc i inni (2005) proponują efektywne obliczeniowo wersje algorytmów Tilt.maximizeRise oraz Tilt.maximizeFall oparte na naprzemiennej minimalizacji SSE F_0 dwiema metodami: 1) wzorem jawnym otrzymanym przez różniczkowanie SSE względem parametrów amplitudowych RFC oraz 2) metodą największego spadku SSE względem parametrów czasowych RFC.

Demenko i Wagner (2006) analizują pół-automatycznie za pomocą modelu Tilt niewielki korpus mowy czytanej języka polskiego (15 minut nagrań, jeden mówca). W celu zlokalizowania zdarzeń intonacyjnych przeprowadzono badania intersubiektywne (5-ciu respondentów), w wyniku których określono, które z sylab w korpusie mają akcent realny. Granice sylab w mowie określono na podstawie tekstu ortograficznego, korzystając z układu automatycznej transkrypcji fonetycznej (Wypych i inni 2003), układu automatycznej sylabizacji oraz pół-automatycznego układu segmentacji mowy. Dodatkowo, wprowadzono dwie zmiany w algorytmie Tilt: 1) wprowadzono zdarzenia intonacyjne typu 'ac' dla *akcentów równych*, tj. akcentów padających na sylabie pozbawionej zauważalnych zmian F_0[3], 2) wprowadzono dodatkowe zdarzenia intonacyjne w granicach sylab po-akcentowanych (sylab następujących po sylabie akcentowanej) ale tylko w przypadku, gdy sylaby akcentowana i po-akcentowana należą do tej samej *grupy akcentowej* (Demenko i Wagner 2006). Pojęcie grupy akcentowej oparto na strukturze składniowej, którą otrzymano automatycznie z tekstu ortograficznego układem analizy składniowej PolEng Jassem (2002a). Ewaluację procedury stylizacji przeprowadzono w oparciu o test subiektywny z trzystopniową skalą podobieństwa przebiegu wysokości tonu: 1) identyczny, 2) nieco różny, 3) bardzo różny. Spośród 400 nagrań poddanych testowi 256 (64%) otrzymało notę 1, 68 (17%) notę 2 a 76 (19%) notę 3. Ze względu na różnice w metodologii pomiaru trudno wyniki te odnieść do wyników Taylora lub Rojca.

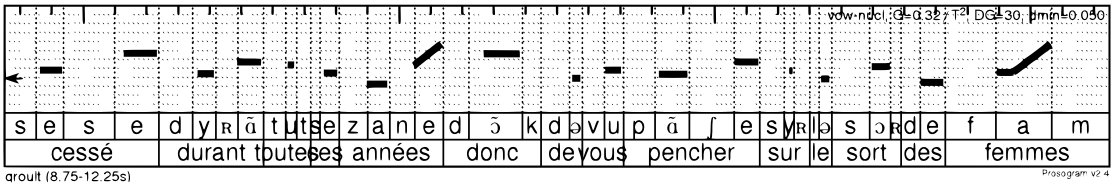
Algorytm Tilt zastosowano do szeregu języków, m.in. angielskiego, hiszpańskiego, włoskiego, słoweńskiego oraz polskiego. Do zalet algorytmu Tilt należą: 1) niskie wymagania co do danych wejściowych (uwzględniając możliwość automatycznego uzyskania wejściowej anotacji fonologicznej), 2) niska korelacja między zmiennymi w etykietach, 3) interfejs do poziomu fonologicznego. Do głównych wad algorytmu Tilt zaliczyć można: 1) oparcie tonalnej analizy fonetycznej na tonalnej anotacji fonologicznej, 2) równoważne traktowanie segmentów przebiegu F_0 niezależnie od harmoniczności oraz energii sygnału, 3) różny sposób kodowania przebiegów rosnąco-opadających (jedno zdarzenie intonacyjne) oraz opadająco-rosnących (dwa zdarzenia intonacyjne).

4.5 Prosogram

Dane wejściowe	sygnał mowy oraz opcjonalnie zakotwiczona anotacja głoskowa albo zakotwiczona anotacja sylabiczna
Anotacja wyjściowa	segmentalna, fonetyczna, percepcyjna, nieciągła, zakotwiczona

Glissando jest pojęciem psychoakustycznym oznaczającym jednokierunkową, niepodzielną zmianę wysokości tonu. **Progiem glissanda** nazywamy wartość $G(T)$, poniżej której monotoniczna zmiana częstotliwości podstawowej o czasie trwania T oraz prędkości $g < G(T)$ nie

[3]Klasa dla akcentów równych występuje także u Taylora, lecz jest zaliczana do takich zjawisk fonologicznych, których z zasady nie reprezentuje się na poziomie fonetycznym(Taylor 2000, 7)



Rycina 4.7: Prozogram przykładowego sygnału mowy w języku francuskim (Mertens 2009).

jest postrzegana jako glissando ('t Hart 1976). Mertens (2004) przyjmuje:

$$G(T) = \frac{\lambda}{T^2}, \tag{4.18}$$

gdzie parametr $\lambda \in \{0.16, 0.24, 0.32\}$ jest dobierany w zależności od tempa komunikatu językowego. Za jednostkę wartości $G(T)$ przyjmuje się półton na sekundę (ST/s).

Niech będą dane segmenty s_1 oraz s_2 obejmujące monotoniczne przebiegi F_0 w pewnym sygnale x i takie, że $\overrightarrow{s_1} = \overleftarrow{s_2}$. Oznaczmy przez g_1 oraz g_2 prędkości średnie zmian F_0 w sygnale x odpowiednio w granicach segmentu s_1 oraz s_2. **Różnicowym progiem glissanda** nazywamy wartość DG [ST/s] taką, że jeśli zachodzi $|g_1 - g_2| < DG$, to przebieg F_0 w przedziale od $\overleftarrow{s_1}$ do $\overrightarrow{s_2}$ jest percypowany jako glissando. Zgodnie z d'Allesandro i Mertens (1995) $DG \in [12; 40]$, przy czym za wartość domyślną przyjmuje się 20 ST/s.

Prozogram jest to wykres cech tonalnych oparty na progach glissanda (Mertens 2004). Nazwy „Prosogram" używamy natomiast do określenia programu, który generuje prozogram. Z prozogramem wiążemy fonetyczną anotację tonalną, opartą na zakotwiczonej, nieciągłej segmentacji oraz etykietach w postaci pary liczb rzeczywistych. Dwie liczby rzeczywiste zawarte w etykiecie segmentu reprezentują wysokość tonu odpowiednio w punkcie czasowym lewej oraz prawej kotwicy segmentu. Przyjmuje się, że przebieg wysokości tonu w granicach segmentu jest liniowy. Kotwice segmentów anotacji prozogramu zlokalizowane są wyłącznie w obrębie ośrodków fonetycznych sylab.

Na rycinie 4.7 przedstawiony jest prozogram przykładowego sygnału mowy (język francuski) w wariancie podstawowym[4]. Podobnie jak w pięciolinii muzycznej, linie poziome (w prozogramie przerywane) oddalone są od siebie o 2 półtony. Strzałka znajdująca się przy lewej krawędzi wskazuje częstotliwość 150Hz. W dolnej części zamieszczone są segmentacja głoskowa oraz pomocnicza segmentacja leksykalna. Centralną część prozogramu zajmuje wykres, na którym przedstawiono segmenty anotacji etykietowane parami rzeczywistymi (segmenty o etykiecie $\varnothing$ nie są reprezentowane na prozogramie).

Prozogram został zastosowany do automatycznej anotacji korpusów mowy m.in. języka angielskiego, francuskiego oraz polskiego. Do podstawowych zalet prozogramu zaliczamy: 1) uwzględnienie zjawisk percepcyjnych, 2) zwartą i prostą w interpretacji anotację wynikową. Wśród wad prozogramu należy wskazać: 1) wysokie wymagania, co do danych wejściowych (dla uzyskania dobrej skuteczności konieczna jest zakotwiczona anotacja głoskowa lub sylabiczna), 2) brak jednoznacznych kryteriów doboru parametrów (progu glissanda, różnicowego progu glissanda) mających duży wpływ na postać anotacji wyjściowej.

[4]Istnieją rozszerzone warianty prozogramu, zawierające wykresy dodatkowych sygnałów oraz anotacji wykorzystywanych w algorytmie analizy prozogramu.

ROZDZIAŁ 5

Fonologiczna analiza tonalna

Szerzej zakrojone badania nad obiektywną fonologiczną analizą tonalną (**analizą intonacyjną**) rozpoczęto w drugiej połowie lat osiemdziesiątych dwudziestego wieku. Główną motywacją w tworzeniu układów analizy intonacji były wartości aplikacyjne zarówno bezpośrednie (*on-line*) jak i pośrednie (*off-line*). Z zastosowaniami bezpośrednimi mamy do czynienia w przypadku układów rozpoznawania oraz rozumienia mowy zaś z zastosowaniami pośrednimi w przypadku anotacji korpusów systemów syntezy mowy. Jak wielokrotnie pokazano, uwzględnienie anotacji intonacyjnej pozwala zwiększyć skuteczność układów automatycznego rozpoznawania mowy (np. Taylor i inni 1998b; Shriberg i Stolcke 2004; Hasegawa-Johnson i inni 2005). Problem integracji układów analizy intonacyjnej z układami rozpoznawania mowy pozostaje jednak otwarty (Batliner i Möbius 2005; Ananthakrishnan i Narayanan 2009). Szereg badań nad analizą intonacyjną prowadzono w ramach prac nad układami rozumienia oraz tłumaczenia (translacji) mowy (Ostendorf i Ross 1997; Strom i inni 1997; Shriberg i Stolcke 2004). Układy analizy intonacyjnej są stałym elementem środowisk produkcyjnych współczesnych systemów syntezy mowy redukując. We współczesnych korpusowych układach syntezy mowy z tekstu, przebieg intonacji syntetyzowanego zdania jest często *produktem ubocznym* procesu selekcji segmentów (Dutoit 2008, 448).

W literaturze dotyczącej fonologicznej analizy tonalnej można wyodrębnić dwa zagadnienia: 1) badanie i rozwój anotacji (systemów fonologicznych), 2) badanie i rozwój układów analizy. Łączność między wynikami prac badaczy zapewniają anotowane korpusy mowy. W sekcji 5.1 przedstawiono stan prac w zakresie fonologicznych anotacji tonalnych. W sekcji 5.2 przedstawiono stan prac w zakresie układów analizy intonacyjnej.

5.1 Fonologiczne anotacje tonalne

Proces tworzenia systemów intonacyjnych nie jest, na obecnym poziomie wiedzy, w pełni automatyzowalny (por. sekcja 13). Definicję systemu intonacyjnego uznaje się za dostatecznie precyzyjną, jeżeli zawiera: 1) formalną specyfikację zbioru suprasegmentalnych anotacji tonalnych oraz 2) korpus mowy zawierający *odpowiednią liczbę* przykładów użycia definiowanej anotacji.

Prekursorami prac nad anotacjami i systemami intonacyjnymi byli Klinghardt i Klemm (1920) oraz Palmer (1922). Podręcznik Palmera wyznaczył kierunki prowadzenia badań nad

systemami intonacyjnymi w obrębie Szkoły Brytyjskiej (zespołu badaczy związanych z londyńskim ośrodkiem naukowym). Szkoła Brytyjska zaliczana jest do strukturalizmu — nurtu metodologicznego w lingwistyce pochodzącego od Ferdinanda de Saussure (1857–1913). W strukturalizmie stosuje się indukcyjną metodę analizy danych oraz akustyczną interpretację pojęć fonologicznych. Współczesny strukturalizm oparty jest na cyfrowym przetwarzaniu sygnałów oraz statystyce matematycznej.

W roku 1980 ukazała się praca Pierrehumbert, która wywarła duży wpływ na współczesne systemy analizy mowy (Pierrehumbert 1980). Pierrehumbert stworzyła podstawy *autosegmentalno-metrycznej teorii intonacji*, którą w późniejszych latach rozwijali m.in. Gussenhoven (1994) oraz Ladd (1996). Teoria autosegmentalno-metryczna zaliczana jest do generatywizmu — nurtu metodologicznego w lingwistyce pochodzącego od Noama Chomskiego (1928-). W generatywizmie stosuje się zstępującą (dedukcyjną) analizę danych oraz psychologiczną (kognitywistyczną) interpretację pojęć fonologicznych. Współczesna lingwistyka generatywistyczna oparta jest na kognitywistyce oraz logice formalnej.

W opisach anotacji tonalnych zamieszczonych w bieżącej sekcji szczególną uwagę poświęcono następującym aspektom: 1) zastosowanej segmentacji, 2) inwentarzowi etykiet, 3) gramatyce ciągów etykiet, 4) spójności analizy intersubiektywnej (*interlabeller agreement*), 5) uniwersalności językowej oraz 6) korpusom mowy, w których daną anotację zastosowano.

W bieżącym przeglądzie przedstawiono wyłącznie te anotacje, które z powodzeniem zastosowano w obiektywnej analizie. Dla oszczędności miejsca, zrezygnowano z podawania przykładów anotacji intonacyjnych konkretnych wypowiedzi.

5.1.1 Szkoła Brytyjska

Podstawy anotacji intonacyjnej Szkoły Brytyjskiej (BT) określił Palmer na początku wieku dwudziestego (Palmer 1922). W kolejnych latach, oparte na BT monografie na temat intonacji języka brytyjskiego przedstawili m.in. Halliday (1967), Crystal (1969), O'Connor i Arnold (1973), Jassem (1983), Cruttenden (1997), oraz Wells (2006). Anotacja BT jest powszechnie stosowana w nauczaniu języka angielskiego brytyjskiego na poziomie *proficiency*.

Anotacja BT jest oparta na segmentacji warstwowej zakotwiczonej zawierającej dwie warstwy: 1) warstwę **znaczników tonalnych** (*tone mark*), oraz 2) warstwę **wzorców wysokości tonu** (*pitch pattern*). Kotwice anotacji BT przypadają wyłącznie na granic sylabowych. Warstwa wzorców wysokości tonu jest wyższa od warstwy znaczników tonalnych w sensie definicji 1.19 ze strony 9). Segment warstwy znaczników tonalnych objemuje co najwyżej jedną sylabę akcentowaną melodycznie. Zakłada się, że każda sylaba akcentowana melodycznie jest jednocześnie pierwszą sylabą pewnego segmentu warstwy znaczników tonalnych (Jassem 1999).

Etykieta w warstwie znaczników tonalnych określa kategorię przebiegu prostych zmian wysokości tonu (Cruttenden 1997, 38). W tabeli 5.1 jest podany zbiór etykiet intonacyjnych dla języka angielskiego brytyjskiego, który zaproponowali O'Connor i Arnold (1973). Zbliżone zbiory etykiet użyli w swoich pracach: Palmer (1922), Jassem (1999) oraz Wells (2006). Ikony znaczników tonalnych mają ustalone pozycje względem poziomej linii odniesienia (np. linii bazowej tekstu, przy którym umieszcza się ikony). Nawiasy kwadratowe otaczające ikony w tabeli służą wyłącznie do pokazania punktu odniesienia. W opisach wysokości tonu zastosowano następujące skróty: xL (bardzo niski, *extra-low*), L (niski, *low*), M (średni, *medium*), H

Tabela 5.1: Etykiety (ikony) tonalne BT dla języka angielskiego brytyjskiego (O'Connor i Arnold 1973).

Ikona	Akcent	Przebieg	Początek	Koniec
[↘]	+	opadający	M	L
[↘]	+	opadający	H	xH
[^]	+	rosnąco-opadający	M	xL
[↗]	+	rosnący	xL	M
[↗]	+	rosnący	M	H
[˅]	+	opadająco-rosnący	H	M
[>]	+	równy	M	M
[ˈ]	+	równy	H	H
[ˌ]	+	równy	xL	xL
[↘]	+	ściśle opadający	H	M, L
[↗]	+	ściśle rosnący	xL	LT
[°]	−	różny	HT, H	różny
[₀]	−	rosnący, równy	xL	M
[¯]	−	równy	H	H
[ ](*brak*)	−	równy	L, M	L, M

(wysoki, *high*), LT (niższy od następnego, *lower than*) oraz HT (wyższy od następnego, *higher than*). W tabeli podano wyłącznie przykładowe przebiegi wysokości tonu, dalsze informacje podają O'Connor i Arnold (1973, 289).

W warstwie wzorców wysokości tonu stosuje się cztery etykiety: `'PreHead'`, `'Head'`, `'Nuclear tone'` oraz `'Tail'`. Przyjmuje się, że akcent melodyczny występuje wyłącznie w granicach segmentów etykietowanych jako `'Head'` oraz `'NuclearTone'`. Dla każdej etykiety wzorca wysokości tonu przyporządkowany jest zbiór znaczników tonalnych, które są dopuszczalne w granicach segmentu o danej etykiecie. W związku z tym znaczniki tonalne BT grupuje się ze względu na przebieg oraz otaczający wzorzec wysokości tonu jako pokazano to w tabeli 5.2. W analizie intonacyjnej języka anlgielskiego pojęcie melodii (*nuclear tone*) rdzennej oraz rdzenia (*nucleus*) wprowadził Palmer. Rdzeniem Palmer nazywa sylabę akcentowaną najbardziej znaczącego (*the most prominent*) wyrazu w granicach frazy intonacyjnej (*tone group*) (Palmer 1922, 7). Halliday (1967) wiąże pojęcie rdzenia z *rematem*, zgodnie z opozycją temat-remat (to, co wiadome na podstawie wcześniejszych wypowiedzi z tym, co nowe). Podobne stanowisko przyjmuje Cruttenden (1997, 81). Jassem (1999) stawia hipotezę, że rdzeń występuje w pozycji leksu najmniej oczekiwanego przez mówcę (*maksimum entropii*), słuchacza, bądź obu w bieżącej frazie intonacyjnej.

Ogólną gramatykę frazy intonacyjnej BT opisuje się wyrażeniem regularnym:

```
/(PreHead)?(Head)*(NuclearTone)(Tail)?/.
```
(5.1)

Na rys. 5.1 przedstawiono gramatykę etykiet intonacyjnych otrzymaną programem FSA6 (van Noord 2009) jako determinizację zbioru wyrażeń regularnych wynikających z tabeli, którą zamieścili O'Connor i Arnold (1973, 288).

Tabela 5.2: Grupy etykiet (ikon) tonalnych BT dla języka angielskiego brytyjskiego.

Oznaczenie	Przebieg	Wzorce wys. tonu	Ikony
lph	Low	`'PreHead'`	[ ](*brak*)
hph	High	`'PreHead'`	[¯]
hh	High	`'Head'`	[|]
lh	Low	`'Head'`	[|]
fh	Falling	`'Head'`	[↘] [↘] [↘]
rh	Rising	`'Head'`	[↗] [↗] [↗]
lf	Low Fall	`'NuclearTone'`	[↘]
hf	High Fall	`'NuclearTone'`	[↘]
lr	Low Rise	`'NuclearTone'`	[↗]
hr	High Rise	`'NuclearTone'`	[↗]
fr	Fall-Rise	`'NuclearTone'`	[˅]
rf	Rise-Fall	`'NuclearTone'`	[^]
ml	Mid-Level	`'NuclearTone'`	[>]

(Jassem 1999) proponuje mniej restrykcyjną gramatykę BT, którą można opisać wyrażeniem regularnym:

$$\texttt{/(wPT)?(sPT)*(NT)/}, \tag{5.2}$$

gdzie `'wPT'` oznacza **melodię przedrdzenną słabą** (*weak Prenuclear Tune*), `'sPT'` oznacza **melodię przedrdzenną silną** (*strong Prenuclear Tune*) oraz `'NT'` oznacza **melodię rdzenną** (*Nuclear Tune*).[1]

Wśród korpusów języka angielskiego zawierających anotację BT wyróżnić można LLC (Greenbaum i Svartvik 1990) oraz MARSEC (Roach i inni 1993). Roach (1994) podaje reguły konwersji popularnej obecnie anotacji ToBI (por. sekcja 5.1.4) na anotację BT.

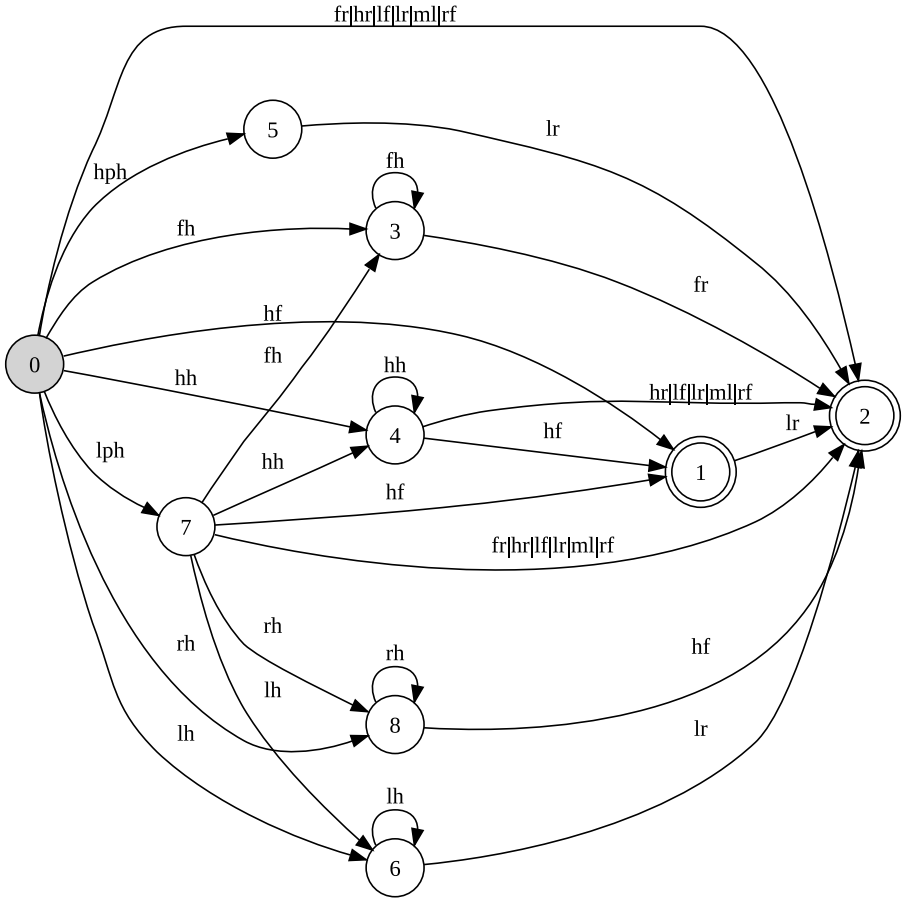
Pomimo iż anotacja BT powstała jako narzędzie analizy języka angielskiego brytyjskiego, szereg przyjętych w niej rozwiązań znalazło zastosowanie w analizie innych języków europejskich. W oparciu o anotację BT zaproponowano językowo-specyficzne anotacje intonacyjne m.in. dla języków: włoskiego (D'Imperio 2002), niemieckiego (Kohler 1991), greckiego (Papazachariou 1994) oraz polskiego (Jassem 2003a).

Do zalet anotacji BT należy zweryfikowana przydatność w dydaktyce oraz relatywnie dobrze opisana zależność od niżej-poziomowych anotacji tonalnych. Wśród wad anotacji BT można wymienić brak standaryzacji zbioru etykiet oraz ich identyfikatorów ASCII[2], utrudniający tworzenie i wymianę korpusów mowy. W analizie mowy spontanicznej niepożądaną cechą BT jest całościowe traktowanie (akceptacja lub odrzucenie) poprawności frazy intonacyjnej (Karpiński 2006, 62). W niektórych pracach kwestionuje się założenie o melodyczności akcentu brytyjskiego przyjmowane w BT, por. Kochanski i inni (2005).[3]

[1] Tematem przewodnim cytowanych prac Jassema jest połączenie anotacji intonacyjnej BT oraz modelu rytmu. Model rytmu Jassema osiągną najwyższy stopień korelacji z danymi empirycznymi w badaniach porównawczych Hirst i Buzon (2005).

[2] W pracy Roach (1994) zaproponowano zbiór etykiet ASCII dla BT, lecz brak ciała, które uczyniłoby podobną propozycję standardem.

[3] Zarówno O'Connor i Arnold (1973) jak i Jassem (1984) dopuszczają w ograniczonej liczbie kontekstów wystąpienie akcentu niemelodycznego.



Rycina 5.1: Gramatyka BT frazy intonacyjnej języka angielskiego brytyjskiego.

5.1.2 IPO

Anotacja IPO opracowana została w latach 70-tych oraz 80-tych XX wieku w Instytucie Badań nad Percepcją (IPO) w Eindhoven ('t Hart i inni 1990, 73). Istotą anotacji IPO jest etykietowanie wyłącznie percepcyjnie istotnych zmian wysokości tonu. Anotacja IPO jest anotacją językowo-specyficzną, w której inwentarz etykiet oraz gramatyka tworzone są przez eksperta na podstawie dobrze określonej, językowo-uniwersalnej heurystyki ('t Hart i inni 1990; Nooteboom 1997). W heurystyce tej stosuje się analizę intersubiektywną. W latach 90-tych dokonano obiektywizacji fonetycznego poziomu analizy tonalnej w ramach IPO (Hermes 2006).

Anotacja IPO oparta jest na segmentacji rozłącznej zakotwiczonej, w której liczba segmentów segmentacji obejmowanych przez dowolną sylabę nie przekracza trzech.

W tabeli 5.3 opisano etykiety anotacji IPO dla języka niderlandzkiego, które zaproponowali 't Hart i inni (1990, 73) oraz Hermes (2006, 53). Etykiety IPO związane z akcentem melodycznym (*accent lending*) oznaczono w tabeli 5.3 znakiem '+'.

Ogólna postać frazy intonacyjnej[4] w anotacji IPO opisywana jest wyrażeniem regularnym:

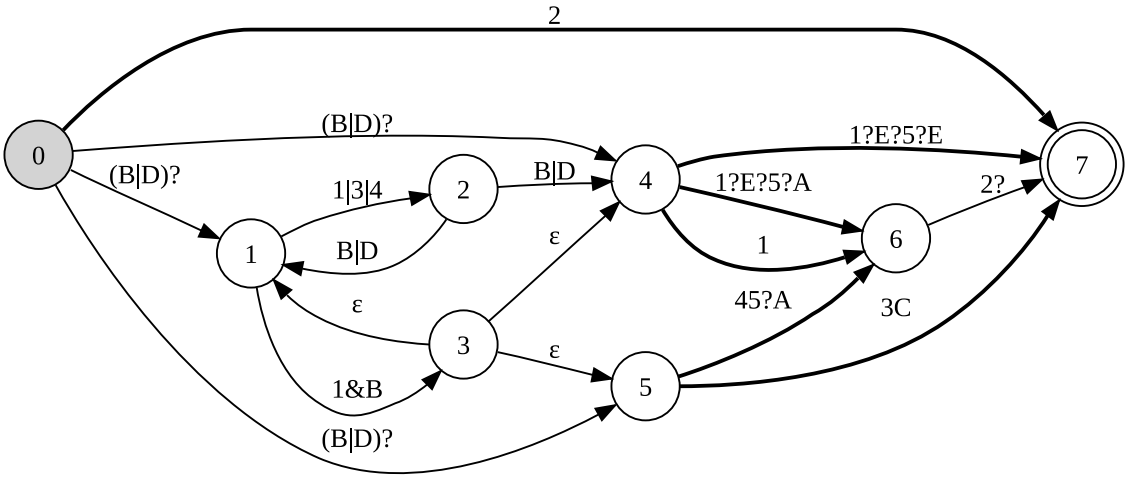
$$\texttt{/(Prefix)*(Root)(Suffix)?/}, \tag{5.3}$$

gdzie `'Prefix'`, `'Root'` oraz `'Suffix'` reprezentują zbiory etykiet. Na rys. 5.2 przedstawiono szczegółową gramatykę IPO frazy intonacyjnej dla języka niderlandzkiego ('t Hart i inni

[4]W pracach na temat anotacji IPO fraza intonacyjna nazywana jest *konturem*.

Tabela 5.3: Etykiety tonalne IPO dla języka niderlandzkiego.

Etykieta	Kierunek	Synchronizacja	Prędkość	Zakres	Akcent
1	wzrost	wczesny	szybki	pełny	+
2	wzrost	bardzo późny	szybki	pełny	-
3	wzrost	późny	szybki	pełny	+
4	wzrost	dowolna	wolny	pełny	-
5	wzrost	wczesny	szybki	częściowy	+
A	spadek	późny	szybki	pełny	+
B	spadek	wczesny	szybki	pełny	-
C	spadek	bardzo późny	szybki	pełny	-
D	spadek	dowolna	wolny	pełny	-
E	spadek	wczesny	szybki	częściowy	+



Rycina 5.2: Gramatyka IPO frazy intonacyjnej języka niderlandzkiego.

1990, 81). Dla zwiększenia czytelności, przejścia między stanami opisano za pomocą wyrażeń regularnych. Symbol '&' wstawiony między etykiety IPO oznacza, że etykiety dotyczą tej samej sylaby. Na rys. 5.2 pogrubiono krawędzie odpowiadające części $Root.

Heurystykę IPO zastosowano do korpusów mowy dla języków niderlandzkiego, brytyjskiego, niemieckiego oraz rosyjskiego, otrzymując rodzinę językowo-specyficznych anotacji (Hermes 2006).

Korpusy mowy z anotacją IPO stworzyli m.in. 't Hart i Collier (1975) dla języka niderlandzkiego oraz Willems i inni (1988) dla języka angielskiego brytyjskiego.

Indukcyjny algorytm analizy danych, bezpośrednie powiązanie etykiet anotacji z przebiegiem częstotliwości oraz rozbudowana gramatyka etykiet upodabniają anotację IPO do anotacji Szkoły Brytyjskiej. Istotnym rozszerzeniem w stosunku do Szkoły Brytyjskiej była częściowa formalizacja heurystyki tworzenia etykiet oraz gramatyki intonacyjnej w oparciu o analizę intersubiektywną korpusów mowy. ('t Hart i inni 1990, 66) stwierdza, że anotacje IPO są *węższe* od anotacji Szkoły Brytyjskiej, tj. w mniejszym stopniu spełniają fonologiczne kryteria dystynktywności.

5.1.3 SAMPROSA

Anotacja SAMPROSA (*SAM Prosodic Transcription*) została opracowana przez Dafydda Gibbona w ramach prac nad standardem SAM (*Speech Assessment Methods*) (Wells i inni 1992). SAMPROSA jest uzupełnieniem fonologicznej anotacji segmentalnej X-SAMPA (językowo uniwersalna wersja *SAM Phonetic Alphabet*), wraz z którą standaryzuje zapis komunikatu językowego w postaci ciągu znaków ASCII. W dalszej części opisu pomijamy te elementy anotacji SAMPROSA, które nie są związane z cechami tonalnymi.

W odróżnieniu do innych opisywanych w niniejszej pracy anotacji intonacyjnych, SAMPROSA nie jest związana z ustalonym systemem intonacyjnym, algorytmem analizy tonalnej bądź też językiem. Rola SAMPROSA ogranicza się do określenia metod synchronizacji anotacji suprasegmentalnych z anotacjami segmentalnymi oraz określenia ciągów znaków ASCII reprezentujących etykiety stosowane w szeregu proponowanych wcześniej fonologicznych anotacji tonalnych (Grice i inni 2000, 53). Anotacja tonalna SAMPROSA oparta jest na segmentacji prostej zaktowiczonej w sygnale mowy i/lub powiązanej z segmentacją sylabiczną/fonematyczną.

Zbiór etykiet SAMPROSA (por. tabela 5.4) jest nadzbiorem zbiorów etykiet intonacyjnych szeregu popularnych systemów intonacyjnych. W tabeli 5.4 podano etykiety opisujące zarówno cele (*tonal targets*) jak i zmiany (*tonal transitions*), zarówno poziomy (*tonal levels*) jak i konfiguracje (*tonal configurations*).

SAMPROSA nie określa gramatyki etykiet tonalnych oraz jest językowo uniwersalna.

Anotacja SAMPROSA została zastosowana m.in. w module polskim korpusu mowy SpeeCon (Marasek i Gubrynowicz 2005).

Głównymi zaletami anotacji SAMPROSA są standaryzacja kodów etykiet (ASCII) oraz niezależność od teorii analizy intonacji. Podstawowymi wadami omawianej anotacji jest brak dostatecznej ścisłości w określeniu zasad stosowania etykiet (nie opisano dokładnie korelatów akustycznych, brak gramatyki) oraz brak dostatecznej liczby kategorii dla objęcia dystynktywnych przebiegów intonacyjnych spotykanych w językach naturalnych. SAMPROSA nie zyskała znaczącej popularności, m.in. w związku z równoległym zaproponowaniem systemu ToBI, który standaryzował identyfikatory ASCII dla etykiet oraz (przynajmniej w założeniach) miał być niezależny od języka i modelu analizy intonacji. wady – możliwość oznaczenia tego samego przebiegu na wiele sposobów (w zależności od systemu intonacyjnego)

5.1.4 ToBI

Anotację ToBI (*Tones and Break Indices*) zaproponowali Silverman i inni (1992) jako standard anotacji intonacyjnej (prozodycznej) w korpusach mowy angielskiej amerykańskiej. ToBI jest oparta na wcześniejszych pracach z zakresu lingwistyki generatywnej, w szczególności na modelu Pierrehumbert (1980) nazywanym obecnie autosegmentalno-metryczną teorią intonacji (Ladd 1996). W stosunku do modelu Pierrehumbert, w anotacji ToBI zmodyfikowano inwentarz etykiet oraz wprowadzono anotację **indeksów granic międzywyrazowych** (*Break Indices*), które określają *siłę* granicy międzywyrazowej. Zgodnie z Beckman i inni (2005) celem anotacji ToBI nie jest odwracalność (percepcyjna) lecz zapis informacji istotnych dla ponadfonologicznych poziomów analizy intonacyjnej.

Tabela 5.4: Etykiety tonalne SAMPROSA (językowo uniwersalne).

Ciąg ASCII	Rodzaj	Ton
H	cel, poziom	wysoki
L	cel, poziom	niski
T	cel, poziom	najwyższy
B	cel, poziom	najniższy
M	cel, poziom	środkowy
+	cel, konfiguracja	wyższy od segmentu poprzedzającego
++	cel, konfiguracja	znacznie wyższy od segmentu poprzedzającego
+-	cel, konfiguracja	wyższy od segmentów sąsiednich
-	cel, konfiguracja	niższy od segmentu poprzedzającego
–	cel, konfiguracja	znacznie niższy od segmentu poprzedzającego
-+	cel, konfiguracja	niższy od segmentów sąsiednich
^	cel, konfiguracja	wysoki, wyższego stopnia (*upstep*)
^^	cel, konfiguracja	wysoki, wyższego stopnia, szeroki (*wide upstep*)
!	cel, konfiguracja	niski, niższego stopnia (*downstep*)
!!	cel, konfiguracja	niski, niższego stopnia, szeroki (*wide downstep*)
= lub > lub S	cel, konfiguracja	równy
-	przejście, konfiguracja	rdzenny równy
’ lub / lub R	przejście, konfiguracja	rdzenny rosnący
‘ lub \ lub F	przejście, konfiguracja	rdzenny opadający
‘’ lub \/ lub FR	przejście, konfiguracja	rdzenny opadająco-rosnący
’‘ lub /\ lub RF	przejście, konfiguracja	rdzenny rosnąco-opadający

Anotacja ToBI składa się z czterech warstw: 1) warstwy sylabicznej, 2) warstwy leksykalnej, 3) warstwy *fraz pośrednich* oraz 4) warstwy *fraz intonacyjnych*. Warstwa $i+1$ w powyższym wypunktowaniu jest wyższa od warstwy i w sensie określonym w def. 1.19 na str. 9. W warstwie 1 określa się rodzaj akcentu melodycznego (*pitch accent*) dla sylaby lub jego brak. W warstwie 2 określa się indeks granicy międzywyrazowej. W warstwach 3 i 4 określa się odpowiedno rodzaj akcentu frazowego (*phrase accent*) oraz tonu granicznego (*boundary tone*).

Na zbiór etykiet ToBI składają się etykiety tonalne (akcentów melodycznych, akcentów frazowych oraz tonów granicznych) oraz etykiety indeksów granic międzywyrazowych. Etykiety tonalne anotacji ToBI oparte są na pojęciach tonu niskiego ’L’ oraz tonu wysokiego ’H’. We wczesnych wersjach anotacji ToBI interpretacja wysokości tonów odnosiła się wyłącznie do średniej wysokości tonu w danej frazie. W późniejszych wersjach ToBI wprowadzono akcenty melodyczne wysokie niższego stopnia (*downstepped*), interpretowane jako akcenty wysokie, których ton jest niższy od poprzedzających je akcentów wysokich (Brugos i inni 2006). W tabeli 5.5 opisano etykiety tonalne ToBI dla języka angielskiego amerykańskiego (Beckman i inni 2005), (Brugos i inni 2006). Etykiety indeksów granic międzywyrazowych oznaczane są liczbami ze zbioru uporządkowanego $\{0, 1, 2, 3, 4\}$, gdzie wartość 0 oznacza brak granicy międzywyrazowej (silną koartykulację), 1 przeciętną granicę wyrazową, 2 granicę niedopaso-

Tabela 5.5: Etykiety tonalne ToBI dla języka angielskiego amerykańskiego.

Etykieta	Warstwa	Opis
H*	sylabiczna	akcent melodyczny wysoki
L*	sylabiczna	akcent melodyczny niski
L+H*	sylabiczna	akcent melodyczny wysoki poprzedzony tonem niskim
L*+H	sylabiczna	ton wysoki poprzedzony akcentem melodycznym niskim
!H*	sylabiczna	akcent melodyczny wysoki niższego stopnia
L+!H*	sylabiczna	akcent melodyczny wysoki niższego stopnia poprzedzony tonem niskim
L*+!H	sylabiczna	ton wysoki niższego stopnia poprzedzony akcentem melodycznym niskim
H+!H*	sylabiczna	akcent melodyczny wysoki niższego stopnia poprzedzony tonem wysokim
L-	fraz pośrednich	akcent frazowy niski
H-	fraz pośrednich	akcent frazowy wysoki
L%	fraz intonacyjnych	ton graniczny niski
H%	fraz intonacyjnych	ton graniczny wysoki

waną (*mismatched*) a wartości 3 i 4 są stosowane obligatoryjnie dla wyrazów bezpośrednio przed granicami fraz pośrednich oraz odpowiednio fraz intonacyjnych.[5]

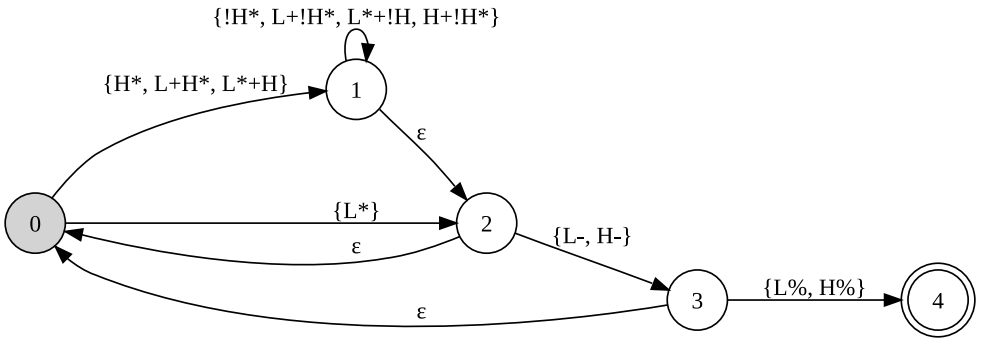
Pierrehumbert (1980) nie nakładaograniczeń na dopuszczalne ciągi etykiet w obrębie warstw. Wraz z wprowadzeniem akcentów niższego stopnia w anotacji ToBI ograniczono dopuszczalne ciągi akcentów melodycznych. Na rys. 5.3 przedstawiono gramatykę etykiet ToBI w zapisie linearnym[6] (Brugos i inni 2006). Należy zauważyć, że ToBI w dalszym ciągu nie nakłada ograniczeń na ciągi etykiet w warstwach akcentów frazowych oraz tonów granicznych.

Obecnie zaleca się, by nazwy ToBI używać w odniesieniu do rodziny językowo-specyficznych anotacji, natomiast anotację dla języka angielskiego amerykańskiego określać jako MAE_ToBI (*Mainstream American English* ToBI) (Beckman i inni 2005). Do najbardziej kompletnych opisów anotacji intonacyjnych z rodziny ToBI należą GToBI (*German* ToBI) (Grice i inni 1996) oraz JToBI (*Japanese* ToBI) (Venditti 1997). Prace nad anotacjami z rodziny ToBI prowadzi się dla języków serbskiego, słowackiego, greckiego, katalońskiego, portugalskiego, koreańskiego i chińskiego (Jun 2005; of Linguistics 2007). Dotychczas nie zaproponowano całościowego opisu języka polskiego, który byłby zgodny z założeniami ToBI.

Pitrelli i inni (1994), Syrdal i McGory (2000) oraz Ostendorf i inni (2001) niezależnie badają spójność intersubiektywnej analizy MAE_ToBI. W tych pracach uzyskano 85%-92%

[5]Powiązanie granic wyrazowych z końcami fraz pośrednich oraz intonacyjnych jest trudne do uzasadnienia na podstawie danych empirycznych i należy do najbardziej kontrowersyjnych aspektów ToBI (Wightman 2002).

[6]Pomimo warstwowości anotacji ToBI w wielu sytuacjach wygodnie jest umieszczać etykiety w jednej sekwencji, tj. w zapisie linearnym.



Rycina 5.3: Gramatyka ToBI frazy intonacyjnej języka angielskiego amerykańskiego (w zapisie sekwencyjnym).

zgodność anotatorów w odniesieniu do występowania lub braku akcentu frazowego oraz tonu granicznego. Podobnie wysoką zgodność anotatorów (81%-91%) odnotowano w przypadku występowania lub braku akcentu melodycznego. Jeśli jednak zliczanie dotyczy równości kategorii (etykiet) w anotacji MAE_ToBI, to odnotowywana zgodność nie przekracza 50% (Syrdal i McGory 2000; Ostendorf i inni 2001). W pracy Ostendorf i inni (2001) po analizie intersubiektywnej, dla otrzymania zadowalających wyników wykonano subiektywną korektę anotacji przez wybranego eksperta MAE_ToBI. W pracy Yoon i inni (2004) badano natomiast spójność anotacji ToBI dla mowy spontanicznej. Dla ograniczonej liczby etykiet (wyłącznie H* oraz L* w przypadku akcentów melodycznych) otrzymano zgodność przekraczającą 80%.

Najważniejszymi korpusami mowy angielskiej amerykańskiej, w których zamieszczono (przynajmniej w części) anotację ToBI są: Boston Direction Corpus (Nakatani i inni 1995), Boston Radio News Corpus (Ostendorf i inni 1996) oraz Switchboard/Callhome (Ostendorf i inni 2001).

Silverman i inni (1992) zamieszczają szereg założeń, które miały wyznaczać kierunki rozwoju ToBI, m.in. 1) neutralność względem teorii lingwistycznych, 2) uniwersalność językową (wystąpiło porównanie z IPA), 3) spójność analizy intersubiektywnej oraz 4) zapis przyjazny dla komputera. Neutralność względem teorii lingwistycznych na samym początku została podważona przyjęciem teorii Pierrehumbert.[7] Trudności w językowo-uniwersalnym traktowaniu ToBI pojawiają się jednak już w obrębie dialektów języka angielskiego (Nolan i Grabe 1997).[8] Językowo-specyficzne anotacje należące do rodziny ToBI odbiegają od siebie dość znacznie, por. GToBI (Grice i inni 1996) oraz JToBI (Venditti 1997). Zgodność analizy intersubiektywnej liczona dla pełnego zestawu etykiet jest bardzo niska (Wightman 2002) (Batliner i Möbius 2005). Przyjazność anotacji dla komputera (nieikoniczne etykiety ASCII) wpłynęła negatywnie na czytelność dla człowieka, w tym możliwości zastosowania w dydaktyce języków obcych (Wells 2006, 262). Pomimo wymienionych wątpliwości ToBI jest obecnie najpopularniejszą fonologiczną anotacją intonacyjną stosowaną w technologii mowy. Jedną z silnych stron ToBI jest dostępność anotowanych korpusów mowy (dot. głównie MAE_ToBI).

[7] Wraz z rozwojem, w ToBI zaczęto asymilować koncepcje wypracowane w niegeneratywistycznych modelach intonacji, m.in. Szkole Brytyjskiej oraz IPO (opis etykiet ToBI w kategoriach sygnałowych, operowanie pojęciem akcentu rdzennego, wprowadzenie akcentów o wysokości względnej), por. (Brugos i inni 2006).

[8] W pracy Grabe i inni (2000) zaproponowano szereg rozszerzeń anotacji ToBI, które umożliwiają analizę porównawczą anotacji intonacyjnych w obrębie dialektów języka angielskiego występujących na Wyspach Brytyjskich.

Tabela 5.6: Etykiety tonalne PROLAB dla języka niemieckiego.

Etykieta	Warstwa	Opis
<&1>	leksykalna	akcent melodyczny osłabiony
<&2>	leksykalna	akcent melodyczny domyślny
<&3>	leksykalna	akcenty melodyczny wzmocniony
<&.>	melodyczna	melodia opadająca
<&,>	melodyczna	melodia rosnąca zakończona niżej
<&?>	melodyczna	melodia rosnąca zakończona wyżej
<&.,>	melodyczna	melodia opadająco-rosnąca zakończona niżej
<&.?>	melodyczna	melodia opadająco-rosnąca zakończona wyżej
<&PGn>	frazowa	fraza zakończona granicą o *sile oddzielania* n

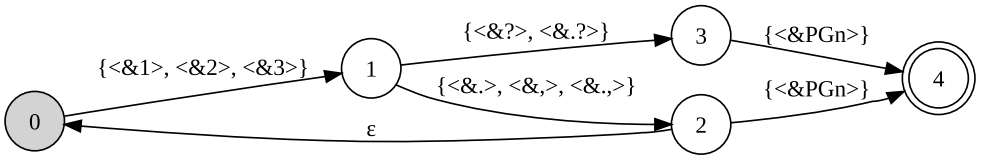
5.1.5 PROLAB

Anotacja PROLAB powstała w oparciu o lingwistyczny model prozodii języka niemieckiego KIM *Kiel Intonation Model* (Kohler 1997). Model KIM (*Kiel Intonation Model*) powstał na początku lat dziewięćdziesiątych w ramach badań nad syntezą mowy (Kohler 1991). Model KIM wzorowany był na podejściu Szkoły Brytyjskiej (Kohler 2006, 125). Późniejsze rozszerzenia modelu KIM i anotacji PROLAB ukierunkowane były na opis mowy spontanicznej (Kohler 2006).

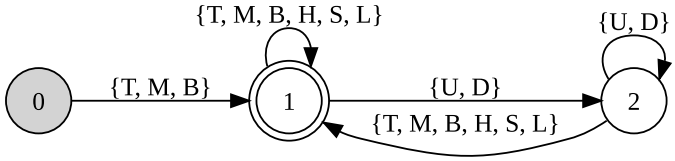
Anotacja tonalna PROLAB składa się z trzech warstw: 1) warstwy leksykalnej, 2) warstwy melodycznej oraz 3) warstwy frazowej. Warstwa $i+1$ w powyższym wypunktowaniu jest wyższa od warstwy i w sensie określonym w def. 1.19 na str. 9). W warstwie 1 określa się lokalizację (z dokładnością do leksu) oraz *intensywność* akcentu melodycznego. W warstwie 2 określa się lokalizację melodii oraz kategorię przebiegu wysokości tonu. W warstwie 3 określa się lokalizację frazy (intonacyjnej) oraz *siłę* granicy frazowej. Każda melodia zawiera dokładnie jeden leks akcentowany. Jak można zauważyć, warstwy anotacji PROLAB odpowiadają pojęciom toniczności (warstwa leksykalna), tonu (warstwa melodyczna) oraz tonalności (warstwa frazowa), które wprowadził (Halliday 1967).

Etykiety PROLAB mają ogólną postać opisywaną wyrażeniem regularnym /<&.*>/. Wyróżniającą cechą modelu KIM jest kategorialny opis synchronizacji przebiegu częstotliwości podstawowej z warstwą segmentalną (w szczególności początkiem sylaby/środkiem samogłoski w sylabie), (Kohler 1997, 195), (Kohler 2003). W anotacji PROLAB, kategoria synchronizacji reprezentowana jest przez pojedynczy symbol ze zbioru mbox{'[', '(', '^', ']', ')'} wstawiany w etykiecie warstwy leksykalnej bezpośrednio po cyfrze. Etykiety ')', '^' oraz '(' reprezentują odpowiednio wczesne, środkowe oraz późne maksimum lokalne przebiegu F_0. Etykiety ']' oraz '[' reprezentują odpowiednio wczesne oraz niewczesne minimum lokalne przebiegu F_0. W tabeli 5.6 opisano etykiety PROLAB (Kohler 2006). Dla przejrzystości, w tabeli 5.6 nie uwzględniono kategorii synchronizacji.

Na rys. 5.4 przedstawiono gramatykę etykiet z tabeli 5.6 w zapisie linearnym. W tekście ortograficznym etykiety PROLAB umieszczane są przed lub po wyrazach.



Rycina 5.4: Gramatyka PROLAB frazy intonacyjnej języka niemieckiego (w zapisie sekwencyjnym).



Rycina 5.5: Gramatyka INTSINT ciągu etykiet intonacyjnych dowolnego języka.

Anotacja PROLAB została zastosowana w analizie korpusu niemieckiej mowy spontanicznej KCSS (IPDS 1995) (oraz późniejszych rozszerzeniach KCSS z lat 1996 oraz 1997).

5.1.6 INTSINT

Hirst i Cristo (1998) zaproponowali fonologiczną anotację tonalną o nazwie INTSINT (*INternational Transcription System for INTonation*). INTSINT oparto na wcześniejszych (przełom lat 80-tych i 90-tych) pracach Hirsta mających na celu stworzenie intonacyjnego odpowiednika alfabetu IPA. Celem twórców INTSINT było stworzenie językowo uniwersalnej anotacji intonacyjnej, m.in. na potrzeby badań porównawczych.

Anotacja INTSINT ma postać $A = (S, a)$, gdzie S jest segmentacją antoacji MOMEL (por. sekcja 4.2) oraz a jest funkcją etykietującą o przeciwdziedzinie będącej zbiorem znaków: {T,M,B,H,S,L,U,D}. Litery zawarte w powyższym zbiorze pochodzą od wyrazów: „*Top*", „*Mid*", „*Bottom*", „*Higher*", „*Same*", „*Lower*", „*Upstepped*" oraz „*Downstepped*" opisujących wysokość tonu. Wśród etykiet INTSINT wyróżnia się dwa rozłączne podzbiory: 1) etykiety bezwzględne {T,M,B} oraz 2) etykiety względne {H,S,L,U,D}. Wysokość tonu dla etykiet bezwzględnych określana jest w odniesieniu do zakresu wysokości tonu mówcy (zakresu globalnego lub w bieżącym komunikacie). W przypadku etykiet względnych wysokość tonu określana jest w odniesieniu do etykiet segmentów sąsiednich w anotacji. W tabeli 5.7 opisane są etykiety literowe oraz odpowiadające im ikony w anotacji INTSINT (Hirst i inni 2000).

Ograniczenia dopuszczalnych ciągów etykiet INTSINT wynikają wyłącznie z definicji etykiet względnych (konieczność istnienia segmentu poprzedniego lub następnego). Na rys. 5.5 przedstawiono gramatykę etykiet INTSINT.

INTSINT został zastosowany do analizy intonacyjnej kilkunastu języków naturalnych, w tym: angielskiego, francuskiego, włoskiego, katalońskiego, portugalskiego, hiszpańskiego, rosyjskiego i arabskiego (Hirst i Cristo 1998), (Hirst 2007).

Anotacja INTSINT otrzymywana jest wyniku obiektywnej analizy tonalnej. Kolejne warianty algorytmu analizy stworzyli m.in. Hirst (2000) oraz Hirst (2007).

Tabela 5.7: Etykiety tonalne INTSINT (językowo uniwersalne).

Etykieta	Ikona	Rodzaj	Wysokość tonu
T	$\Uparrow$	bezwzględna	najwyższa dla komunikatu językowego lub mówcy
M	$\Rightarrow$	bezwzględna	średnia dla komunikatu językowego lub mówcy
B	$\Downarrow$	bezwzględna	najniższa dla komunikatu językowego lub mówcy
H	$\uparrow$	względna	większa od wysokości tonu segmentów sąsiednich
S	$\rightarrow$	względna	równa wysokości tonu segmentu poprzedniego
L	$\downarrow$	względna	mniejsza od wysokości tonu segmentów sąsiednich
U	$<$	względna	większa od wysokości tonu segmentu poprzedniego oraz mniejsza od wysokości tonu segmentu następnego
D	$>$	względna	mniejsza od wysokości tonu segmentu poprzedniego oraz większa od wysokości tonu segmentu następnego

Najważniejszymi korpusami mowy, dla których wykonano analizę INTSINT są EUROM1 (Chan i inni 1995) (język angielski brytyjski oraz francuski) oraz Aix-MARSEC (Auran i inni 2004) (język angielski brytyjski).

W przypadku analizy określonego języka wadą anotacji INTSINT może być niskopoziomowość (ignorowanie fonologicznych kryteriów dystynktywności, por. sekcja 1.3). Hermes (2006) do wad anotacji INTSINT zaliczona brak środków do oznaczania synchronizacji przebiegu F_0 z anotacją segmentalną (środki te ma np. anotacja PROLAB opisana w sekcji 5.1.5).

5.1.7 Anotacje statystyczne

Anotacje (systemy) intonacyjne opisane we wcześniejszej części pracy powstały na gruncie subiektywnych oraz intersubiektywnych kryteriów dystynktywności. Dotychczas opublikowano relatywnie niewiele prac na temat obiektywizacji algorytmu tworzenia systemów intonacyjnych. Systemy (anotacje) intonacyjne opisane w bieżącej sekcji oparto na kryterium statystycznym oraz technikach uczenia się bez nadzoru.[9] Koncepcja tworzenia systemów intonacyjnych wyłącznie w oparciu o obiektywne kryteria dystynktywności jest bardzo atrakcyjna, gdyż rozwiązuje problemy spójności (otrzymuje się obiektywny algorytm analizy) oraz uniwersalności językowej (kryterium stosuje się na jedno lub wielojęzycznym korpusie mowy). Zgodnie z opisem w sekcji 1.3 na stronie 13 wyróżnia się dwa rodzaje obiektywnych kryteriów dystynktywności: statystyczne oraz dystrybucyjne. W bieżącej sekcji przedstawione są prace nad użyciem statystycznego kryterium dystynktywności w analizie fonologicznej cech tonalnych sygnału mowy przeprowadzone w ostatnich kilkunastu latach.

[9] Zgodnie z terminologią wprowadzoną w sekcji sekcja 1.3, zastosowanie wyłącznie statystycznego kryterium dystynktywności prowadzi do otrzymania systemu fonetycznego.

Veronis i Campione (1998) proponują językowo-uniwersalny tonalny system fonologiczny RSCI (*Reversible Symbolic Coding of Intonation*), który oparty jest na trzech założeniach: 1) normalności rozkładu ekstremów przebiegu wysokości tonu, 2) niezależności między zmianą wysokości tonu oraz czasem trwania tej zmiany oraz 3) niezależnością wysokości tonu od lokalizacji względem wypowiedzi.

Punktem wyjścia w systemie RSCI jest fonetyczna anotacja tonalna MOMEL opisana w sekcji 4.2. System fonologiczny RSCI określony jest przez ciąg funkcji rzeczywistych $(f^V[0], \ldots, f^V[k-1]$ gdzie parametr V reprezentuje cechy tonalne pozajęzykowe ustalonego głosu. Celem funkcji należących do f^V jest szacowanie wysokości tonu segmentu i na podstawie wysokości tonu segmentu $i-1$.

Niech będzie dana anotacja MOMEL $A = (S, a)$. Przyjmijmy, że $h[i] = \overline{S[i]}[1]$, tj. $h[i]$ oznacza wartość docelową F_0 zwartą w etykiecie i-tego segmentu anotacji A. Istotą analizy RSCI jest wyznaczenie ciągu indeksów $C[0], \ldots, C[|S_M| - 1]$ minimalizującego wartość:

$$\sum_{i=0}^{|S|-1} (h[i+1] - f^V[C[i]])^2. \tag{5.4}$$

Anotacja wyjściowa przyjmuje postać (S, a_{RSCI}), gdzie

$$a_{RSCI}(\#s[i]) = C[i]. \tag{5.5}$$

Jak wynika z powyższego, w anotacji RSCI nie występują ograniczenia dopuszczalnych ciągów etykiet (brak gramatyki).

Najprostszym rozpatrywanym modelem jest $M_2 = (f_L^{(\mu,\sigma)}, f_H^{(\mu,\sigma)})$:

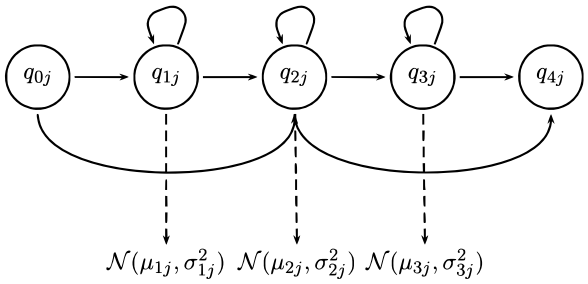
$$f_L^{(\mu,\sigma)}(h) = -\frac{1}{\sqrt{2\pi}(1 - \Phi(h))} e^{-x^2/2} \tag{5.6}$$

oraz

$$f_H^{(\mu,\sigma)}(h) = \frac{1}{\sqrt{2\pi}(1 - \Phi(h))} e^{-x^2/2}, \tag{5.7}$$

gdzie funkcja Φ jest dystrybuantą rozkładu normalnego z parametrami (μ, σ) (Veronis i Campione 1998). Elementy M_2 reprezentują wartość oczekiwaną F_0 w przypadku odpowiednio spadku oraz wzrostu, przy założeniu normalności rozkładu. Veronis i Campione (1998) testują także modele M_3, M_4, ..., M_9 obejmujące większą liczbę kategorii spadku, wzrostu oraz przebiegu stałego (w przypadku modeli o indeksach nieparzystych) stosując m.in. dystans MSE otrzymany w resyntezie. W wyniku testów przeprowadzonych na korpusie EUROM Chan i inni (1995) zaproponowano przyjęcie modelu M_7 za najlepszy kompromis między liczbą kategorii a jakością odwzorowania. (W testach wzięto pod uwagę języki angielski, francuski, niemiecki, włoski i hiszpański.) Funkcjom modelu M_7 przypisano etykiety L_3, L_2, L_1 (spadki od największego do najmniejszego), S (przebieg stały), H_1, H_2, H_3 (wzrosty od najmniejszego do największego).

W pracy (Möhler i Conkie 1998) zaproponowano algorytm VQPaIntE (Vector Quantization Parametric Intonation Event) oparty na kategoryzacji etykiet fonetycznej anotacji tonalnej PaIntE za pomocą algorytmu VQ (*Vector Quantization*). Anotacja PaIntE oparta jest na segmentacji sylabicznej oraz na etykietach w postaci sześcio-elementowych wektorów rzeczywistych, które są parametrami funkcji aproksymującej przebieg F_0 w granicach sylaby. Funkcja aproksymująca reprezentowana przez etykietę PaIntE jest funkcją ciągłą powstałą w wyniku sklejenia dwóch sigmoid.



Rycina 5.6: Topologia HMM w algorytmie UHCF0C (Lolive i inni 2007).

Istotą algorytmu proponowanego w pracy (Möhler i Conkie 1998) jest znalezienie zakładanej liczby skupień w zbiorze etykiet PaIntE metodą k-średnich. Etykiety PaIntE badanego korpusu mowy podlegają standaryzacji oraz podziałowi na zbiór uczący (80Zbiór uczący poddawany jest kwantyzacji za pomocą algorytmu Lloyda dla $k = 4, 6, 8, 16, 32$.

Algorytm VQPaIntE testowano na korpusie mowy złożonym z nagrań 145 minut mowy angielskiej (dialekt amerykański) pojedynczego mówcy. Korpus obejmował dwie sekcje: mowy czytanej neutralnie (104 minuty) oraz mowy ekspresywnej (41 minut). Po utrzorzeniu zbioru skupeń zgodnie z algorytmem VQPaIntE, zmierzono na zbiorze testowym odległość RMS między wejściowymi konturami F_0 a konturami zrekonstruowanymi z reprezentacji fonologicznej VQPaIntE. Najniższe wyniki uzyskano dla $k = 4$ oraz mowy ekspresywnej: RMS=36.8 Hz. Najwyższe wyniki uzyskano dla $k = 32$ oraz mowy czytanej neutralnie: RMS=25.1 Hz. Wartości otrzymane w obu przypadkach przekraczają znacząco próg JND.

Lolive i inni (2007) proponują algorytm UHCF0C (*Unsupervised HMM Classification of F_0 Curves*) oparty na statystycznym kryterium dystynktywności.

Danymi wejściowymi UHCF0C jest zbiór $\mathcal{V}$ złożony z par postaci (A_F, A_S), gdzie A_F jest sygnałową anotacją tonalną oraz A_S jest anotacją prostą reprezentującą struktury sylab fonologicznych. Zakłada się, że sygnałowe anotacje tonalne zawarte w danych wejściowych algorytmu A.11 mają logarytmiczną skalę F_0, poddane zostały interpolacji w granicach ramek bezdźwięcznych oraz wygładzaniu za pomocą regresji liniowej. Zbiór $\mathcal{V}$ przekazywany jest w postaci dwóch zbiorów: zbioru uczącego $\mathcal{V}^T$ oraz zbioru walidującego $\mathcal{V}^V$. Przyjmuje się, że $\mathcal{V}^T \cup \mathcal{V}^V = \mathcal{V}$ oraz $\mathcal{V}^T \cap \mathcal{V}^V = \emptyset$. Algorytm UHCF0C jest oparty na modelowaniu przebiegu F_0 (anotacja A_F) w obrębie sylaby za pomocą niejawnego modelu Markowa (HMM). Na rycinie 5.6 przedstawiona jest topologia HMM zastosowana w UHCF0C. Stany emitujące modeli HMM stosowanych w UHCF0C są związane z segmentami strukutury sylaby, odpowiednio: nagłosem, ośrodkiem fonologicznym oraz codą (anotacja A_S). Rozkłady emisji stanów modelu HMM M_j reprezentowane są w postaci jednowymiarowych rozkładów normalnych $N(\mu_i j, \sigma_i^2 j)$, gdzie $i \in \{1, 2, 3\}$ jest numerem stanu emitującego.

Algorytm A.11 zastosowano na zbiorze 10000 sylab (8000 na zbiór uczący, 3000 na zbiór walidujący) wybranych losowo z korpusu 7000 zdań czytanych przez jednego mówcę. Wyniki średniego dystansu RMS na zbiorze walidującym pokazują, że osiągnięcie błędu średniego równego w przybliżeniu progowi JND (autorzy przyjmują JND=4Hz) jest możliwe przy zastosowaniu 32 modeli HMM ($|\mathcal{M}| = 32$). Lolive i inni (2007) nie biorą pod uwagę kolejności modelowanych sylab (brak porządku, brak gramatyki).

Algorytmy przedstawione powyżej umożliwiają (częściową) automatyzację konstruowania fonetycznych i fonologicznych systemów tonalnych. Należy zastrzec, że algorytmów VQPaIntE oraz UHCF0C nie można bezpośrednio zaliczyć do metod analizy suprasegmentalnej (etykieta anotacji przypisywana jest wyłącznie na podstawie własności sygnału w granicach segmentu). W obu przypadkach moża jednak zaproponować nieznaczne rozszerzenia polegające na objęciu w procesie wyróżniania skupień sąsiedztwa segmentu sylabicznego, którego etykieta jest wyznaczana. VQPaIntE oraz UHCF0C testowane były na korpusach pojedynczych mówców mowy czytanej, co zmniejszyło wariancję sygnału mowy podlegającego badaniu. W przypadku algorytmów RSCI oraz UHC anotacje wejściowe analizowanych korpusów otrzymane były pół-automatycznie (korekta anotacji przez eksperta).

5.1.8 Anotacje intonacyjne języka polskiego

Steffen-Batóg 1996

Steffen-Batogowa (1996) opracowała pierwszą anotację intonacyjną dla języka polskiego (praca wykonana została ponad dwie dekady przed opublikowaniem w formie książkowej). Anotacja Batogowej opracowana została w wyniku analizy dedykowanego korpusu mowy. Autorka wykonała nagrania 5 kobiet i 4 mężczyzn oraz zebrała nagrania radiowe. W korpusie znalazły się wypowiedzi improwizowane (nagrane za wiedzą mówcy lecz bez narzucenia treści), wypowiedzi publiczne, deklamacje, słuchowiska oraz teksty czytane (Steffen-Batogowa 1996, 10).

Anotacja intonacyjna zaproponowana przez Batogową składa się z trzech warstw: 1) warstwy segmentów intonacyjnych oraz 2) warstwy zestrojów akcentowych oraz 3) warstwy fraz intonacyjnych. Warstwa 1 jest niższa od warstwy 2, warstwa 2 jest niższa od warstwy 3 w sensie definicji 1.19 ze strony 9). Segmentacja warstwy segmentów intonacyjnych jest tożsama z segmentacją sylabiczną (sylabizacja fonetyczna) (Steffen-Batogowa 1996, 14). Etykiety warstwy segmentów intonacyjnych należą do zbioru skończonego $K \times L \times A$, gdzie K jest zbiorem kategorii przebiegu wysokości tonu, L jest zbiorem kategorii względnych wysokości tonu oraz A jest zbiorem kategorii akcentów. Zbiór K został opracowany w wyniku analizy subiektywnej i składa się z 7 kategorii przebiegu wysokości tonu w obrębie sylaby: 1) równej, 2) rosnącej, 3) opadającej, 4) słabo opadająco-rosnącej, 5) mocno opadająco-rosnącej, 6) słabo rosnąco-opadającej oraz 7) mocno rosnąco-opadającej. Zbiór L składa się z 3 kategorii reprezentujących lokalizację początku wysokości tonu segmentu bieżącego względem wysokości tonu końca segmentu poprzedniego: 1) percepcyjnie niższej, 2) percepcyjnie równej, 3) percepcyjnie wyższej (Steffen-Batogowa 1996, 30). Zbiór kategorii akcentów A jest dwu elementowy: 1) akcent, 2) brak akcentu.

Każdy segment warstwy zestrojów akcentowych obejmuje jeden lub więcej segmentów intonacyjnych, spośród których dokładnie jeden jest ma akcent. Granice zestrojów akcentowych przypadają wyłącznie w miejscach granic leksów. Korzystając z analizy dystrybucyjnej (strukturalizm) zestrojów akcentowych Batogowa wyodrębniła 26 intonemów (Steffen-Batogowa 1996, 74-96). Identyfikatory intonemów są etykietami segmentów w warstwie jednostek tonalnych. Najwyższa warstwa, warstwa fraz intonacyjnych, grupuje zestroje akcentowe należące do tej samej frazy intonacyjnej.

Demenko 1999

Demenko (1999) zaadaptowała do języka polskiego anotację Szkoły Brytyjskiej w wariancie, który zaproponowali O'Connor i Arnold (1973) oraz Jassem (1996b). Przyjmując strukturę

frazy intonacyjnej Jassema (por. wyrażenie 5.2 na stronie 60) Demenko bada dystynktywność 9-ciu kategorii melodii rdzennych: 1) opadającej pełnej (`'HL'`), 2) opadającej niskiej (`'ML'`), 3) opadającej wysokiej (`'HM'`), 4) opadającej ekstra-niskiej (`'xL'`), 5) rosnącej niskiej (`'LM'`), 6) rosnącej wysokiej (`'MH'`), 7) rosnącej pełnej (`'LH'`), 8) rosnąco-opadającej (`'LHL'`, `'MHL'`) 9) równej (`'MM'`) oraz dwóch kategorii melodii przedrdzennych silnych: 1) wysokiej (`'H'`), 2) niskiej (`'L'`). W badaniach dystynktywności Demenko stosuje m.in. kryterium percepcyjne (Demenko 1999, 73) oraz kryterium statystyczne (Demenko 1999, 152).

Korpus mowy do badań otrzymano dwuetapowo. Najpierw nagrano 60 fraz intonacyjnych o różnorodnej strukturze wypowiedzianych przez fonetyka. Następnie poproszono 26 osób (studentów) o powtórzenie (imitację) nagranych wypowiedzi. Intersubiektywne kryterium percepcyjne sprawdzono na grupie 20-tu słuchaczy. Najniższą dystynktywność stwierdzono dla melodii rdzennych `'LM'`, oraz `'MH'`.

Karpinski 2002

W 2002 roku ukończono prace nad korpusem mowy do badań nad intonacją języka polskiego „PoInt" (Karpiński 2002). PoInt zawiera ponad 30 godzin nagrań, w których uczestniczyło blisko 50 mówców (w przybliżeniu równa liczba kobiet i mężczyzn) wymawiających trzy typy wypowiedzi: 1) mowę czytaną, 2) monologi pół-spontaniczne oraz 3) dialogi. Na potrzeby korpusu PoInt opracowano anotację intonacyjną wzorowaną na anotacji Jassema (2002b) (Karpiński 2002).

Anotacja intonacyjna korpusu PoInt oparta jest na segmentacji nieciągłej rozłącznej, w której anotowane są wyłącznie fragmenty frazy intonacyjnej znajdujące się w obrębie melodii rdzennej. (Karpiński 2002) stosuje 8 etykiet reprezentujących relatywne (względem zakresu wysokości głosu mówcy) wysokości tonu: 1) niski (`'L'`), 2) średni (`'M'`), 3) wysoki (`'H'`), 4) ekstra-niski graniczny (`'xL|'`), 5) niski graniczny (`'L|'`), 6) średni graniczny (`'M|'`), 7) wysoki graniczny (`'H|'`), 5) extra-wysoki graniczny (`'xH|'`). Tony graniczne (podobnie jak w ToBI) występują na końcu frazy. W oparciu o anotację korpusu PoInt wykonano analizę porównawczą intonacji polskiej i angielskiej (Grabe i Karpiński 2003) oraz analizę polskich melodii rdzennych (Francuzik i inni 2005). Zbliżonej, 5-cio poziomowej (wyłączając etykiety graniczne) anotacji użyli Durand i inni (2002) w analizie intonacyjnej pytań polskich.

Jassem 2003

Jassem (2003a) proponuje anotację intonacyjną dla języka polskiego opartą na własnym wariancie gramatyki Szkoły Brytyjskiej (por. wyrażenie 5.2 na stronie 60). W oparciu o założenie, że język polski ma akcent melodyczny (por. Jassem 1962; Dogil 1995) Jassem proponuje zbiór melodii przedstawiony w tabeli 5.8. Jassem (2003a) dla każdej melodii podaje częściowo sformalizowany fonetyczno–akustyczny opis oczekiwanego przebiegu częstotliwości podstawowej. W pierwszej kolumnie tabeli 5.8 zamieszczono dodatkowe cztero lub sześcioliterowe identyfikatory nadane w niniejszej pracy melodiom Jassema. W identyfikatorach tych pierwsza para liter określa kategorię gramatyczną melodii (*weak*, *strong* lub *nuclear*), druga para liter określa przebieg melodii (*level*, *rising*, *falling*, *falling-rising* lub *rising-falling*) a trzecia opcjonalna para określa lokalizację melodii względem melodii sąsiednich lub granic wysokości tonu podstawowego mówcy (*below*, *above*, *high*, *wide* lub *low*).

Według Jassema (2003a) gramatykę polskiej frazy intonacyjnej przedstawia wyrażenie regularne:

```
/(wPT)?(sPT)*(NT)/,
```
(5.8)

Tabela 5.8: Etykiety tonalne Jassema dla języka polskiego.

Melodia	Kategoria gram.	Przebieg	Lokalizacja
'weleab'	słaba	równy	nad następną
'welebe'	słaba	równy	pod następną
'weribe'	słaba	rosnący	pod następną
'stleab'	silna	równy	nad następną
'stlebe'	silna	równy	pod następną
'stfaab'	silna	opadający	nad następną
'stfabe'	silna	opadający	pod następną
'striab'	silna	rosnący	nad następną
'stribe'	silna	rosnący	pod następną
'stfrab'	silna	opadająco-rosnący	nad następną
'stfrbe'	silna	opadająco-rosnący	pod następną
'strfab'	silna	rosnąco-opadający	nad następną
'strfbe'	silna	rosnąco-opadający	pod następną
'nule'	rdzenna	równy	nie dotyczy
'nufahi'	rdzenna	opadający	wysoka
'nufawi'	rdzenna	opadający	pełna
'nufalo'	rdzenna	opadający	niska
'nurihi'	rdzenna	rosnący	wysoka
'nuriwi'	rdzenna	rosnący	pełna
'nurilo'	rdzenna	rosnący	niska
'nufr'	rdzenna	opadająco-rosnący	nie dotyczy
'nurf'	rdzenna	rosnąco-opadający	nie dotyczy

gdzie 'wPT' reprezentuje dowolną melodię słabą, 'sPT' dowolną melodię silną oraz 'NT' dowolną melodię rdzenną. Jassem dopuszcza występowanie niejednoznaczności w analizie indukcyjnej, tj. sytuacji, w których danemu sygnałowi akustycznemu przypisuje się więcej niż jedną anotację intonacyjną na podstawie anotacji sygnałowej oraz fonetycznej. W takich przypadkach decyzja o wyborze wariantu anotacji wspomagana jest wiedzą na temat pozycji akcentu potencjalnego. W rozdziale 10 przedstawiono korpus, w którym zastosowano anotację Jassema.

Oliver 2007

Oliver i Clark (2005) oraz Oliver (2008) proponuję metodę pół-automatycznego tworzenia zbioru etykiet anotacji intonacyjnej na podstawie korpusu mowy. Danymi wejściowymi algorytmu Oliver są: zbiór sygnałów mowy, zbiór sylabizacji danych sygnałów mowy oraz anotacja intonacyjna użyta w korpusie PoInt. Metoda tworzenia zbioru etykiet obejmuje cztery etapy (Oliver 2008, 138):

1. parametryzację — dla każdej sylaby akcentowanej generowany jest wektor rzeczywisty obejmujący: 1) odległość czasową między początkiem sylaby akcentowanej a maksimum F_0 sylaby akcentowanej, 2) czas trwania sylaby akcentowanej wraz z sylabą poprzedzającą oraz sylabą następującą, 3) rozpiętość znormalizowanego F_0, 4) parametry anotacji Tilt opisanej w sekcji 4.4,

2. analizę skupień SOM — dla danego zbioru wektorów po parametryzacji stosowana jest analiza skupień metodą SOM (*Self-Organising Maps*) (Kohonen 2001),

3. analizę skupień HAC — dla danego zbioru skupień otrzymanego metodą SOM przeprowadza się analizę HAC (*Hierarchical Agglomerative Clustering*),

4. wybór klastrów wynikowych (przez eksperta) na podstawie analizy dendrogramu otrzymanego w wyniku analizy HAC.

W wyniku zastosowania powyższej metody na wypowiedziach czytanych oraz pół-spontanicznych z korpusu PoInt, Oliver proponuje 3-elementowy zbiór etykiet anotacji tonalnej: 1) rosnąco-opadającą (`'RF'`), 2) opadającą (`'F'`), 3) rosnącą (`'R'`) (Oliver 2008, 141).

Wagner 2008

W pracach Demenko i inni (2006), Wagner (2008) oraz Wagner (2009) przedstawiono anotację intonacyjną rozwijaną na potrzeby systemu syntezy mowy języka polskiego. Punktem wyjścia dla prac Demenko i Wagner była anotacja intonacyjna systemu syntezy mowy BOSS (Klabbers i inni 2001) oraz korpus mowy czytanej (jeden lektor) dla głosu polskiego w systemie BOSS.

Segmentacja przedstawiona przez Wagner (2008) składa się z trzech warstw : 1) warstwy akcentów melodycznych (*pitch accents*), 2) warstwy fraz (*boundary tones*, '2'), 3) warstwy zdań (*boundary tones*, '5'). Warstwa 1 jest niższa od warstwy 2, warstwa 2 jest niższa od warstwy 3. Kotwice segmentów warstwy 2 i 3 przypadają wyłącznie w miejscach granic leksów. Kotwice segmentów warstwy 1 przypadają dodatkowo w miejcach granic samogłosek.

Wagner proponuje 7 etykiet dla akcentów melodycznych: 1) opadający z akcentem na pierwszej sylabie (`'H*L'`), 2) opadający z akcentem na ostatniej sylabie (`'HL*'`), 3) rosnący z akcentem na pierwszej sylabie (`'L*H'`), 4) rosnący z akcentem na ostatniej sylabie (`'LH*'`), 5) rosnąco-opadający z akcentem na najwyższej sylabie (`'LH*L'`), 6) równy, o wysokości tonu różnej od poprzedzającej sylaby (`'LI'`), 7) równy, o wysokości tonu takiej samej jak dla sylaby poprzedzającej (`'LD'`)(Wagner 2008, 115-117). Identyfikatory etykiet akcentów melodycznych zostały zbudowane analogicznie do etykiet ToBI, tj. zastosowano dwie wysokości „H" oraz „L" oraz znak „*" wskazujący pozycję sylaby akcentowanej. W odróżnieniu od ToBI etykietuje się przebiegi dwukierunkowe (`'LH*L'`), podobnie jak np. w BT. W warstwie fraz występują dwie etykiety reprezentujące przebieg wysokości tonu na ostatnim leksie frazy: 1) rosnący (`'?'`) oraz 2) opadający (`'.'`) (Wagner 2008, 120). W warstwie zdań Wagner proponuje 6 etykiet w postaci par uporządkowanych ze zbioru {`'.'`,`'?'`}×{`'.'`,`'?'`,`'!'`}. Pierwszy element etykiety zdaniowej określa przebieg wysokości tonu na pierwszym akcentowanym wyrazie w zdaniu. Drugi element etykiety zdaniowej określa przebieg wysokości tonu na ostatnim wyrazie w zdaniu. Ogólna koncepcja etykiet fraz oraz zdań została odziedziczona po systemie BOSS i wykazuje podobieństwo do anotacji PROLAB.

W pracach Wagner (2008) oraz Wagner (2009) zastosowano szereg statystycznych kryteriów dystynktywności uzasadniając proponowany system fonologiczny.[10] W cytowanych pracach nie pokazano gramatyki proponowanych etykiet.

5.2 Fonologiczna analiza tonalna

Zgodnie z modelem procesu komunikacji głosowej (por. strona 18), w układzie analizy intonacji wyróżniamy trzy warstwy: sygnałową, fonetyczną oraz fonologiczną. (W pewnych

[10] W testach dystynktywności nie brano pod uwagę etykiet `'LI'` oraz `'LD'` ze względu na ich niejasną interpretację w kategoriach akcentu melodycznego.

przypadkach warstwa fonetyczna może być pominięta.) Algorytmy warstw sygnałowej oraz fonetycznej opisano w rozdziałach odpowiednio 3 oraz 4. W bieżącym rozdziale przedstawiono wybrane techniki stosowane w implementacji warstwy fonologicznej, ze szczególnym uwzględnieniem układów budowanych w celu analizy mowy polskiej.

W warstwie fonologicznej układu analizy intonacyjnej wyodrębnimy trzy grupy algorytmów: 1) segmentacji intonacyjnej, 2) klasyfikacji intonacyjnej oraz 3) parsingu intonacyjnego. Algorytmem **segmentacji intonacyjnej** nazywamy algorytm, który dla danej anotacji fonetycznej (lub sygnałowej) zwraca segmentację anotacji intonacyjnej. Do algorytmów segmentacji intonacyjnej zaliczamy detekcję wydatności (*prominence detection*), detekcję granicy frazowej[11] (*phrase boundary detection*) oraz detekcję zdarzeń prozodycznych (*prosodic event detection*). Algorytmem **klasyfikacji intonacyjnej** nazywamy algorytm, który dla danej anotacji fonetycznej (lub sygnałowej) oraz danego segmentu zwraca etykietę anotacji intonacyjnej dla segmentu. Algorytmem **parsingu intonacyjnego** nazywamy algorytm, który dla danej anotacji fonetycznej (lub sygnałowej) wyznacza anotację fonologiczną poprzez przeszukiwanie (optymalizacja) w przestrzeni rozwiązań zawierającej anotacje o niezależnie zmieniających się segmentacjach oraz etykietyzacjach. Zgodnie z powyższym podziałem warstwa fonologiczna układów analizy intonacyjnej ma budowę dwuczłonową (kolejno następuje segmentacja intonacyjna oraz klasyfikacja intonacyjna) albo jednoczłonową (parsing intonacyjny). Oczekiwana skuteczność układu dwuczłonowego jest iloczynem skuteczności obu członów (zakładając niezależność odpowiednich zmiennych losowych).

Problem pomiaru skuteczności układów analizy intonacyjnej jest powiązany z problemem określenia odległości anotacyjnej (definicja na stronie 10) na podzbiorze anotacji fonologicznych. Często stosuje się pomocnicze założenie, że porównywane anotacje zawierają identyczne warstwy sylab.

Najbardziej popularną miarą skuteczności układów analizy intonacyjnej jest **zgodność** (*agreement*). Jeśli p oraz p' są ścieżkami sylab w porównywanych ze sobą anotacjach fonologicznych A oraz A', to przez zgodność rozumiemy wartość:

$$A = \frac{|\{i : \psi(A, p[i]) = \psi(A', p'[i])\}|}{|p|}, \tag{5.9}$$

gdzie funkcja ψ podaje etykietę intonacyjną dla danego segmentu sylabicznego (np. pozyskaną z innej warstwy anotacji). W większości publikacji autorzy podają wartość średnią (oraz przedziały ufności) miary zgodności na zbiorach testowych. Pomimo licznych propozycji udoskonalenia miary zgodości (por. np. Carletta 1996; Rosenberg 2009, 204), podstawowa jej definicja, ze względu na popularność, jest obecnie jedyną miarą skuteczności (z wieloma zastrzeżeniami podanymi poniżej) użyteczną w zastosowaniach przeglądowych.

Wyniki pomiaru skuteczności układów analizy intonacyjnej podawane w literaturze należy zestawiać z dużą ostrożnością.[12] Porównując skuteczność układów analizy intonacyjnej należy uwzględnić:

1. analizowany korpus mowy — język, rodzaj wypowiedzi (czytana, spontaniczna), liczba mówców, kwalifikacje mówców (fonetyczne, lektorskie, przeciętne), czas trwania analizowanych sygnałów (pojedyncza fraza, kilka fraz, nagranie wielominutowe),

[11] Pojęcie frazy intonacyjnej jest definiowane w sposób zależny od przyjętej fonologicznej anotacji tonalnej.

[12] Sytuacja jest tutaj inna niż np. w rozpoznawaniu mowy, gdzie istnieją ogólnie przyjęte sposoby (m.in. WER na korpusie Switchboard) oraz narzędzia (np. SCTK — *NIST Scoring Toolkit*) pomiaru skuteczności.

2. wybór zbioru testowego — losowy, losowy z zawężeniem do pewnego rodzaju kontekstów (np. sylab akcentowanych),

3. rodzaj anotacji intonacyjnej — ToBI, BT, INTSINT, itd.,

4. liczba etykiet intonacyjnych — wynikająca z anotacji, zgrupowanie etykiet w anotacji, etykiety binarne,

5. tryb przetwarzania danych (on-line, off-line) — zdecydowana większość układów opisanych w literaturze to układy off-line; w niektórych układach, dla zmniejszenie błędu analizy, stosuje się ręczną korektę anotacji warstw pośrednich.

Ze względu na ograniczoną objętość bieżącego przeglądu do zupełnego minimum ograniczono opisy ogólnie znanych technik rozpoznawania wzorców. Techniki rozpoznawania wzorców opisano głównie na podstawie monografii, które opublikowali Huang i inni (2001) oraz Benesty i inni (2008). Polskojęzyczne opracowania na temat rozpoznawania wzorców opublikowali w ostatnich latach m.in. Kornacki i Ćwik (2008) oraz Krzyśko i inni (2008).

5.2.1 Techniki oparte na wiedzy

(Petrillo 2003) przedstawił oparty na wiedzy układ segmentacji intonacyjnej. W algorytmach segmentacji Petrillo umieścił kilkanaście parametrów, które następnie optymalizował metodami symulowanego wyżarzania oraz programowania genetycznego. Otrzymano wyniki w granicach 57% zgodności.

(Tamburini 2003) zaproponował układ segmentacji intonacyjnej oparty na współczynniku wydatności sylaby

$$Prom = \max(en_{500-2000} \cdot dur, en_{ov} \cdot ev_{amp}), \tag{5.10}$$

gdzie $en_{500-2000}$ jest energią sylaby w paśmie 500-2000Hz, *dur* jest czasem trwania sylaby, en_{ov} jest energią ośrodka fonetycznego sylaby oraz ev_{amp} jest amplitudą TILT dla sylaby. Maksima lokalne współczynnika Prom w porządku segmentów sylabicznych identyfikują segmenty akcentowane. Raportowana przez Tamburiniego zgodność układu wynosi 80.2%.

Hirst (2007) przedstawił oparty na wiedzy układ analizy INTSINT. Układ Hirsta zaliczamy do grupy algorytmów klasyfikacji intonacyjnej (segmentacja w INTSINT pochodzi wprost z anotacji MOMEL). Układ zaimplementowano w środowisku Praat (Boersma i Weenink 2008) implementując reguły analizy określone wraz z anotacją INTSINT. Ze względu na uniwersalność językową oraz obiektywną definicję anotacji INTSINT, układ Hirsta nie wymaga adaptacji do nowych języków.

Braunschweiler (2006) wykonał Prosodizer — układ oparty na wiedzy, zawierający algorytmy segmentacji oraz klasyfikacji dla ToBI. Układ Braunschweilera opracowano w oparciu o anotacje subiektywne z korpusu systemu syntezy mowy: sygnałowe, głoskowe, sylabiczne oraz anotacje morfo–syntaktyczne (algorytm indukcyjno–dedukcyjny). Algorytm analizy Prosodizera opiera się na dwustopniowej detekcji segmentów o przebiegach sygnału mowy charakterystycznych dla każdej z etykiet ToBI (reguły detekcji opracowano na podstawie analizy fonetycznej korpusu mowy). W pierwszym stopniu detekcji zastosowano anotację sygnałową, w drugim stopniu detekcji zastosowano anotację fonetyczną (sylabiczną). Testy zgodności tonów granicznych (segmentacja intonacyjna) oraz akcentów melodycznych w przypadku języka angielskiego wykazały odpowiednio 68% oraz 60%. Na uwagę zasługuje przyjęcie przez Braunschweilera pełnego zbioru etykiet ToBI. Dla języka niemieckiego analogiczne zgodności

wynosiły odpowiednio 71% oraz 65%. Układy oparte na wiedzy stanowią obecnie zdecydowaną mniejszość wśród proponowanych układów analizy intonacyjnej. Do zalet układów opartych na wiedzy należy możliwość rozwijania układu bez dostępu do kosztownych zasobów (anotowanych korpusów mowy) oraz interpretowalność algorytmów i parametrów układu w kategoriach zastosowanej wiedzy. Podstawową wadą układów opartych na wiedzy są wysokie koszty rozwoju algorytmów oraz dostosowywania do nowych języków (pomijając np. INTSINT).

5.2.2 Niejawne modele Markowa

Jensen i inni (1993) przedstawili układ parsingu intonacyjnego anotacji Szkoły Brytyjskiej oparty na HMM. W układzie tym nie wykorzstano warstwy fonetycznej. Ciąg obserwacji HMM składa się z F_0 oraz pierwszej i drugiej różnicy sąsiednich wartości F_0: ΔF_0, $\Delta^2 F_0$. Rozkłady emisji stanów modelowane są rozkładem CDGMM. Na uwagę zasługuje modyfikacja standardowego CDGMM, w wyniku której dla każdego rozkładu składowego o średniej μ generowane są dwa dodatkowe rozkłady składowe o średnich 0.5μ oraz 2μ. Celem modyfikacji rozkładu jest objęcie popularnych błędów analizy sygnałowej (*pitch-doubling*, *pitch-doubling*). Każdy wzorzec wysokości tonu BT reprezentowany jest przez oddzielny HMM liczący od dwóch do sześciu stanów oraz od dwóch do dziewięciu składników CDGMM. Układ Jensena osiągnął 67.4% zgonności miejsca i rodzaju etykiet BT (77.4% zgodności dla melodii rdzennych) na korpusie mowy pojedynczego mówcy.

Fach i Wokurek (1995) opisuje układ klasyfikacji anotacyjnej dla anotacji ToBI. Analizie poddano korpus języka niemieckiego. Liczbę etykiet anotacji ToBI zredukowano do dwóch (L*H oraz H*L). W układzie pominięto warstwę analizy fonetycznej. Dla i-tego segmentu anotacji sygnałowej (przebiegu F_0) wyznaczany jest N-wymiarowy wektor zawierający wartości F_0 segmentów o indeksach od i do $i + N - 1$. Każdej z (dwóch) etykiet przyporządkowano HMM o pięciu stanach z topologią bez nawrotów i możliwością pominięcia następnego stanu. Układ Facha osiągnął 81% zgodności na sylabach akcentowanych dla N=3 na korpusie mowy pojedynczego mówcy.

proponują specjalizowany model analizy intonacyjej statystyczny do rozpoznawania intonacji oparty na HMM. Ostendorf i Ross (1997) proponują model oparty na HMM, który wyróżnia się rozbudowaną funkcją rozkładu prawdopodobieństwa emisji stanów obejmujący: zakres F_0, kontur F_0 w granicach sylaby oraz czas trwania sylaby. Wektor obserwacji zbudowany jest na podstawie specjalizowanej anotacji fonetycznej, w której generowaniu wykorzystuje się transkrypcję ortograficzną w celu zlokalizowania sylab fonologicznych w sygnale mowy. Przewidzano wariant indukcyjno–dedukcyjny algorytmu Ostendorf i Ross (1997), w którym na podstawie wejściowej transkrypcji ortograficznej określane są sylaby mające akcent potencjalny. W pracy wykorzsytano anotację ToBI, przy czym liczbę etykiet zmniejszono do 9-ciu (poprzez scalenie): 4 na poziomie akcentu melodycznego, 2 na poziomie tonu granicznego oraz 3 na poziomie akcentu frazowego. Sumaryczna liczba stanów emitujących w proponowanym modelu wynosi ok. 600. Opisany parsing intonacyjny model testowano na sygnale mowy jednego mówcy (lektor radiowy). Dodatkowo przyjęto za ustalone etykiety akcentu frazowego. W powyżej opisanych warunkach osiągnięto 63% zgodności w zakresie granic frazowych oraz 85% zgodności w zakresie rodzaju akcentu melodycznego.

HMM zastosowano w kontekście prac na modelem Tilt Wright i Taylor (1997), Taylor (2000). Model Taylora wykonuje segmentację intonacyjną na potrzeby analizy Tilt opierając się na czterech trzystanowych CDHMM modelujących odpowiednio akcent (*accent*), granicę

(*boundary*), połączenie (*connection*) oraz ciszę (*silence*). Proponowane modele nie uwzględniają lokalizacji segmentów głoskowych i sylabicznych przyjmując wektor obserwacji złożony ze standaryzowanej wartości F_0, wartości ΔF_0, $\Delta^2 F_0$ oraz energii chwilowej. Taylor (2000) testuje układ na korpusach zawierających dużą liczbę mówców oraz nagrania w różnorodnych warunkach akustycznych otrzymując zgodność w zakresie od 60% do 80%.

Ananthakrishnan i Narayanan (2005) proponują stosowanie w analizie intonacyjnej *wielowarstwowego* modelu HMM. Dalsze rozszerzania modelu HMM na potrzeby analizy intonacyjnej zaproponowali m.in. Inanoglu i Young (2005), Chen i inni (2006) oraz Sridhar i inni (2008).

W pracy Wypych (2005) zaproponowano indukcyjno–dedukcyjny układ analizy intonacyjnej dla języka polskiego oparty na CDHMM. Układ jest implementacją parsingu intonacyjnego dla anotacji zaproponowanej w pracy Jassem (2003a). W układzie zastosowano specjalizowaną anotację fonetyczną (anotację tę, wraz z późniejszymi rozszerzeniami, opisano w rozdziale 8) oraz reguły akcentu leksykalnego. Topologię HMM określono na podstawie gramatyki Jassema (por. wyrażenie 5.8 na stronie 73). Dla każdej etykiety anotacji przyporządkowano dwu (przebiegi jednokierunkowe) lub trzystanowe (przebiegi dwukierunkowe) modele HMM z możliwością pominięcia stanu. Modele HMM dla etykiet połączono w jeden HMM dodając krawędzie o niezerowym prawdopodobieństwie między stanem początkowym oraz końcowym HMM etykiety w miejscach, gdzie było to dopuszczalne przez gramatykę Jassema. Układ osiągnął zgodność 78.2% w zakresie lokalizacji sylab akcentowanych melodycznie na niewielkim korpusie testowym. W rozdziale 10 przedstawiono nowe podejście do problemu parsingu intonacyjnego, w którym uwzględniono m.in. doświadczenia z prac nad układem Wypych (2005).

5.2.3 Drzewa klasyfikacji i regresji

W pracy Wightman i Ostendorf (1992) przedstawiono indukcyjno–dedukcyjny układ parsingu intonacyjnego oparty na CART. Na podstawie specjalizowanej anotacji fonetycznej (m.in. ekstrema F_0 w obrębie sylaby oraz różnice F_0 sylab sąsiadujących) oraz etykiet anotacji wyżej-poziomowych (pozycja sylaby w leksie, akcent leksykalny sylaby) drzewo CART pozwala dla każdego segmentu określić etykiety kandydujące. W następnym etapie wyznaczana jest optymalna ścieżka wśród etykiet kandydujących za pomocą statystycznej (n-gramowej) gramatyki etykiet ToBI. Zgodność układu wyniosła 81% na głosie jednego mówcy.

W pracy Hirschberg i Nakatani (1998) przedstawiono układ segmentacji intonacyjnej oparty CART. W układzie tym drzewo CART klasyfikuje kolejne 10 milisekundowe segmenty segmentacji sygnałowej jako INPHRASE (należące do frazy intonacyjnej) oraz INBREAK (należące do granicy między frazami intonacyjnymi). Klasyfikacja odbywa się na podstawie specjalizowanej anotacji sygnałowej, której etykiety opisują standaryzowane RMS energii w obrębie 150 ms, średnie standaryzowane F_0 w obrębie 190 ms oraz wartość autokorelacji dla bieżącego segmentu (10 ms). Hirschberg i Nakatani (1998) kierują wiele uwagi na testowanie zależności między doborem zbioru uczącego oraz wynikami pracy układu na zbiorze trenującym. Zgodność mierzono na korpusie mowy czterech mówców zawierającym mowę czytaną oraz spontaniczną. Otrzymane wyniki mieszczą się w granicach od 64% do 91%. Najniższe wyniki zgodności otrzymano w przypadku, gdy CART uczony na mowie czytanej testowany był an mowie spontanicznej. Na podstawie powyższego klasyfikatora zbudowano algorytm segmentacji intonacyjnej przyjmując, że sekwencja kolejnych trzech ramek sklasyfikowanych

jako INBREAK lokalizuje granicę akcentową.[13] Miara F dla otrzymanego układu segmentacji wyniosła w testach od 0.278 do 0.412.

Sun (2002a) stosuje CART w indukcyjno–dedukcyjnym układzie klasyfikacji intonacyjnej. Danymi wejściowymi jest tonalna anotacja fonetyczna oparta na segmentacji sylabicznej oraz anotacja wyżej-poziomowa zawierająca m.in. akcent leksykalny, część mowy oraz pozycję sylaby w leksie. Danymi wyjściowymi jest niepełna anotacja ToBI, w której liczbę kategorii akcentu melodycznego sprowadzono do trzech poprzez scalenie (inne warstwy segmentacji ToBI nie wysępują). Zgodność uzyskana przez układ CART w pracy Sun (2002a) wynosi 84.26% na korpusie wielu mówców.

W pracach Wagner (2008) oraz Wagner (2009) zaprezentowano oparte na CART układy segmentacji oraz klasyfikacji intonacyjnej mowy polskiej. W układzie segmentacji intonacyjnej zastosowano specjalizowaną anotację fonetyczną opartą na segmentacji sylabicznej (sylaby fonologiczne), której etykiety zawierają 5 parametrów: 1) całkowitą zmianę F_0 w obrębie sylaby, 2) względny czas trwania sylaby, 3) względny czas trwania ośrodka fonologicznego sylaby, 4) Tilt, 5) maksymalną wartość F_0 w obrębie sylaby. Podobnie jak w Rapp (1998a) czasy trwania 2 i 3 liczone są względem sumy oczekiwanych (średnia liczona na korpusie) czasów trwania głosek wchodzących w skład sylaby (ośrodka). W układzie klasyfikacji intonacyjnej zastosowano inne etykiety anotacji fonetycznej, w skład których weszło 8 parametrów: 1) amplituda wzrostu F_0, 2) amplituda spadku F_0, 3) względne średnie F_0, 4) względne maksymalne F_0, 5) względne minimalne F_0, 6) Tilt, 7) amplituda Tilt oraz 8) współczynnik kierunku oparty na średnim F_0 oraz amplitudzie Tilt. Analizę fonetyczną wykonano pół-automatycznie — analiza obejmuje m.in. manualną korektę wyników ekstrakcji F_0. Anotacja intonacyjna zawiera pięć etykiet akcentów melodycznych: LH*, L*H, H*L, HL*, LH*L (por. z sekcją 5.1.8). Uczenie oraz testowanie przeprowadzono na korpusie mowy czytanej lektorskiej pojedynczego mówcy (lektor) zawierającym 1052 wypowiedzi (15566 sylab), anotowanym przez jednego eksperta (analiza subiektywna). Opublikowane wyniki zgodności wynoszą 79.13% dla zadania segmentacji intonacyjnej (dotyczy wyłącznie warstwy akcentów melodycznych) oraz 81.61% dla zadania klasyfikacji intonacyjnej. Segmentacja intonacyjna wykonywana jest wyłącznie dla sylab z akcentem potencjalnym (z tego względu ten algorytm analizy zaliczamy do indukcyjno–dedukcyjnych). Klasyfikacja intonacyjna wykonywana jest wyłącznie na sylabach wcześniej sklasyfikowanych jako akcentowane melodycznie.

5.2.4 Sieci neuronowe

W pracy (Kiessling i inni 1994) zastosowano MLP w układzie klasyfikacji intonacyjnej. Przyjęto topologię MLP $40:40:20:6$, tj. 40 neuronów w warstwie wejściowej, 40 oraz 20 neuronów w kolejnych warstwach ukrytych oraz 6 neuronów w warstwie wyjściowej. Układ działa w oparciu o segmentację głoskowo-sylabiczną zsynchronizowaną z sygnałem mowy za pomocą układu ASR. Na wejściu MLP dawane są parametry sylaby otrzymane ze specjalizowanej anotacji fonetycznej, m.in.: 1) czas trwania sylaby, 2) czas trwania ośrodka fonologicznego sylaby, 3) znormalizowana wartość średnia energii, 4) znormalizowana wartość maksymalna energii, 5) współczynniki regresji liniowej przebiegu F_0 w granicach sylaby, 6) maksimum F_0 oraz 7) minimum F_0. Wyjściowa anotacja fonologiczna oparta jest na segmentacji sylabicznej o 6-cio elementowym zbiorze etykiet intonacyjnych (3 dla sylab akcentowanych oraz 3 dla sylab nieakcentowanych). Układ uczono na zbiorze 6900 zdań wypowiedzianych przez 69 osób (44 głosów męskich, 25 głosów kobiecych). Przedstawiony układ osiągnął 70.4% zgodności na

[13] Podajemy tu wersję uproszczoną, w artykule Hirschberg i Nakatani (1998) sprawdzano podobne, lecz nieco bardziej złożone algorytmy.

korpusie testowym 2100 zdań wypowiedzianych przez 21 osób (12 głosów męskich, 9 głosów kobiecych).

W pracy Taylor (1995b) zastosowano RNN w układzie segmentacji intonacyjnej. Układ Taylora wyznacza granice **zdarzeń intonacyjnych**, tj. akcentów melodycznych oraz tonów granicznych bez zrozróżniania między nimi. Taylor stosuje neuronowe klasyfikatory binarne o topologii Ellmana, w których za dane wejściowe przyjęto przekazywane chronologicznie etykiety anotacji sygnałowej. Stosowana anotacja sygnałowa zawiera energię sygnału (RMS), wygładzone i interpolowane F_0, różnice wygładzonego i interpolowanego F_0 między kolejnymi ramkami oraz 12 współczynników cepstralnych. Klasyfikatory binarne na podstawie współczynników cepstralnych określają dwie segmentalne cechy sygnałowe: *samogłoskowość* oraz *zwarto-głoskowość*. Następnie na podstawie pozostałych etykiet anotacji sygnałowej, samogłoskowości oraz *zwarto-głoskowości* trzeci klasyfikator binarny określa przynależność do zdarzenia intonacyjnego dla każdej 10 ms ramki. Wartości etykiet anotacji sygnałowej są normalizowane poprzez standaryzację oraz pomnożenie przez 0.5 (tak, aby ok. 95% wartości zmiennych mieściła się w przedziale od -1 do +1). Wygładzone wyjście z klasyfikatora determinuje pozycję segmentu zdarzenia intonacyjnego (początek na zboczu dodatnim, koniec na zboczu ujemnym). Zgodność przedstawionego układu wyniosła 85.7% na niewielkim korpusie jednej osoby zawierającym 249 sylab i 112 zdarzeń intonacyjnych.

W pracy (suk Kim i inni 2003) przedstawiono układ segmentacji intonacyjnej oparty na TDRNN. W zastosowaniej sieci TDRNN występuje pięć warstw: 1) warstwa wejściowa (2 neurony), 2) warstwa ukryta (5 neuronów), 3) warstwa ukryta wysokości tonu (1 neuron), 4) warstwa ukryta kontekstu długookresowaego (1 neuron), 5) warstwa wyjściowa. W warstwach 1 oraz 4 zastosowano opóźnienie czasowe o długościach odpowiednio 2 i 18, por. (Waibel i inni 1989). Warstwa 4 zapewnia rekurencyjny przepływ danych z warstwy 3 do warstwy 2. Danymi wejściowymi sieci jest anotacja sygnałowa, której etykiety zawierają znormalizowaną F_0 (skala logarytmiczna) oraz znormalizowaną energię chwilową (skala logarytmiczna). Wyjście sieci TDRNN klasyfikuje każdy segment anotacji sygnałowej jako należący do zdarzenia intonacyjnego (> 0.5) lub nie ($\leqslant 0.5$). Jeśli co najmniej 50% segmentów anotacji sygnałowej obejmowanych przez pewną sylabę uzyskało wynik powyżej 0.5, to przyjmuje się, że segment tej sylaby jest segmentem zdarzenia intonacyjnego. Zbiór uczący układu składał się z ponad 2000 zdarzeń intonacyjnych w sygnale mowy czytanej jednego mówcy (lektor, głos kobiecy). Zbiór testowy obejmowa blisko 7000 zdarzeń intonacyjnych w sygnale mowy czytanej innego mówcy (lektor, głos kobiecy). Prezentowany układ osiągnął 91.9% zgodności w zakresie sylab akcentowanych oraz 91.0% zgodności w zakresie sylab nieakcentowanych..

W pracy (Ren i inni 2004) zaproponowano układ klasyfikacji intonacyjnej będący rozszerzeniem układu przedstawionego w (suk Kim i inni 2003). Do topologii sieci wprowadzono dodatkową warstwę z opóźnieniem czasowym zapewniającą przepływ danych z warstwy 5 do warstwy 2 oraz zwiększono liczbę neuronów w warstwach. Na wejściu zastosowano specjalizowaną anotację fonetyczną opartą na segmentacji sylabicznej, której etykietą jest 16-to wymiarowy wektor rzeczywisty obejmujący 13 współczynników cepstralnych, wysokość tonu, czas trwania oraz energię. Anotacja wyjściowa oparta została na podzbiorze czterech etykiet ToBI: H*, L*, L+H*, !H* (pozostałe etykiety odrzucono ze względu na zbyt małą liczbę przykładów). Zbiór uczący oraz testowy dla prezentowanego układu oparto na mowe czytanej trzech lektorów zawodowych (głosy kobiece, w sumie 208 minut) w proporcjach 78% (uczący) do 22% (testowy). Opisany układ uzyskał 81.21% zgodności a w wyniku dalszych ulepszeń topologii TDRNN 83.64%.

W pracy Ananthakrishnan i Narayanan (2008) zaproponowano indukcyjno–dedukcyjny układ segmentacji intonacyjnej oparty na MLP. Przyjęto topologię MLP $9:25:2$ (9 neuronów w warstwie wejściowej, 25 w warstwie ukrytej oraz 2 neurony w warstwie wyjściowej. MLP otrzymuje na wejściu etykiety specjalizowanej anotacji fonetycznej opartej na segmentacji sylabicznej. Każdej sylabie przyporządkowuje się następujące parametry wyliczane na podstawie anotacji sygnałowej w obrębie sylaby: 1) zakres F_0, 2) różnicę między maksymalnym a średnim F_0, 3) różnicę między średnim a minimalnym F_0, 4) różnicę między średnim F_0 sylaby a średnim F_0 wypowiedzi, 5) znormalizowany czas trwania samogłoski w ośrodku fonologycznym sylaby, 6) czas trwania pauzy po ostatniej sylabie w wyrazie, 7) zakres energii chwilowej, 8) różnicę między maksymalną energią chwilową oraz energią średnią, 9) różnicę między energią średnią oraz minimalną energią chwilową. Oprócz tego układ na wejściu przyjmuje anotację ponadfonologiczną określającą nazwy sylab, pozycje sylab akcentowanych leksykalnie oraz części mowy. Dedukcyjność algorytmu analizy zaproponowanego w pracy Ananthakrishnan i Narayanan (2008) wynika z zastosowania n-gramowego modelu statystycznego, który modeluje prawdopodobieństawo wystąpienia akcentu melodycznego oraz granicy frazowej w danym miejscu warunkowane etykietami wejściowej anotacji ponadfonologicznej. Dodatkowo, podobnie jak w pracy Wightman i Ostendorf (1992), w wynikach kasyfikacji (tutaj ANN) odnajdywana jest ścieżka Viterbiego przy uwzględnieniu ograniczeń n-gramowej gramatyki intonacyjnej. Na wyjściu układu otrzymuje się segmentację akcentową oraz segmentację frazową (frazy pośrednie oraz frazy intonacyjne) ToBI. Zbiór uczący układu przygotowano na podstawie korpusu mowy czytanej sześciu mówców (lektorzy, 3 głosy kobiece i 3 głosy męskie) zawierającego 37047 sylab. W wyniku testowania na zbiorze 7343 sylab stwierdzono zgodność 84.59% w zakresie segmentów akcentowych oraz 91.38% w zakresie granic frazowych.

W pracach Demenko (1999) oraz Demenko i Jassem (1999) opublikowano wyniki dla zastosowania ANN w układach segmentacji oraz klasyfikacji intonacyjnej mowy polskiej. Na podstawie wstępnych testów, w których porównywano skuteczność sieci MLP, sieci RBF oraz sieci probabilistycznych, do dalszych badań przyjęto 3-warstwowy MLP. Za liczby neuronów w warstwie wejściowej oraz wyjściowej przyjęto liczby wymiarów wektorów wejściowego oraz odpowiednio wyjściowego. Liczbę neuronów warstwy ukrytej określono ekspertymentalnie osiągając najwyższą skuteczność układu w przypadku MLP o topologii $8:20:3$ (Demenko 1999, 166). Dane wejściowe MLP w układzie segmentacji intonacyjnej Demenko obliczane są na podstawie specjalizowanej fonetycznej anotacji tonalnej, w której każdemu segmentowi samogłoskowemu przypisywany jest wektor rzeczywisty o następujących wymiarach: 1) F_{vmin} — minimum F_0, 2) F_{vmax} — maksimum F_0, 3) F_{vp} — wartość początkowa F_0, 4) F_{vk} — wartość końcowa F_0. 8-mio wymiarowy wektor wejściowy MLP zawiera czas trwania segmentu samogłoskowego oraz 7 parametrów pozostających w prostej zależności z etykietami anotacji fonetycznej dla samogłoski bieżącej i samogłosek sąsiednich. Z kolei w układzie klasyfikacji intonacyjnej zaproponowanym w tej samej pracy zastosowano MLP o topologii $11:9:5$. W układzie klasyfikacji zastosowano specjalizowaną fonetyczną anotację tonalną zawierającą segmenty samogłosek, melodii oraz fraz intonacyjnych, której etykiety zawierają wartości średnie, początkowe, końcowe i ekstremalne przebiegu F_0 w granicach segmentu. 11-to wymiarowy wektor wejściowy MLP zawiera czas trwania oraz energię ostatniej samogłoski w melodii oraz 9 parametrów pozostających w prostej zależności z etykietami anotacji fonetycznej. Anotacja wyjściowa układu klasyfikacji intonacyjnej Demenko (1999, 164-167) oparta jest na anotacji fonologicznej przyjętej w tej samej pracy (por. sekcja 5.1.8), przy czym ze względu na niewielki korpus uczący liczbę etykiet w anotacji zredukowano do 6-ciu: P (przedrdzenny słaby), L (przedrdzenny silny niski), H (przedrdzenny silny wysoki), R (rdzenny rosnący), F (rdzenny opadający) oraz MM (rdzenny równy). Za zbiór uczący dla

Tabela 5.9: Nowe systemy uczące się w fonologicznej analizie tonalnej.

Nazwa	Autorzy
Random Forest	Schweitzer i Möbius (2009)
Bagging	Sun (2002a), Schweitzer i Möbius (2009)
AdaBoost	Sun (2002a), Chen i inni (2006), Margolis i inni (2010b)
Conditional Random Fields	Kumar i inni (2008), Levow (2008)
Support Vector Machines	Levow (2006), Rosenberg (2009)
Logistic Regression	Rosenberg i Hirschberg (2009), Schweitzer i Möbius (2009)
Maximum Entropy	Brenier i inni (2005), Sridhar i inni (2008)

opisanych układów oparto na korpusie imitacji opisanym na stronie 72. Analizę fonetyczną wykonywano pół-automatycznie (korekta przez eksperta). Zgodność segmentacji intonacyjnej mierzona na podzbiorze testującym korpusu imitacji wyniosła 75.0%. Zgodność klasyfikacji przy testowaniu na korpusie mowy ciągłej jednego mówcy wyniosła 82.0% (na podstawie tabeli 13.3 z pracy Demenko (1999, 167)).

Ostatnio prace nad zastosowaniem ANN w analizie intonacyjnej polskiej opublikowała Wagner (2008, 2009). Stosując ANN zamiast CART, przy zachowaniu pozostałych warunków jak to opisano na stronie 80, Wagner otrzymała następujące wyniki: 81.72% zgodności segmentacji dla sieci MLP o topologii 5:17:1, 81.95% zgodności segmentacji dla sieci RBF o topologii 5:82:1, 77.52% zgodności klasyfikacji dla sieci MLP o topologii 8:15:5 oraz 72.75% zgodności klasyfikacji dla sieci RBF o topologii 8:82:5.

5.2.5 Nowe kierunki rozwoju

W bieżącej sekcji wymienione są nowe techniki analizy intonacyjnej, których skuteczności nie została (jeszcze) potwierdzona w dostatecznie dużej liczbie niezależnych eksperymentów. Ze względu na nieugruntowaną terminologię polską, w bieżącej sekcji stosowana jest terminologia angielska.

W tabeli 5.9 zamieszczono wykaz technik rozpoznawania wzorców, które zastosowano w ostatnich latach z sukcesem w analizie intonacyjnej.

W ostatnich wykonano latach kilka prób zastosowania systemów uczących się pod częściowym nadzorem (*semi-supervised learning*) do problemu segmentacji intonacyjnej. W pracy Chen i inni (2006) zaproponowano indukcyjno–dedukcyjny algorytm segmantacji intonacyjnej oparty na HMM, którego zbiór uczący obejmował 500 zdań segmentowanych intonacyjnie przez eksperta oraz 5412 zdań segmentowanych automatycznie na podstawie transkrypcji ortograficznej. Układ ten osiągnął 92.1% zgodności w przypadku uczenia przy wykorzystaniu wyłącznie segmentacji eksperckich oraz 94% zgodności po poszerzeniu zbioru uczącego o segmentacje automatyczne. (W uczeniu oraz testach wykorzystano korpus mowy czytanej jednego lektora segmentowany przez pojedynczego anotatora). W pracy Levow (2006) zaprezentowano układ segmentacji intonacyjnej oparty na modelu Manifold Regularization. W eksperymencie wykorzystano anotowany intonacyjnie zbiór uczący zawierający 300 sylab oraz nieanotowany intonacyjnie zbiór uczący zawierający 700 sylab. W wyniku uczenia pod

nadzorem układu opartego na modelu Laplacian SVM uzyskano 81.5% zgodności segmentacji wobec 84% zgodności uzyskanej przez analogiczny układ uczony pod nadzorem na dużym korpusie mowy. W pracy Jeon i Liu (2009) przedstawiono indukcyjno–dedukcyjny algorytm segmentacji intonacyjnej oparty na algorytmie *co–training* zastosowanym w odniesieniu do ANN oraz SVM. Zbiór uczący w tym eksperymencie zawierał 573 sylaby anotowane intonacyjnie (dwóch lektorów, głosy męskie) oraz blisko 130 tyś. sylab nieanotowanych intonacyjnie (trzeci lektor, głos męski). Zbiór testowy zawierał 8962 sylaby (dwóch lektorów, głos kobiecy i męski, lektorzy inni niż w zbiorze uczącym). W ramach metody *co-trainingu* testowano alternatywne strategie rozszerzania anotowanego zbioru uczącego na podstawie nieanotowanego zbioru uczącego osiągając maksymalną zgodność w przypadku segmentacji wzorców akcentowych 85.3%. Zgodność analogicznego układu uczonego wyłącznie pod nadzorem na dużym korpusie mowy wyniosła 87.6%.

Część III

Prace badawcze

ROZDZIAŁ 6

Architektura programowa układu

Komponentem nazywamy jednostkę modularyzacji układu (oprogramowania), dla której ściśle określono zbiór interfejsów wymaganych oraz zbiór interfejsów realizowanych (OMG 2010, 147). **Architekturą programową** nazywamy te części układu (oprogramowania), które wspomagają tworzenie, konfigurację oraz łączenie komponentów. **Środowiskiem programowym** nazywamy architekturę programową rozwijaną z zamiarem zastosowania w układach o różnej funkcjonalności.

Układem przetwarzania mowy będziemy nazywać implementację symulującą funkcje dowolnej części głosowego systemu komunikacyjnego (por. sekcja 1). Do najwcześniejszych układów przetwarzania mowy, w których zastosowano dedykowane architektury porgramowe należą: układ rozumienia mowy Hearsay-II (Erman i inni 1980) oraz układ syntezy mowy MITalk (Allen i inni 1987). W latach 80-tych stworzono pierwsze środowiska programowe dla układów przetwarzania mowy, m.in. SFS (Huckvale i inni 1987), Delta (Hertz 1988) oraz HTK (Woodland i Young 1993). Od połowy lat 90-tych zaproponowano co najmniej kilkanaście środowisk programowych dla układów przetwarzania mowy o różnym stopniu specjalizacji (por. sekcja 6.1). Wielokrotnie potwierdzono, że stosowanie środowisk programowych pozwala znacząco zmiejszyć koszty implementacji, rozwoju oraz dystrybucji układów przetwarzania mowy (Cunningham i inni 2002).

W dotychczasowych pracach z zakresu analizy intonacyjnej problem architektury programowej nie zyskał należnej uwagi. W części prac zrezygnowano z integracji układu wykorzystując zbiór niezależnych aplikacji uruchamianych przez eksperta.[1]

Układ rozpoznawania struktur intonacyjnych zaimplementowano w oparciu o środowisko programowe SLOPE (*Spoken Language Ontology Processing Environment*) (Wypych 2001). W 2005 roku, środowisko SLOPE zostało gruntownie przeprojektowane przez autora z uwzględnieniem szerszych wymagań układów analizy oraz syntezy mowy. Kolejne wersje SLOPE zastosowano m.in. w systemie detekcji błędów wymowy oraz systemie tłumaczenia tekstów przez analogię. W sekcjach bieżącego rozdziału przedstawiono wybrane własności środowiska SLOPE w wersji 0.8.3 z lipca 2010 roku.

[1] Obecnie jednym z nielicznych zintegorwanych (bez wykorzystania środowiska programowego) układów analizy intonacyjnej jest AuToBI (Rosenberg 2010).

6.1 Typologia architektur programowych

Przyjmujemy, że podstawowymi wyróżnikami architektury programowej są strategie: 1) reprezentowania anotacji, 2) przekazywania danych między instancjami komponentów oraz 3) aktywacji instancji komponentów.[2]

Wyróżniamy dwie strategie reprezentowania anotacji: specjalizacyjną oraz unifikacyjną. W strategii **specjalizacyjnej** anotacja reprezentowana jest w postaci szeregu prostych struktur danych, które są dobierane stosownie do potrzeb komunikacyjnych niewielkich podzbiorów (w szczególności par) komponentów. W strategii **unifikacyjnej** w obrębie architektury programowej stosowana jest jedna, zunifikowana reprezentacja anotacji.

Wyróżniamy trzy strategie przekazywania danych (sygnałów, anotacji) między instancjami komponentów: jednostkową, masową oraz agregacyjną. W strategii **jednostkowej** dla każdego nadawcy/odbiorcy jawnie przypisany jest zbiór odbiorców/nadawców. W strategii **masowej** wysłanie danych rozgłaszane jest wśród wszystkich instancji komponentów, które następnie decydują o odbiorze na podstawie typu i zawartości danych. W strategii **agregacyjnej** nadawca dodaje dane do współdzielonej struktury danych obserwowanej przez potencjalnych odbiorców. Odbiorca wczytuje dane ze struktury współdzielonej nie usuwając ich.

Wyróżniamy trzy strategie aktywacji (przydziału zasobów obliczeniowych) instancji komponentów: hierarchiczną, heterarchiczną oraz autonomiczną. W przypadku strategii **hierarchicznej** aktywacja następuje według niezmiennego, z góry ustalonego planu zapisanego jawnie w postaci listy wywołań lub wynikającego z ról nadawcy/odbiorcy w przykazywaniu danych. W przypadku strategii **heterarchicznej** w architekturze programowej występuje planista, którego celem jest wypracowanie pożądanych wyników przy minimalnej liczbie aktywacji instancji komponentów. W przypadku strategii **autonomicznej** instancje komponentów decydują o aktywacji same w ramach przypisanych im wątków lub procesów. Strategia autonomiczna stosowana jest m.in. w systemach rozproszonych oraz w systemach wieloagentowych.

Strategie: specjalizacyjna, jawna oraz hierarchiczna obniżają krótko-terminowe koszty opracowania układu (zakładając, że nie jest dostępne odpowiednie środowisko programowe) oraz narzuty związane z architekturą programową. Strategie: unifikacyjna, anonimowa oraz heterarchiczna/autonomiczna obniżają długo-terminowe koszty opracowania i rozwoju układu, zwiększają skalowalność oraz ułatwiają ponowne użycie komponentów.

W tabeli 6.1 zamieszczono zestawienie architektur programowych wybranych układów oraz środowisk programowych stosowanych w przetwarzaniu mowy. W zestawieniu uwzględniono wyłączenie takie architektury programowe, które jednocześnie wspierają przetwarzanie mowy na poziomie sygnałowym oraz fonologicznym.[3]

[2] Dutoit (1997, 58-62), Cunningham (2000) oraz Corkill (2003) przedstawili w swoich pracach analizy porównawcze architektur programowych w układach przetwarzania mowy, języka i sztucznej inteligencji.

[3] Przykładami środowisk programowych, które znajdują zastosowanie na pewnych etapach przetwarzania mowy, lecz nie spełniają przyjętych kryteriów są GStreamer (wyłącznie poziom sygnałowy), Ecasound (wyłącznie poziom sygnałowy), Ellogon (wyłącznie poziomy ponadfonologiczne) oraz NLTK (wyłącznie poziomy ponadfonologiczne).

Tabela 6.1: Architektury oraz środowiska programowe układów przetwarzania mowy.

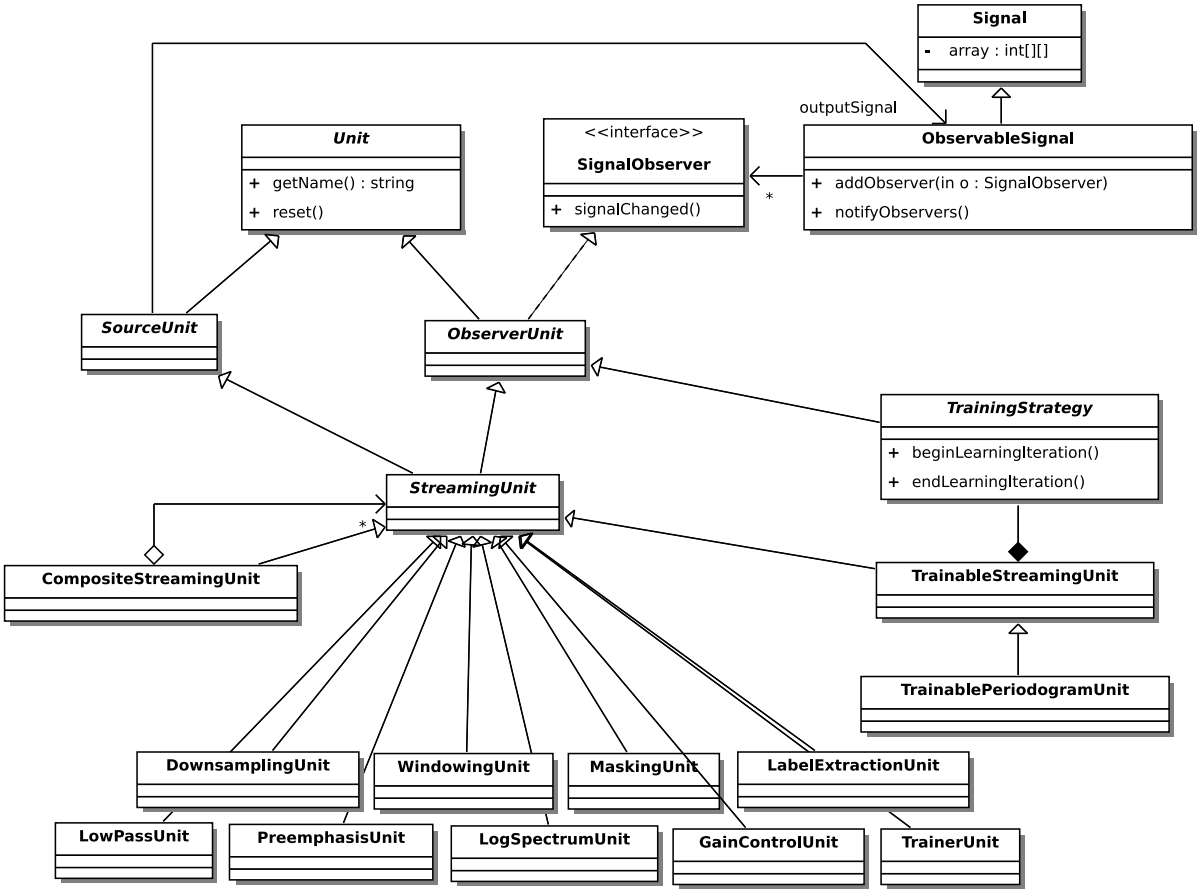
Nazwa	Kategoryzacja	Aktualny opis	Aktualna afiliacja
Hearsay-II	unifikacyjna, agregacyjna, heterarchiczna	Erman i inni (1980)	nierozwijany
Delta	unifikacyjna, agregacyjna, hierarchiczna	Hertz (1988)	nierozwijany
SFS	specjalizacyjna, jednostkowa, hierarchiczna	Huckvale i inni (1987)	Division of Psychology and Language Sciences, UCL
MITalk	specjalizacyjna, jednostkowa, hierarchiczna	Allen i inni (1987)	nierozwijany
Galaxy-II	unifikacyjna, jednostkowa, autonomiczna	Seneff i inni (1998)	Spoken Language Systems Group, MIT
Verbmobil II	unifikacyjna, agregacyjna, autonomiczna	Klüter i inni (2000)	nierozwijany
Festival	unifikacyjna, agregacyjna, hierarchiczna	Taylor i inni (1998a)	CSTR, University of Edinburgh
GATE	unifikacyjna, agregacyjna, hierarchiczna	Cunningham i inni (2002)	University of Sheffield
BOSS 3	unifikacyjna, agregacyjna, hierarchiczna	Breuer i inni (2005)	IKP, University of Bonn
DORIS	unifikacyjna, jednostkowa, autonomiczna	Alain i inni (2009)	INRIA

Architekturą **potokową** będziemy nazywać architekturę ze specjalizacyjną reprezentacją anotacji, jednostkową dystrybucją danych oraz hierarchiczną lub autonomiczną aktywacją. Architekturą **tablicową** (*blackboard*) będziemy nazywać architekturę z unifikacyjną reprezentacją anotacji, agregacyjną dystrybucją danych oraz heterarchiczną lub autonomiczną aktywacją.[4] Architekturą **wieloagentową** będziemy nazywać architekturę z unifikacyjną reprezentacją anotacji, jednostkową dystrybucją danych oraz autonomiczną aktywacją.

Środowisko programowe SLOPE oparte jest na połączeniu architektury tablicowej (integracja komponentów obejmujących wiele poziomów analizy mowy) z architekturą potokową (integracja algorytmów sygnałowych). W SLOPE stosuje się dwie biblioteki obliczeń numerycznych: 1) IPP (*Intel Performance Primitives*) mającą API w języku C oraz 2) Colt (Hoschek 2004) zaimplementowaną w języku Java. SLOPE tworzone jest w języku Java metodą *roundtrip engineering* (utrzymywanie zgodności między modelem UML oraz kodem źródłowym) przy użyciu aplikacji BOUML Pages (2010). Jak wynika z naszych doświadczeń, szereg architektur wymienionych w tabeli 6.1 rozrosło się do poziomu utrudniającego ich zastosowanie w produktach. Priorytetem w budowaniu SLOPE było ograniczanie złożoności oraz narzutów związanych z architekturą programową.

Kody źródłowe środowiska SLOPE umieszczone są w pakiecie pakiecie (przestrzeni nazwowej) slope. W pakiecie slope występuje ponad 200 klas w języku Java oraz ponad 20 wzorców

[4] Corkill (2003) przprowadza interesującą analizę porównawczą architektur tablicowych oraz wieloagentowych.



Rycina 6.1: Architektura potokowa w SLOPE. Diagram klas UML.

klas w języku C++. Pakiet slope obejmujeponad 20 tysięcy linii kodu Java/C/C++ (w tym 24% stanowi dokumentacja).

6.2 Realizacja architektury potokowej

Środowisko programowe SLOPE wspiera architektrue potokową w zastosowaniu do integracji algorytmów przetwarzania mowy na poziomie sygnałowym. Architektura potokowa SLOPE została zaimplementowana w języku C++ jako zbiór komponentów udostępniających funkcje biblioteki programowej IPP 6.0. Biblioteka IPP jest zbiorem algorytmów numerycznych operujących na macierzach danych przy zastosowaniu instrukcji maszynowych SIMD (*Single Instruction Multiple Data*) oraz przetwarzania równoległego w procesorach wielordzeniowych. Biblioteka IPP ma API w języku C oraz jest dostępna dla środowisk Linux oraz Windows. W pracy zastosowano kompilatory C/C++ z pakietu GCC 4.4.

Diagram 6.1 przedstawia klasy abstrakcyjne proponowanej architektury programowej oraz wydziedziczone z nich klasy konkretne zastosowane w niniejszej pracy (por. rozdział 7). Przedstawiona architektura umożliwia implementecję strumieniowych algorytmów analizy anotacyjnej w oparciu o segmentacje proste oraz segmentacje ramkowe.

Klasa Signal reprezentuje sygnał cyfrowy w postaci dwuwymiarowej macierzy numerycznej sparametryzowanej (wzorzec klasy C++) typem komórki. Do najczęściej stosowanych

typów komórek należą: 16-bitowa liczba stałopozycyjna ze znakiem oraz 32-bitowa liczba zmiennopozycyjna[5].

Klasa Unit reprezentuje nazwane, stanowe obiekty funkcyjne. W klasie Unit metoda getName zwraca nazwę obiektu a metoda reset wprowadza obiekt w stan początkowy. Wyróżnia się dwie podklasy klasy Unit: SourceUnit oraz ObserverUnit, które odpowiednio *wysyłają* oraz *odbierają* sygnały. Przekazywanie sygnałów między obiektami klasy Unit zrealizowano przy zastosowaniu wzorca projektowego „obserwator" (*observer*). W celu implementacji wzorca „obserwator", z klasy Signal wywiedziono klasę ObservableSignal, której obiekty są w relacji zero do wielu z obiektami implementującymi interfejs SignalObserver. Obiekt modyfikujący obiekt klasy ObservableSignal zobowiązany jest po modyfikacji wywołać metodę notifyObservers na obiekcie zmodyfikowanym. Metoda notifyObervers wywołuje metody signalChanged na każdym z obiektów implementujących interfejs SignalObserver powiązanych z obiektem ObservableSignal.

Klasa StreamingUnit reprezentuje strumieniowe algorytmy analizy anotacyjnej (por. opis na stronie 12). Anotacja wynikowa w obiektach klasy StreamingUnit reprezentowana jest w postaci ciągu sygnałów-etykiet kolejnych segmentów. Lokalizacje czasowe segmentów anotacji wynikowej określone są na podstawie albo parametrów anotacji ramkowej albo lokalizacji sygnałów wejściowych (w przypadku gdy segmentacja wynikowa jest segmentacją prostą).

Złączenie dwóch lub większej liczby obiektów klasy StreamingUnit za pomocą obiektów klasy ObservableSignal umożliwia tworzenie złożonych algorytmów analizy anotacyjnej w trakcie działania programu. Klasa CompositeStreamingUnit pozwala na traktowanie zbiorów złączonych obiektów klasy StreamingUnit tak samo jak pojedynczych obiektów klasy StreamingUnit.

Klasa TrainableStreamingUnit reprezentuje strumieniowe algorytmy analizy anotacyjnej uczące się pod nadzorem. Zgodnie ze wzorcem projektowym *strategia* (*Strategy*) obiekty klasy TrainableStreamingUnit posiadają obiekty klasy TrainingStrategy, które reprezentują algorytm uczenia się pod nadzorem. Klasa TrainingStrategy jest wywiedziona z klasy ObserverUnit, gdzie zakłada się, że sygnał oczekiwany zawiera pożądaną (uczenie pod nadzorem) wartość sygnału wyjściowego obiektu TrainableStreamingUnit. W klasie TrainingStrategy określono dwie metody: beginLearningIteration oraz endLearningIteration odpowiednio rozpoczynające oraz kończące iteracje (epoki) uczenia.

Algorytmy zaimplementowane w pozostałych klasach diagramu 6.1 opisano w rozdziale 7, w sekcjach 7.1 oraz 7.2.

Ze względu na znaczną liczbę funkcji oraz typów danych w bibliotece IPP, prezentowaną architekturę programową zaimplementowano z zastosowaniem techniki *trait* (Mayers 1995). Technika trait ułatwia zarządzanie wzorcami klas C++ o znacznej liczbie parametrów nie wprowadzając narzutów wydajnościowych w czasie wykonania.

[5]Stosowanie 32-bitowych liczb zmiennopozycyjnych zamiast liczb 64-bitowych daje dodatkowe korzyści wydajnościowe przy wykorzystaniu instrukcji SIMD.

6.3 Reprezentacja wiedzy współdzielonej

Pakiet slope.represent określa reprezentacje wiedzy na wszystkich rozpatrywanych w rozdziale 1 poziomach analizy mowy. Do slope.represent należą 3 pakiety: 1) slope.represent.signal, 2) slope.represent.ontology oraz 3) slope.represent.graph.

Na diagramie 6.2 przedstawiono pakiet slope.represent.signal zawierający reprezentacje sygnałów cyfrowych. Interfejs Signal reprezentuje wielowymiarowy (wielokanałowy) sygnał cyfrowy. Dodatkowo, sygnał ma etykietę napisową nazywaną **specyfikatorem**, która wskazuje właściwą interpretację sygnału (np. ciśnienie akustyczne, widmo chwilowe, MFCC). Interfejs SignalAppendable reprezentuje obiekty, do których można dopisać (poprzez konkatencację) sygnał cyfrowy. Interfejs Timeline reprezentuje listę zsynchronizowanych sygnałów cyforwych o unikalnych specyfikatorach. W przypadku dopisywania sygnału do obiektu implementującego Timeline następuje konkatenacja sygnałów o zgodnych specyfikatorach albo dodanie sygnału do listy (jeśli wcześniej nie było tam sygnału o tym samym specyfikatorze). Klasa Subsignal reprezentuje wycinek innego sygnału cyfrowego. Klasy abstrakcyjne o nazwach zaczynających się od Basic zawierają atrybuty oraz funkcje współużytkowane w klasach potomnych. Klasy konkretne OrthographicSignal, AcousticSignal oraz CircuralAcousticSignal reprezentują odpowiednio sygnał ortograficzny, sygnał akustyczny oraz sygnał akustyczny z buforem okrężnym.

Pakiet slope.represent.ontology określa reprezentację obiektów dziedziny przedmiotowej, tj. fonetyki informatycznej. Na rycinach 6.3 oraz 6.4 przedstawiono ontologie SLOPE z poziomów stosowanych w niniejszej pracy (SLOPE zawiera też ontologie dla wyższych poziomów przetwarzania mowy). Nazwy interfejsów w slope.represent.ontology odnoszą się bezpośrednio do pojęć wprowadzonych w rozdziale 1.

Rycina 6.5 przedstawia diagram UML reprezentacji anotacji w SLOPE. Podobnie jak w biblitece AGLIB (Bird i inni 2007), zastosowano tutaj grafową interpretację segmentacji, w której wierzchołki grafu odpowiadają kotwicom a krawędzie grafu segmentom. W SLOPE reprezentuje się wyłącznie anotacje oparte na segmentacjach kratowych (por. def. 1.18 na str. 9).

Przed opisaniem metod interfejsu Graph krótko przedstawimy cztery interfejsy pomocnicze: Arc, ArcIterator, NodeIterator oraz Walker. Arc jest to etykietowana krawędź w grafie skierowanym. ArcIterator oraz NodeIterator są iteratorami krawędzi oraz wierzchołków grafu w porządku nieustalonym. Walker jest to specjalizowany iterator służący do przechodzenia po krawędziach grafu.

Interfejs Graph reprezentuje graf skierowany. Metody createArc oraz createPath wprowadzają do grafu krawędzie oraz ścieżki (wraz z etykietami). Metody removeArc oraz clear usuwają odpowiednio określoną krawędź oraz wszystkie krawędzie i wierzchołki w grafie. Metody forwardWalker oraz backwardWalker zwracają obiekty Walker pozwalające przechodzić graf zgodnie oraz przeciwnie do kierunku krawędzi w grafie. Metody setNodeLabel oraz getNodelLabel pozwalają przypisywać etykiety wierzchołkom (w klasach wydziedziczonych z Graph przyjmuje się, że kotwice anotacji są etykietami wierzchołków grafu). Metody arcs oraz nodes zwracają odpowiednio ArcIterator oraz NodeIterator grafu. Metody format oraz scan służą do zapisywania oraz odczytywania grafu ze strumieni danych zgodnie z wzorcem projektowym „odwiedzający".

Klasa LinkedGraph realizuje interfejs Graph w oparciu o listy dwukierunkowe, które są dobrze dopasowane do sposobu wykorzystania anotacji w SLOPE. Implementacje klas pomocniczych dla LinkedGraph noszą nazwy: LinkedArc, LinkedArcsIterator, LinkedNodeIterator oraz LinkedWalker.

Interfejs Lattice reprezentuje graf o strukturze kratowej. Metody getFirstNode oraz getLastNode zwracają minimalny oraz maksymalny wierzchołek kraty. Metoda prependArc dodaje do kraty nowy wierzchołek a następnie krawedź od nowego wierzchołka do dotychczasowego wierzchołka minimalnego. Metoda prependArc dodaje do kraty nowy wierzchołek a następnie krawędź od dotychczasowego wierzchołka maksymalnego do wierzchołka nowego. Wprowadzając dodatkowo odpowiednie ograniczenia dla argumentów metod createArc oraz createPath można zapewnić, że graf utrzymywany w Lattice będzie kratą. Klasa LinkedLattice realizuje interfejs Lattice w oparciu o listy dwukierunkowe.

Interfejs AnnotationLattice rozszerza interfejs Lattice o pojęcia kotwic oraz osi czasowych (por. interfejs Timeline na rycinie 6.2). Metody setAnchor oraz getAnchor umożliwiają ustawienie kotwicy wierzchołka. Metody zawierające wyraz „*timeline*" w nazwie służą do obsługi tablicy obiektów typu Timeline związanych z instancją realizującą AnnotationLattice. Metody isAnchored ułatwiają testowanie zakotwiczenia wierzchołków oraz krawędzi.

Kończąc niniejszy opis implementacji anotacji przedstawimy interfejs Walker oraz klasy z nim powiązane. Metody graph oraz node zwracają graf oraz wierzchołek z którym aktualnie związany jest Walker. Metody firstArc oraz nextArc umożliwiają iterowanie po krawędziach grafu powiązanych z aktualnym wierzchołkiem. Metoda stepThis ustawia jako wierzchołek aktualny Walkera drugi koniec krawędzi zwróconej przy ostatnim wywołaniu metod iterujących. W pakiecie slope.represent.graph zdefiniowano klasy pomocnicze realizujące interfejs Walker, których celem jest ograniczenie zbioru krawędzi zwracanych przy iteracji (użycie wzorca projektowego „dekorator"). Klasy LinkedForwardWalker oraz LinkedBackwardWalker ograniczają zbiory iterowanych krawędzi do wychodzących oraz odpowiednio wchodzących. Klasa SelectiveWalker umożliwia ograniczenie listy iterowanych krawędzi na podstawie zawartości etykiety krawędzi. Interfejs ObjectFilter reprezentuje predykat na etykiecie krawędzi. Z interfejsem ObjectFilter związana jest hierarchia klas (nie przedstawiona na diagramie) umożliwiająca budowanie złożonych wyrażeń filtrujących w algebrze Boole'a (wzorzec projektowy *Composite*).

6.4 Realizacja architektury tablicowej

Na rycinie 6.6 przedstawiono pakiet slope.integrate.blackboard realizujący architekturę tablicową.

Oprócz interfejsów opisanych wcześniej, na diagramie 6.6 występuje interfejs Initializable. Interfejs initializable przeznaczony jest dla obiektów, przed których użyciem konieczne jest zarezerwowanie określonych zasobów. Metoda deinitialize zwalnia zasoby zarezerwowane przez obiekt.

Klasa Blackboard reprezentuje współdzieloną strukturę danych, za pośrednictwem której komunikują się komponenty systemu. W SLOPE przyjmuje się, że współdzielona strutkura danych ma postać grafu anotacyjnego. Klasa Blackboard jest obiektem opakowującym (*wrapper*), który zgłasza obiektom koordynującym (por. klasa Mediator) zmianę stanu kraty anotacyjnej (obiekt powiązany asocjacją impl). Klasa BlackboardTimeline opakowuje powiązane obiekty realizujące interfejs Timeline.

Klasa abstrakcyjna Agent reprezentuje źródło danych w architekturze tablicowej. Agent realizuje interfejsy Named oraz Initializable. Metoda receive przyjmuje komunikaty dotyczące zmiany stanu współdzielonej struktury danych. Rodzaj przyjmowanych komunikatów określany jest za pomocą powiązanego obiektu klasy MessageFilter. Na diagramie 6.7 pokazano hierarchię komunikatów w SLOPE. Metoda receive wywoływana jest synchronicznie, co przekazuje sterowanie obiektowi klasy Agent. Metody registerTo oraz unregisterFrom umożliwiają wykonanie dodatkowych operacji inicjalizacyjnych w momencie budowania środowiska (por. klasa Environment). Metoda broadcast umożliwia przekazanie komunikatu, który nie dotyczy zmian we współdzielonej strukturze danych.

Klasa abstrakcyjna Mediator odpowiada za aktywację agentów oraz dystrybujcę komunikatów między agentami.

Klasa abstrakcyjna Environment agreguje główne elementy architektury tablicowej: zbiór Agentów, Mediator oraz Blackboard. Biblioteki programowe realizowane w architekturze tablicowej tworzą klasy konkretne wydziedziczone z Environment, realizujące interfejs fasadowy (API) danej biblioteki.

6.5 Przeszukiwanie przestrzeni rozwiązań

Pakiet slope.search zawiera algorytmy optymalizacyjne działające na reprezentacjach wiedzy stosowanych w środowisku SLOPE.

Niech będzie dana anotacja kratowa $A = (S, a)$. **Funkcją kosztu** nazywamy dowolną funkcję $f_a :\bowtie S \mapsto \mathbb{R}$. Wartość $f_a(p)$ nazywamy **kosztem ścieżki** p. Przyjmijmy oznaczenie:

$$\bowtie (S, t, u) = \{p \in \bowtie S : p[0] = t \wedge p[|p| - 1] = u\}. \tag{6.1}$$

Ścieżką minimalną anotacji (S, a) nazywamy ścieżkę:

$$p_{\min} = \operatorname*{argmin}_{p \in \bowtie(S, \overleftarrow{S}, \overrightarrow{S})} f_a(p). \tag{6.2}$$

Funkcję f_a definiujemy w oparciu o **funkcję kosztu cząstkowego** $\check{f}_a : \mathbb{S} \times \mathbb{S} \mapsto \mathbb{R}$ następująco:

$$f_a(p) = \check{f}_a(\varnothing, p[0]) \sum_{i=1}^{|p|-1} \check{f}_a(p[i-1], p[i]), \tag{6.3}$$

gdzie $\check{f}_a(\varnothing, p[0])$ jest kosztem cząstkowym rozpoczęcia ścieżki od segmentu $p[0]$. [6]

Proponowany algorytm 6.1 służy do znalezienia ścieżki minimalnej w anotacji kratowej. W algorytmie 6.1 zastosowano programowanie dynamiczne, unikając wielokrotnego obliczania kosztu podścieżek, por. (Cormen i inni 2004, 436-366). Pesymistyczna złożoność algorytmu 6.1 wynosi $O(n)$, gdzie n jest liczbą par segmentów mających wspólną kotwicę. W przypadku typowych anotacji kratowych zachodzi $n \propto |S|$; w wariancie pesymistycznym $n \propto |S|^2$. Należy zaznaczyć, że naiwny algorytm poszukiwania ścieżki minimalnej, w którym następuje iteracja po wszystkich możliwych ścieżkach, ma złożoność obliczeniową $O(|S| \bowtie S)$.

[6] Wyrażenie 6.3 jest sformułowane przez analogię do łańcuchów Markowa pierwszego rzędu.

Algorytm 6.1 Wytyczanie ścieżki minimalnej w anotacji kratowej (środowisko SLOPE).

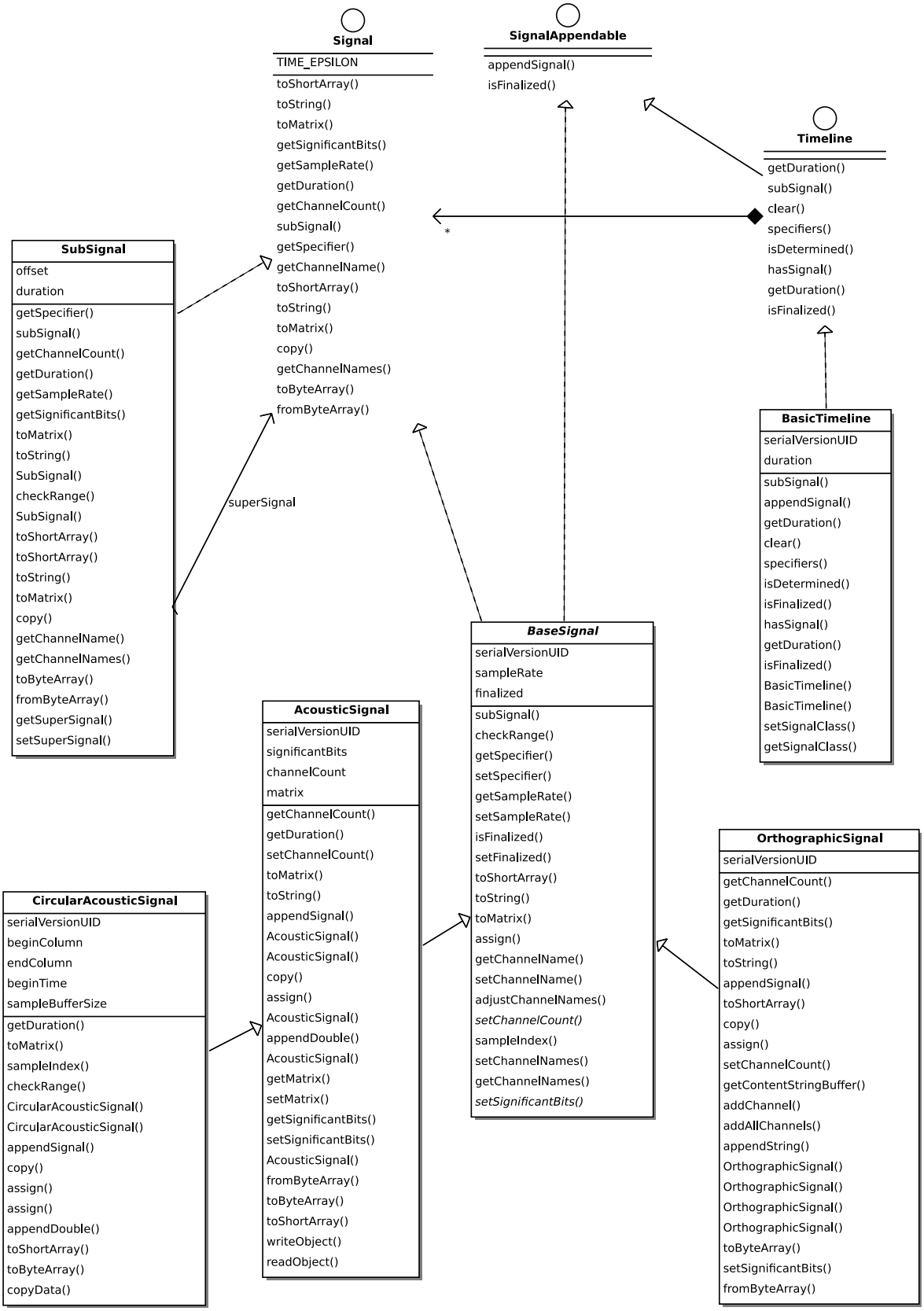
1: minPath(S:Segmentation, $\check{f}_a$): array<int>
Wejście S: segmentacja kratowa
Wejście $\check{f}_a$: funkcja kosztu cząstkowego w anotacji (S, a)
Wyjście tablica identyfikatorów segmentów ścieżki minimalnej
2: queue E
3: array<int> C, B, D
4: {1. Inicjalizacja}
5: **for all** $s \in S : \overleftarrow{s} = \overleftarrow{S}$ **do**
6: enqueue(E, s)
7: $C[\#s] \leftarrow \check{f}(\varnothing, s)$
8: $B[\#s] \leftarrow \varnothing$
9: **end for**
10: {2. Przeszukiwanie}
11: **while** |E|>0 **do**
12: $s \leftarrow$dequeue(E)
13: $C[\#s] \leftarrow \infty$
14: **for all** $s_0 \in S : \overrightarrow{s_0} = \overleftarrow{s}$ **do**
15: $c \leftarrow C[\#s_0] + \check{f}_a(s_0, s)$
16: **if** $c < C[\#s]$ **then**
17: $C[\#s] \leftarrow c$
18: $B[\#s] \leftarrow \#s_0$
19: **end if**
20: **end for**
21: **for all** $s_1 \in S : \overrightarrow{s} = \overleftarrow{s_1}$ **do**
22: $D[\#s_1] \leftarrow D[\#s_1] + 1$
23: **if** $D[\#s_1] = |\{s_0 \in S : \overrightarrow{s_0} = \overleftarrow{s_1}\}|$ **then**
24: enqueue(S, s_1)
25: **end if**
26: **end for**
27: **end while**
28: {3. Odczyt wyniku}
29: $c \leftarrow \infty$
30: **for all** $s \in S : \overrightarrow{s} = \overrightarrow{S}$ **do**
31: **if** $C[\#s] < c$ **then**
32: $c = C[\#s]$
33: $b \leftarrow B[\#s]$
34: **end if**
35: **end for**
36: array<int> R
37: **while** $b \neq \varnothing$ **do**
38: append(R, b)
39: $b \leftarrow B[b]$
40: **end while**
41: **return** reverse(R)

6.6 Konfigurowalność oraz trwałość

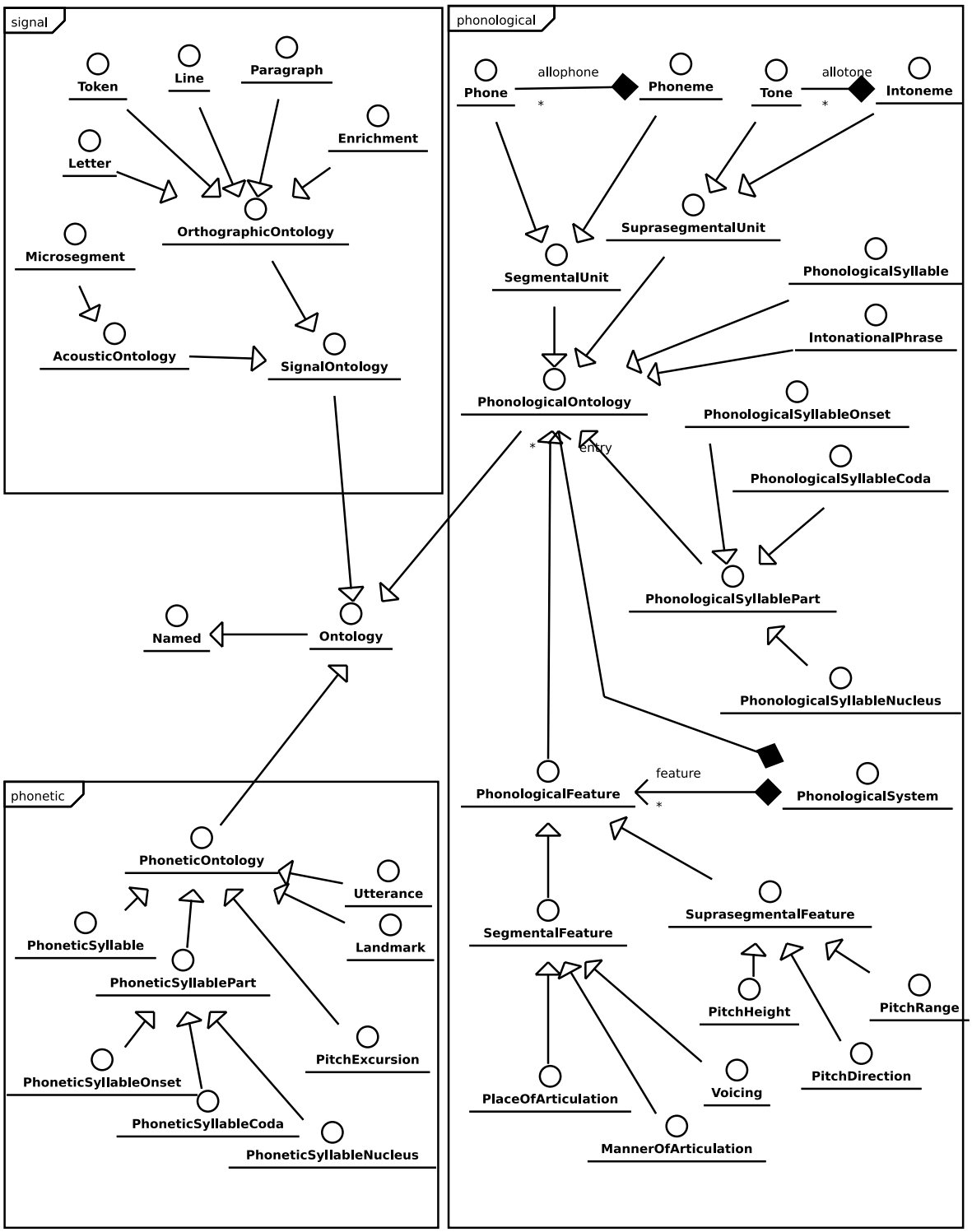
Pakiet slope.configure ułatwia ustalenie parametrów oraz struktury układu bez rekompilacji kodów źródłowych. Pakiet slope.configure oparto na wzorcu projektowym *Dependency Injection* (Fowler 2004). Klasy pakietu slope.configure wnioskują liczbę, rodzaj oraz parametry (konwencja JavaBean) komponentów układu na podstawie pliku tekstowego o składni zbliżonej do linii poleceń programów konsolowych.[7]

Pakiet slope.retrieve wspomaga utrwalanie anotacji oraz sygnałów. W pakiecie slope.store zastosowano obiektowy system zarządzania bazą danych db4o. Dzięki zastosowaniu db4o zamiast relacyjnych systemów zarządzania bazami danych w SLOPE nie występują kłopotliwe w utrzymaniu obiekty DAO (*Data Access Object*) opisujące odpowiedniość obiektowo-relacyjną (*O-R mapping*). Klasy pakietu slope.store stosowane są m.in. do przechowywania słowników oraz anotacji korpusów.

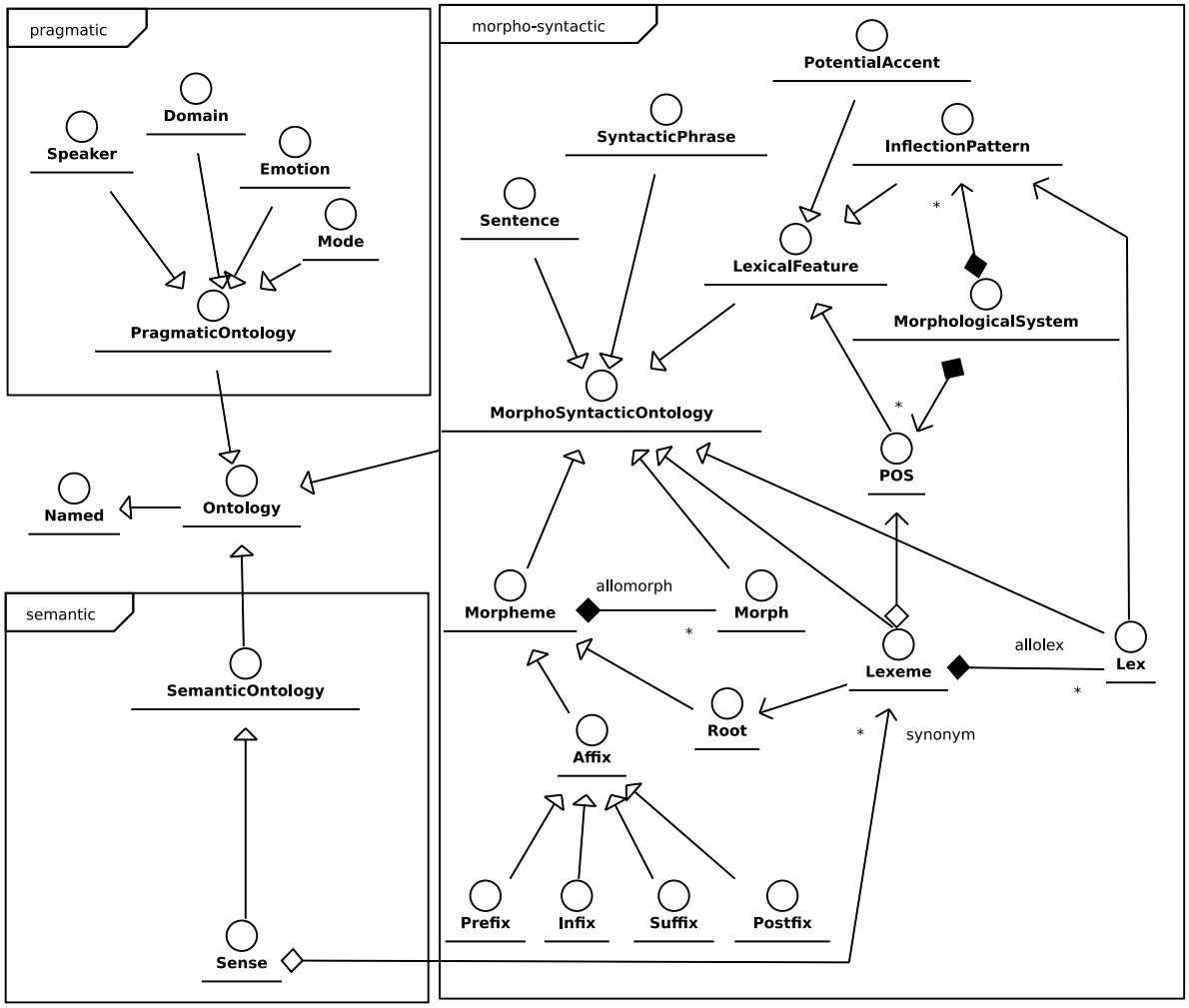
[7] W plikach konfiguracyjnych zrezygnowano z popularnego formatu XML ze względu na niekorzystny stosunek wielkości pliku do reprezentowanej treści.



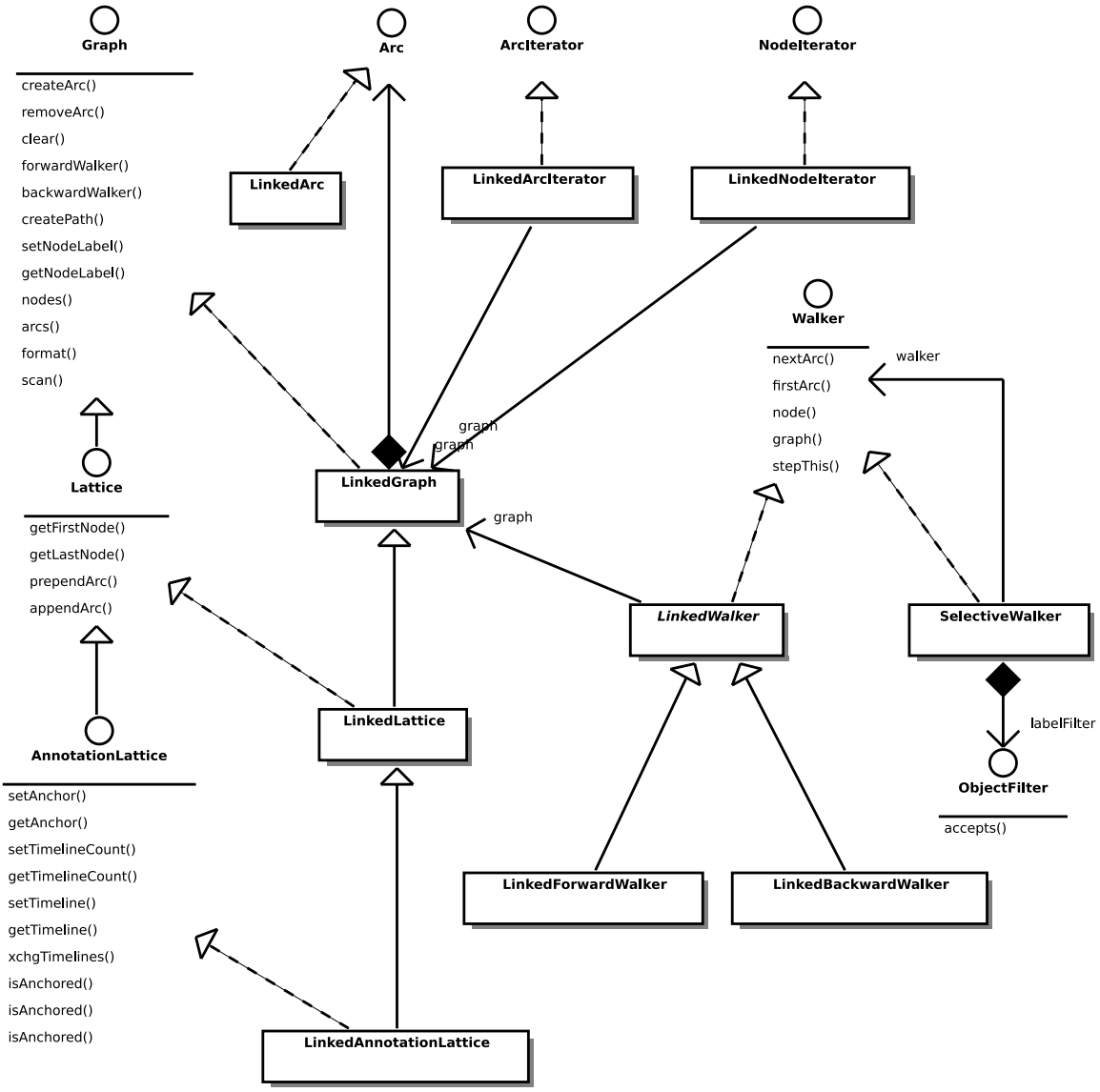
Rycina 6.2: Reprezentacje sygnału w SLOPE. Diagram klas UML.



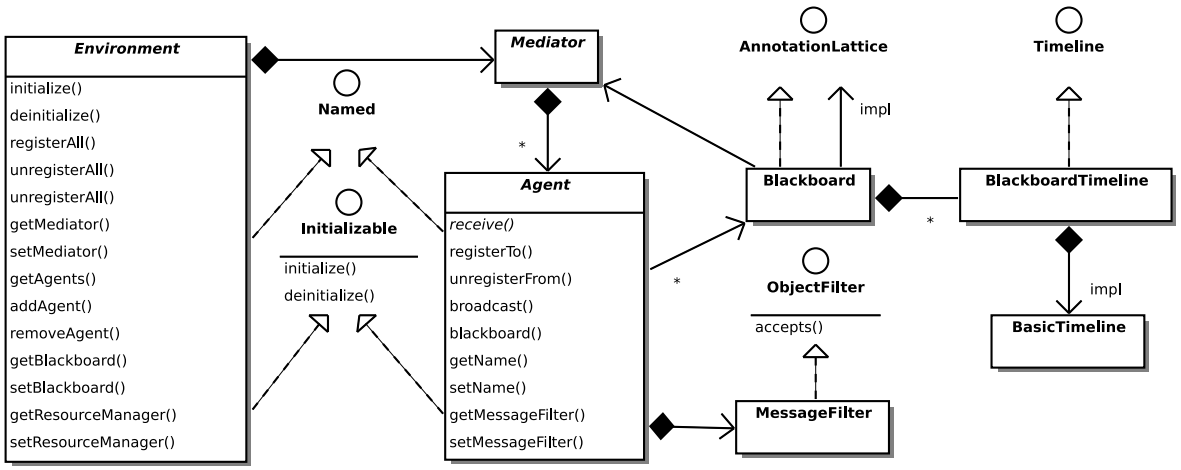
Rycina 6.3: Ontologie niskopoziomowe w SLOPE. Diagram klas UML.



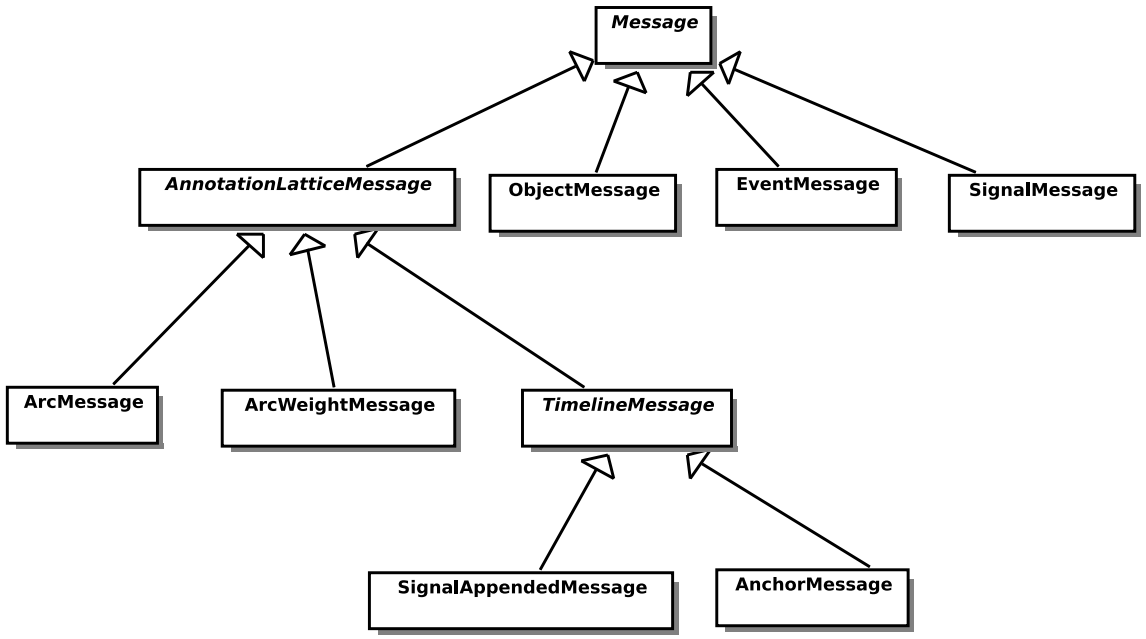
Rycina 6.4: Ontologie wysokopoziomowe w SLOPE. Diagram klas UML.



Rycina 6.5: Grafowa reprezentacja anotacji w SLOPE. Diagram klas UML.



Rycina 6.6: Architektura tablicowa w SLOPE. Diagram klas UML.



Rycina 6.7: Hierarchia komunikatów w SLOPE. Diagram klas UML.

ROZDZIAŁ 7

Układ sygnałowej analizy tonalnej

Analizę wymagań dla układu sygnałowej analizy tonalnej (ekstraktora F_0) wykonano przy założeniu, że zostanie on zastosowany jako podukład tworzonego układu fonologicznej analizy tonalnej. Za pożądane własności ekstraktora F_0 uznano:

1. Skuteczność w szerokim zakresie wariancji tonalnej (m.in. płci oraz mówcy) bez modyfikacji parametrów układu.
2. Możliwość analizy sygnału w czasie rzeczywistym, w szczególności: strumieniowość, umiarkowany koszt obliczeniowy oraz umiarkowane opóźnienie czasowe.

Za nieistotne własności ekstraktora F_0 uznano:

1. Wysoką dokładność pomiaru.
2. Realizację suprasegmentalnej sygnałowej analizy tonalnej (por. 3.3). Przyjęto, że ewentualna korekta błędów ekstrakcji F_0 jest wykonywana w ramach fonetycznej lub fonologicznej analizy tonalnej.

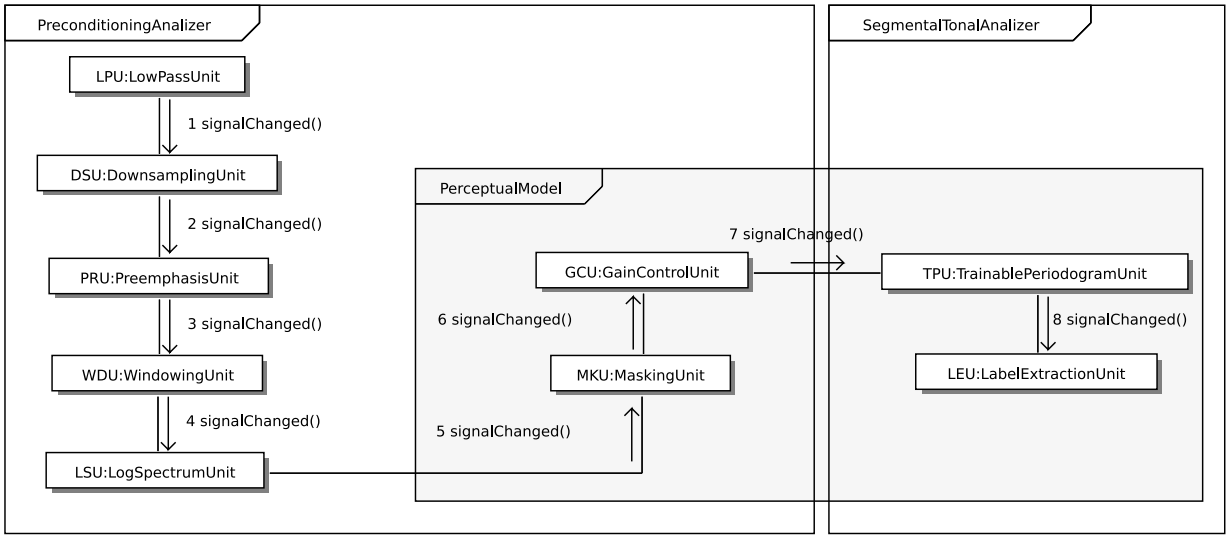
Przyjęto, że etykiety wynikowej sygnałowej anotacji tonalnej powinny zawierać jednopunktowy rozkład F_0 oraz harmoniczność.

W roku 2004 wykonano przegląd ekstraktorów F_0 dostępnych w otwartych aplikacjach/bibliotekach programowych, m.in.: Praat, EST (Edinburgh Speech Tools), SFS (Speech Filing System) oraz Snack. W wyniku przeglądu stwierdzono, że zaden z dostępnych ekstraktorów F_0 nie spełnia wyżej wymienionych wymagań. W związku z powyższym rozpoczęto realizację własnego algorytmu oraz układu ekstrakcji F_0.

W ekstraktorze F_0 zastosowano elementy modelowania psychoakustycznego oraz nowy, uczący się filtr grzebieniowy. W sekcjach 7.1 oraz 7.2 opisano dwa tryby działania układu: tryb analizy oraz tryb uczenia się. W sekcji 7.3 przedstawiono skuteczność ekstraktora F_0.

7.1 Tryb analizy

Na rycinie 7.1 przedstawiono proponowany ekstraktor F_0 w trybie analizy. Zgodnie z ogólnym modelem ekstraktora F_0 (rys. 3.1 na str. 29), na rycinie 7.1 wyodrębniono dwa podukłady: 1) podukład wstępnej analizy sygnałowej (PreconditioningAnalyzer) oraz 2) podukład



Rycina 7.1: Ekstraktor F_0 w trybie analizy. Diagram współpracy UML.

sygnałowej segmentalnej analizy tonalnej (SegmentalTonalAnalyzer). Oprócz tego wyróżniono grupę komponentów opartych na modelach percepcyjnych (PerceptualModel). W kolejnych akapitach opisano algorytmy przetwarzania oraz analizy sygnału implementowane w poszczególnych komponentach na rycinie 7.1. Proponowany ekstraktor F_0 zintegrowano przy użyciu architektury potokowej środowiska SLOPE (por. sekcja 6.2).

Częstotliwość próbkowania oraz rozdzielczość próbki wejściowego cyforwego sygnału w proponowanym ekstraktorze F_0 wynosi odpowiednio 16kHz oraz 32 bity (liczba zmiennopozycyjna pojedynczej precyzji).

Komponent LPU jest filtrem dolnoprzepustowym typu FIR o częstotliwości odcięcia 800Hz. Odpowiedź impulsową LPU uzyskano poprzez okienkowanie oknem Hamminga (N=32) odpowiedzi impulsowej idealnego filtra dolnoprzepustowego.

Komponent DSU jest decymatorem, tj. przetwornikiem sygnału cyfrowego o następującej zależności między sygnałem wejściowym x a sygnałem wyjściowym $DSU(x)$:

$$DSU(x)[n] = x[Mn], \tag{7.1}$$

przy czym w DSU przyjmuje się $M = 8$ (wyjściowa częstotliwość próbkowania wynosi 2kHz).

Komponent PRU wykonuje preemfazę za pomocą filtra FIR rzędu 1 ze współczynnikiem $a = 0.97$.

Komponent WDU tworzy ciąg sygnałów kolejnych segmentów segmentacji ramkowej. W WDU stosuje się okno Hamminga oraz segmentację ramkową z czasem trwania segmentu 64 ms (128 próbek) oraz krokiem 10 ms (20 próbek).

Komponent LSU wyznacza logarytm 64-punktowego widma amplitudowego DFT sygnału wejściowego przy zastosowaniu algorytmu FFT. Dla sygnału wejściowego x sygnał $LSU(x)$ określony jest następująco:

$$LSU_N(x)[0][k] = \ln \left| \sum_{n=0}^{N-1} x_N[n] e^{-i2\pi nk/N} \right|, \tag{7.2}$$

gdzie $N = 128$. Sygnał $LSU_N(x)$ ma jedną próbkę, której wartością jest wektor. Przyjmujemy, że k-ty wymiar wektora $LSU_N(x)[0]$ odnosi się do k tej składowej bazy ortogonalnej transformacji DFT, czyli do funkcji:

$$e^{-i2\pi k/N}. \tag{7.3}$$

Elementy wektora $LSU_N(x)[0]$ nazywamy **składnikami**. Z wartościami składników związana jest skala decybelowa. Okresem próbkowania sygnału $LSU_N(x)$ jest krok segmentacji ramkowej użytej w komponencie WDU. Z symetrii widma amplitudowego sygnałów rzeczywistych wynika, że $LSU_N(x)[0][k] = LSU_N(x)[0][N-k]$, w związku z czym na wyjściu LSU reprezentowane są wartości $LSU(x)$ wyłącznie dla $k \in \{0, 1, \ldots, 63\}$.

Komponent MKU zeruje składniki maskowane oraz składniki nieharmoniczne. Przez składnik nieharmoniczny rozumie się składnik, który nie jest harmoniczną analizowanego sygnału. W celu wykrycia składników do zerowania wprowadzono specjalizowany model psychoakustyczny o niskim koszcie obliczeniowym.

Dla uproszczenia notacji przyjmijmy oznaczenie $X_{MKU} = LSU_N(x)[0]$. Próg maskowania składnika k przez składniki poprzedzające określony jest jako:

$$M^F[k] = \max_{0 \leqslant i < k} \left(X_{MKU}[i] - \alpha - 2(k-i)\delta \right), \tag{7.4}$$

gdzie $\alpha = 5$ oraz $\delta = 0.1$. Obliczenie wektora M^F wykonaywane jest w czasie $O(|M|)$ przy zastosowaniu rekurencji prawostronnej:

$$M^F[i] = \max\left(X_{MKU}[i] - \alpha, M^F[i-1] - \delta \right). \tag{7.5}$$

Próg maskowania składnika k przez składniki następujące określony jest jako:

$$M^B[k] = \max_{k < i < |X_{MKU}|} \left(X_{MKU}[i] - \alpha - 5(i-k)\delta \right). \tag{7.6}$$

(Obliczanie wektora M^B wykonywane jest także w czasie $O(|M|)$.) Wektor maskowany X'_{MKU} określony jest następująco:

$$X'_{MKU}[k] = \begin{cases} X_{MKU}[k] & \text{dla } X_{MKU}[k] > \max\left(M^F[k], M^B[k], X_{MKU}[k-1], X_{MKU}[k+1] \right) \\ 0 & \text{w przec. przyp..} \end{cases} \tag{7.7}$$

Należy zauważyć, że przy obliczaniu X'_{MKU} nie stosuje się pojęcia bezwzględnego progu słyszalności, w związku z czym przedstawiony algorytm maskowania nie wymaga przyjmowania bezwzględnej skali fizycznej dla wartości składników.

Określmy funkcję pomocniczą:

$$E'[k] = \sum_{i \in \{k-2, k-1, k+1, k+2\}} |X'_{MKU}[k]|. \tag{7.8}$$

Jeśli $E'[k] > 0$, to k-ty składnik uznaje się za nieharmoniczny. Ostatecznie sygnał wyjściowy MKU określony jest następująco:

$$MKU(X_{MKU})[0][k] = \begin{cases} X'_{MKU}[k] & \text{dla } E'[k] = 0 \\ 0 & \text{w przec. przyp..} \end{cases} \tag{7.9}$$

Komponent GCU jest uproszczonym modelem tolerancji percepcyjnej dla sygnałów o szerokim zakresie energii. Przyjmijmy oznaczenie $X_{GCU} = MKU(X_{MKU})[0]$. Komponent GCU przekształca wartości składników zgodnie ze wzorem:

$$GCU(X_{GCU})[0][k] = \zeta_{GCU}(X_{GCU}[k]), \tag{7.10}$$

gdzie ζ_{GCU} jest skalą liniową taką, że dla dowolnego $e \in \mathbb{R}$ zachodzi

$$\zeta_{GCU}(e) = \frac{e - \mu(X_{GCU})}{\max(X_{GCU})}, \tag{7.11}$$

przy czym $\mu(X_{GCU})$ oznacza wartość średnią elementów wektora X_{GCU}.

Komponent TPU reprezentuje periodogram obliczany za pomocą jednowarstwowej sieci neuronowej. Przyjmijmy oznaczenie $X_{TPU} = GCU(X_{GCU})[0]$. X_{TPU} jest wektorem kolumnowym rzędu $N = 64$. Określmy X'_{TPU} następująco:

$$X'_{TPU}[k] = \begin{cases} X_{TPU}[k] & \text{dla } k \neq N-1 \\ 0 & \text{w przec. przyp..} \end{cases} \tag{7.12}$$

Przyjmijmy, że dana jest macierz A rzędu $M \times N = 192 \times 64$, w której i-ty wiersz reprezentuje wzorcową wartość transponowanego wektora X'_{TPU} dla sygnału mowy o ustalonej częstotliwości podstawowej f_i. (W sekcji 7.2 przedstawiony jest algorytm wyznaczania macierzy A.) Określmy funkcję unipolarną sigmoidalną $f_u : \mathbb{R} \mapsto \mathbb{R}$ z parametrem κ:

$$f_u(v) = \frac{1}{1 + e^{-\kappa v}}. \tag{7.13}$$

Proponowany periodogram ma postać:

$$TPU(X_{TPU})[n] = f_u((AX'_{TPU})[n]). \tag{7.14}$$

Wzór 7.14 opisuje jednowarstwową sieć neuronową (por. np. Tadeusiewicz 2000).

Komponent LEU wyznacza F_0 oraz harmoniczność segmentu na podstawie periodogramu. Dla danego periodogramu X_{LEU} sygnał wyjściowy komponentu LEU jest określony następująco:

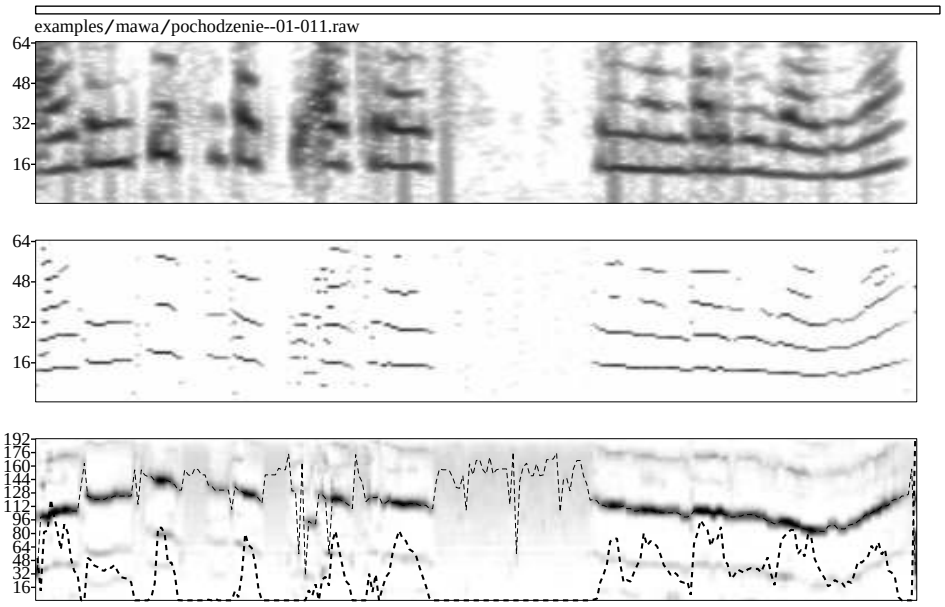
$$LEU(X_{LEU})[0][0] = f_0^{min} + \frac{f_0^{max} - f_0^{min}}{192} \operatorname*{argmax}_i X_{LEU}[i] \tag{7.15}$$

oraz

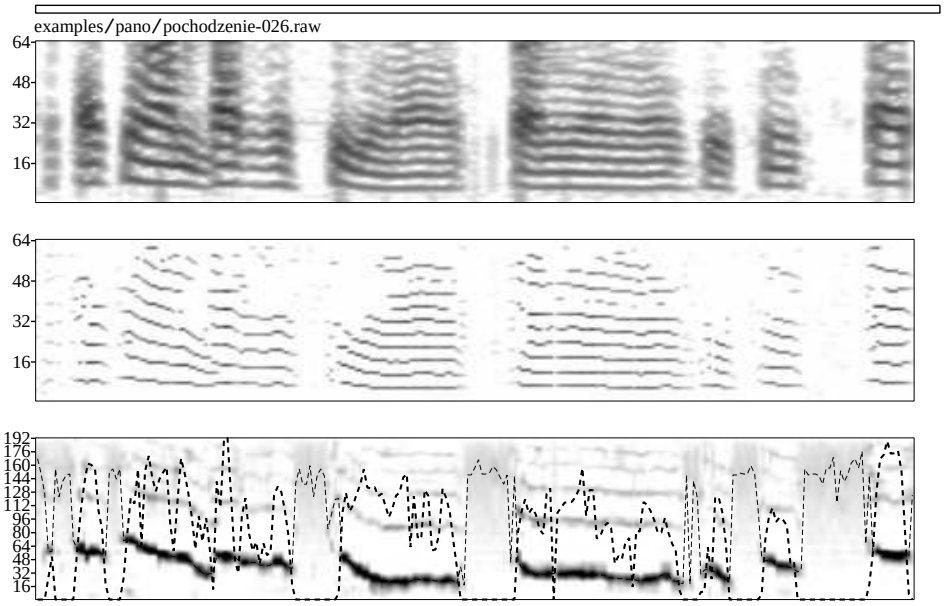
$$LEU(X_{LEU})[0][1] = \max(0, \max_i(X_{LEU}[i]) - \min_i(X_{LEU}[i]) - \gamma), \tag{7.16}$$

gdzie $\gamma = \frac{1}{5}$. Wartości F_0 zawarte są w zakresie $[f_{min}; f_{max})$, natomiast rodzaj skali (np. liniowa, logarytmiczna) określany jest przez zawartość macierzy A (por. równość 7.14).

Ryciny 7.2 oraz 7.3 przedstawiają sygnały wyjściowe komponentów LSU, MKU oraz TPU+LEU (LEU nałożony na TPU, gdzie linia przerywana cieńsza oznacza F_0 a linia przerywana grubsza oznacza harmoniczność) dla dwóch przykładowych fraz intonacyjnych. Użyto frazy intonacyjne z korpusu PoInt (Karpiński 2002). Transkrypcja ortograficzna sygnału z ryciny 7.2: «Ale ten pierwszy taki okres wspominam bardzo dobrze.». Transkrypcja ortograficzna sygnału z ryciny 7.3: «To takie mam bardzo silne wspomnienie z dzieciństwa.».



Rycina 7.2: Przebieg sygnałów na wyjściu komponentów (kolejno od góry): LSU, MKU oraz TPU/LEU. Fraza intonacyjna: mawa/pochodzenie--01-011, mówca: „mawa" (kobieta).

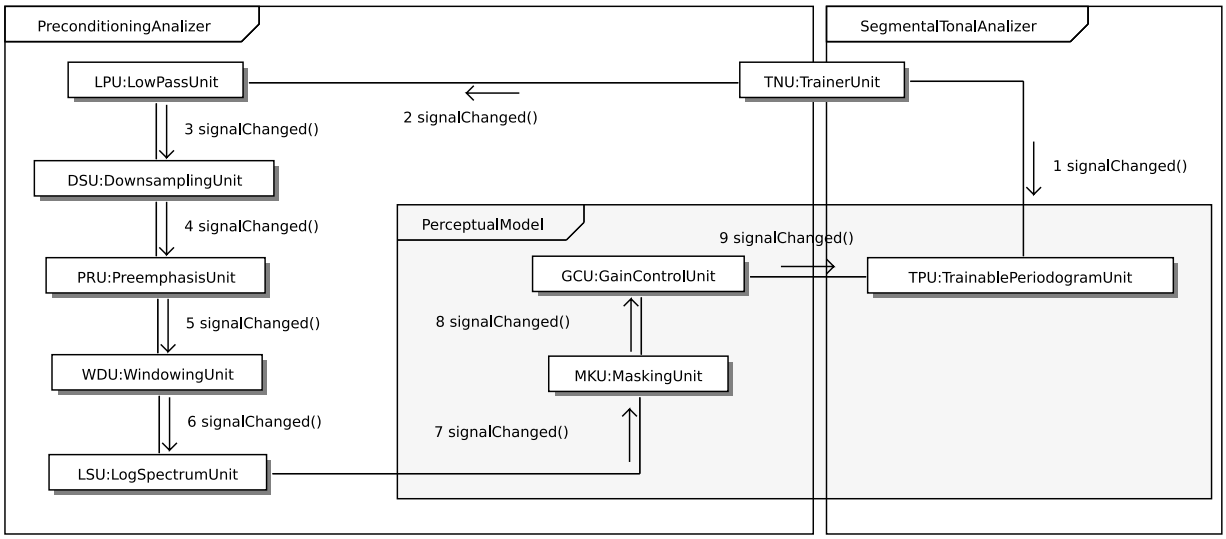


Rycina 7.3: Przebieg sygnałów na wyjściu komponentów (kolejno od góry): LSU, MKU oraz TPU/LEU. Fraza intonacyjna: pano/pochodzenie-026, mówca: „pano" (mężczyzna).

7.2 Tryb uczenia

Macierz A występującą w równaniu 7.14 na stronie 104 będziemy nazywać **macierzą periodogramu**. Macierz periodogramu jest wyznaczana przez układ ekstrakcji F_0 w trybie uczenia. Proponowany algorytm wyznaczania macierzy periodogramu opiera się na uczeniu pod nadzorem z zastosowaniem gradientowego algorytmu optymalizacji.

Rycina 7.4 przedstawia układ ekstrakcji F_0 w trybie uczenia. Funkcje większości komponentów występujących na rycinie 7.4 opisano w sekcji 7.1. Układ ekstrakcji F_0 w trybie uczenia zintegrowano przy użyciu architektury potokowej SLOPE (por. sekcja 6.2). W porównaniu



Rycina 7.4: Ekstraktor F_0 w trybie uczenia. Diagram współpracy UML.

do trybu analizy, w trybie uczenia: 1) jest dodatkowy komponent TNU (zbiór uczący), 2) brak komponentu LEU, 3) komponenty TPU działa w trybie uczenia się. Przez zbiór uczący rozumie się tutaj zbiór $\{(x_0, y_0), (x_1, y_1), \ldots, (x_N, y_N)\}$ taki, że jeśli na wejściu LPU dany jest sygnał x_i, to y_i jest pożądanym sygnałem na wyjściu TPU.

Komponent TNU wykonuje N iteracji takich, że w i-tej iteracji do obiektów LPU oraz TPU przekazywane są odpowiednio sygnały x_i oraz y_i. Przyjmijmy, że:

$$\ln f_i = \ln f_{\min} + \frac{i}{N}(\ln f_{\max} - \ln f_{\min}), \tag{7.17}$$

gdzie $N = 192$, $f_{\min} = 64$[Hz], $f_{\max} = 512$[Hz]. Sygnał x_i określony jest jako:

$$x_i[n] = \sum_{j=1}^{M_j} j^{\alpha} \cos\left(j \frac{f_i}{f}\right), \tag{7.18}$$

gdzie $\alpha \in [-0.15; -0.35]$ oraz $f = 16000$ jest częstotliwością próbkowania. Jak wynika ze wzoru 7.18 sygnał x_i jest tonem złożonym o częstotliwości podstawowej f_i oraz spadku amplitudy widma zależnym od α. Sygnał y_i określony jest jako:

$$y_i[n] = \beta + (1 - \beta) exp(\gamma |f_n - f_i|), \tag{7.19}$$

gdzie $\beta \in [0.45; 0.60]$ oraz $\gamma \in [0.18; 0.62]$. Wzory 7.18 oraz 7.19 zostały dobrane eksperymentalnie w celu osiągnięcia jak najszybszej zbieżności macierzy periodogramu.

Komponent TPU, jako element układu sygnałowej analizy tonalnej został opisany na stronie 104. W trybie uczenia komponent TPU aktywuje dodatkowe wejście sygnałowe przyjmujące pożądaną wartość sygnału wyjściowego (periodogramu), którą będziemy oznaczać przez y_i.

Oznaczmy przez X'_i sygnał X'_{TPU} (por. wzór 7.12 na stronie 104) w sytuacji gdy na wejściu LPU dany jest sygnał x_i. Oznaczmy przez y'_i sygnał:

$$y'_i[n] = f_u((AX'_i)[n]), \tag{7.20}$$

gdzie A jest macierzą periodogramu oraz f_u jest funkcją sigmoidalną unipolarną (por. wzór 7.13 na stronie 104). Oznaczmy przez $\mathcal{V}$ zbiór wszystkich macierzy rzeczywistych rzędu $N \times M = 192 \times 64$. Dla ustalonego zbioru uczącego (x_i, y_i), $i = \{0, 1, \ldots, N-1\}$ definiujemy funkcję celu $E : \mathcal{V} \mapsto \mathbb{R}$ następująco:

$$E(A) = \frac{1}{2} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (y_i'[j] - y_i[j])^2, \tag{7.21}$$

por. Osowski (1996, 39). Problem wyznaczenia macierzy periodogramu utożsamiamy z problemem znalezienia $\widehat{A} \in \mathcal{V}$ minimalizującego funkcję celu, tj.:

$$\widehat{A} = \underset{A \in \mathcal{V}}{\operatorname{argmin}}\, E(A). \tag{7.22}$$

W celu wyznaczenia macierzy $\widehat{A}$ stosowany jest algorytm największego spadku, który opisał np. Osowski (1996, 54). Określmy macierz gradientów funkcji E następująco:

$$\nabla E = \begin{bmatrix} \frac{\partial E}{\partial A[0][0]} & \frac{\partial E}{\partial A[0][1]} & \cdots & \frac{\partial E}{\partial A[0][N-1]} \\ \frac{\partial E}{\partial A[1][0]} & \frac{\partial E}{\partial A[1][1]} & \cdots & \frac{\partial E}{\partial A[1][N-1]} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial E}{\partial A[M-1][0]} & \frac{\partial E}{\partial A[M-1][1]} & \cdots & \frac{\partial E}{\partial A[M-1][N-1]} \end{bmatrix}, \tag{7.23}$$

gdzie $\frac{\partial E}{\partial A[i][j]}$ jest pochodną funkcji E względem $A[i][j]$.

Pochodna funkcji sigmoidalnej unipolarnej f_u ma postać:

$$\frac{df_u(v)}{dv} = \kappa f_u(v)(1 - f_u(v)). \tag{7.24}$$

Uwzględniając równość 7.24 dostajemy:

$$\frac{\partial E}{\partial A[i][j]} = 2\kappa(y_i'[j] - y_i[j])y_i'[j](1 - y_i'[j])X_i'[0][j]. \tag{7.25}$$

Zgodnie z algorytmem największego spadku minimalizację funkcji E przeprowadzamy w kierunku $-\nabla E(A)$. Iteracyjny proces minimalizacji rozpoczyna się inicjalizacją macierzy $\widehat{A}_0$ wartościami zerowymi. Kolejne przybliżenia macierzy $\widehat{A}$ wyznaczane są na podstawie wzoru:

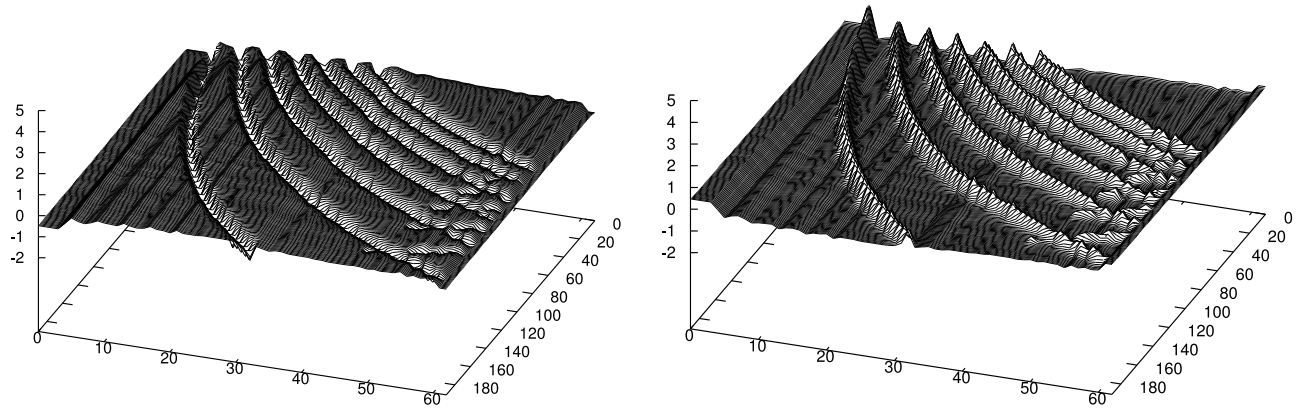
$$\widehat{A}_{i+1} = \widehat{A}_i - \eta \nabla E(\widehat{A}_i), \tag{7.26}$$

gdzie parametr $\eta \in \mathbb{R}$ jest współczynnikiem uczenia. Za współczynnik uczenia w i-tej iteracji przyjmujemy:

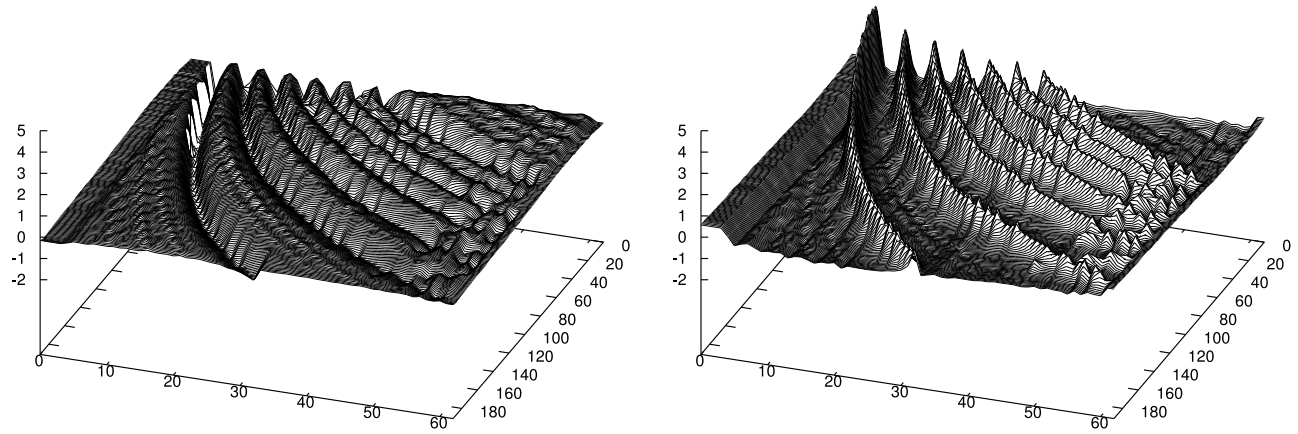
$$\eta = \frac{2}{i+2}. \tag{7.27}$$

Wykonujemy 6 iteracji.

Na rycinach 7.5, 7.6 oraz 7.7 przedstawiono wykresy gradientów oraz macierzy periodogramu w wybranych iteracjach procesu minimalizacji.



Rycina 7.5: Uczenie periodogramu w iteracji pierwszej. Gradient $\nabla E(\widehat{A}_0)$ (strona lewa) oraz macierz $\widehat{A}_1$ (strona prawa). Współczynnik $\eta = 1$.



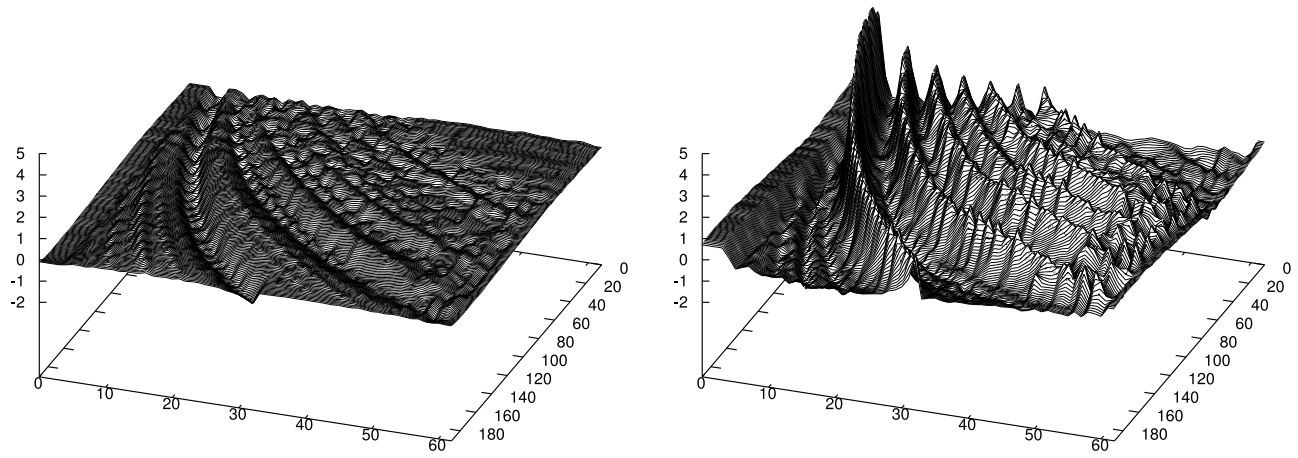
Rycina 7.6: Uczenie periodogramu w iteracji drugiej. Gradient $\nabla E(\widehat{A}_1)$ (strona lewa) oraz macierz $\widehat{A}_2$ (strona prawa). Współczynnik $\eta = 2/3$.

7.3 Uczenie i wyniki

Pomiar skuteczności proponowanego ekstraktora F_0 wykonano w oparciu o *Keele Pitch Database* (Plante i inni 1995). Keele Pitch Database (dalej jako KPD) jest ogólnodostępnym, otwartym zbiorem sygnałów oraz wzorcowych anotacji sygnałowych, przeznaczonym do pomiaru skuteczności ekstraktorów F_0. W KPD dla każdego sygnału dana jest anotacja sygnałowa, której etykiety (zweryfikowane subiektywnie) określają F_0 oraz harmoniczność segmentów o czasie trwania 25.6ms. W skład KPD wchodzi korpus mowy oraz zestaw nagrań stosowanych w eksperymentach psychoakustycznych, które opracowali Meddis i Hewitt (1991). Sygnały mowy w ramach KPD są rejestracją (użyto mikrofonu oraz elektroglotografu) 15-tu realizacji zrównoważonego fonologicznie tekstu „The North Wind Story" przez 15-tu mówców będących rodowitymi użytkownikami języka angielskiego (równa liczba głosów męskich, kobiecych i dziecięcych).

W tabelach 7.1 oraz 7.2 przedstawiono wartość uśrednioną błędu grubego $d^{0,2}_{\text{gross}}$ (por. równanie 3.5 na stronie 30) wybranych ekstraktorów F_0, w tym proponowanego (wyjście komponentu TPU). Wyniki pozostałych ekstraktorów F_0 wzięto z pracy Sun (2002b). Oznaczenia użyte w tabelach 7.1 oraz 7.2:

1. PDA: układ ekstrakcji F_0 należący do pakietu EST (Edinburgh Speech Tool Library),
2. GETF0: układ ekstrakcji F_0 należący do pakietu ESPS (Entropic Signal Processing



Rycina 7.7: Uczenie periodogramu w iteracji szóstej. Gradient $\nabla E(\widehat{A}_5)$ (strona lewa) oraz macierz $\widehat{A}_6$ (strona prawa). Współczynnik $\eta = 1/4$.

Tabela 7.1: Odsetek grubych błędów ekstrakcji F_0 w korpusie Keele. Głosy męskie.

Nazwa układu	M1	M2	M3	M4	M5	Średnia
PDA	5.17	10.22	3.40	3.16	5.15	5.42
GETF0	1.49	11.36	2.74	2.59	1.59	3.95
To Pitch (ac)	3.36	8.32	1.30	2.96	1.59	3.30
SHRP	4.29	4.49	0.41	0.55	0.68	2.08
TPU	3.34	3.67	2.98	2.17	2.21	2.87

System),

3. To Pitch (ac): funkcja ekstrakcji F_0 dostępna w programie Praat,
4. SHRP: układ ekstrakcji F_0 proponowany w pracy Sun (2002b).

Nazwy M1–M5 oraz F1–F5 używane w nagłówkach kolumn tabel odnoszą się do głosów męskich oraz kobiecych o identyfikatorach odpowiednio od 1 do 5. Na korpusie Keele proponowany układ ekstrakcji wykazuje skuteczność wyższą od układów PDA oraz GETF0 oraz zbliżoną do układu PRAAT. Na szczególną uwagę zasługuje niska wariancja błędu ze względu na mówcę. Oczekuje się, że istotny wzrost skuteczności nastąpi w wyniku wykorzystania możliwości adaptacyjnych proponowanego układu ekstrakcji F_0.

Testy skuteczności przedstawione w tabelach 7.1 oraz 7.2 wykonano na komputerze z procesorem Intel Core 2 Duo T7500 2.2GHz pracującym pod kontrolą systemu Linux (jądro

Tabela 7.2: Odsetek grubych błędów ekstrakcji F_0 w korpusie Keele. Głosy kobiece.

Nazwa układu	K1	K2	K3	K4	K5	Średnia
PDA	7.28	4.97	4.22	14.06	4.48	7.00
GETF0	11.23	6.15	6.62	7.15	2.74	6.78
To Pitch (ac)	4.31	2.21	2.98	4.66	1.08	2.99
SHRP	2.22	1.63	1.66	2.61	0.59	1.74
TPU	3.07	2.36	2.27	3.24	2.03	2.58

2.6.25.14, kompilacja standardowa dla procesorów i686). W testach wykorzystywano pojedynczy rdzeń procesora. Podczas testów średni czas wykonania iteracji uczącej wyniósł 163ms. Średni czas analizy jednej sekundy sygnału o częstotliwości próbkowania 16kHz oraz rozdzielczości 16-tu bitów wyniósł 94ms.

Cechami wspólnymi proponowanego układu ekstrakcji F_0 oraz układów ekstrakcji F_0 opisanych w literaturze są m.in.:

- wstępna analiza sygnałowa obejmująca maskowanie tonalne (por. model Terhardta),
- wstępna analiza sygnałowa obejmująca automatyczną kontrolę wzmocenienia (por. model Lyona),
- periodogram oparty na rozpoznawaniu wzorców w dziedzinie częstotliwościowej (por. periodogramy harmoniczne oraz grzebieniowe),
- założenie, że percepcja wysokości tonu jest zdolnością wyuczoną w wyniku odbioru sygnałów o wysokiej harmoniczności (por. hipoteza Terhardta),

Cechami wyróżniającymi proponowany układ ekstrakcji F_0 w zestawieniu z układami ekstrakcji F_0 opisanymi w literaturze są m.in.:

- periodogram uczący się,
- periodogram oparty na rozpoznawaniu wzorców metodą dyskryminacyjną,
- relatywnie niska, 192-punktowa rozdzielczość dyskretnej skali częstotliwości,
- wysoka wydajność (w klasie algorytmów percepcyjnych).

Głównym tematem dalszych prac badawczych będzie adaptacja macierzy periodogramu na bieżąco podczas analizy sygnału mowy. Wzorcowa trajektoria F_0 (uczenie pod nadzorem) otrzymana zostanie jako sprzężenie zwrotne z wyższych poziomów analizy tonalnej. Oprócz tego planowana jest ewaluacja nieliniowych, heurystycznych (percepcyjnych) algorytmów wstępnej analizy sygnałowej.

ROZDZIAŁ 8

Układ fonetycznej analizy tonalnej

Na etapie projektowania proponowanego układu oceniono własności fonetycznych anotacji tonalnych w kontekście zastosowań w fonologicznej analizie tonalnej. Za pożądane własności anotacji fonetycznej uznano:

1. zastosowanie segmentacji sylabicznej,
2. jednorodną formę etykiet (równa liczba zmiennych skalarnych),
3. niską zawartość cech pozajęzykowych.

Jednocześnie mniejszą wagę przypisuje się zagadnieniom:

1. odwracalności (por. z uwagą o syntezie intonacyjnej na s. 57),
2. korelacji zmiennych w etykietach (przyjmujemy, że poza trywialnymi przypadkami korelacji wynikającymi z zależności funkcjonalnych, odrzucanie wymiarów na podstawie częściowej korelacji byłoby przedwczesne).

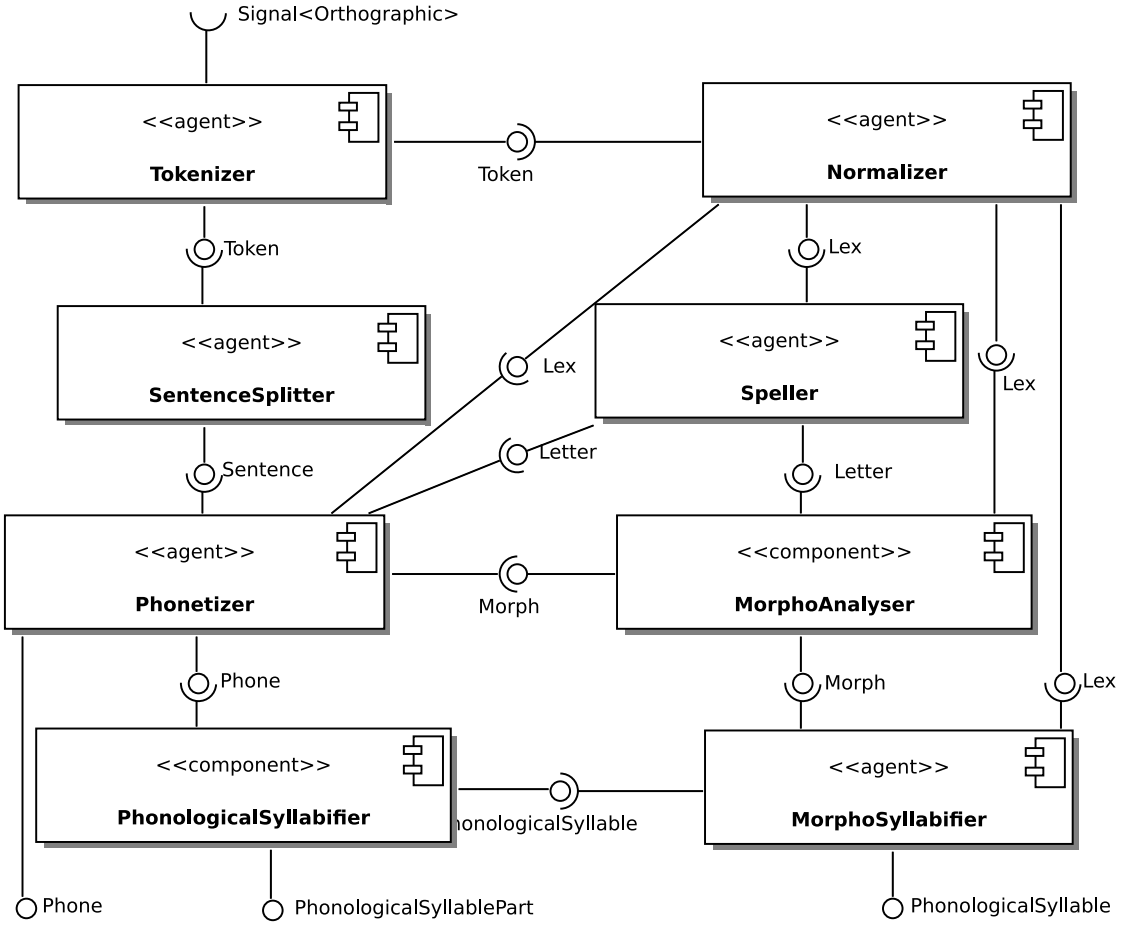
Za istotne własności algorytmu analizy (układu) uznano odporność na lokalne błędy ekstrakcji F_0, strumieniowość oraz wydajność.

Danymi wejściowymi proponowanego układu fonetycznej analizy tonalnej jest trójka uporządkowana: 1) sygnał akustyczny, 2) sygnał ortograficzny oraz 3) napisowy identyfikator mówcy. Przyjmuje się, że oba sygnały wejściowe są zapisem tej samej wypowiedzi. Sygnał ortograficzny pełni funkcję wspomagającą przy określaniu segmentacji. Identyfikator mówcy jest stosowany przy standaryzacji etykiet anotacji wyjściowej (opis w sekcji 8.2.5).

W prezentowanym układzie zastosowano algorytmy oparte na wiedzy, zintegrowane w środowisku SLOPE z zastosowaniem architektury tablicowej (por. sekcja 6.4).

8.1 Podukład analizy ortograficznej

Na rycinie 8.1 przedstawiono diagram komponentów UML podukładu analizy ortograficznej. Przedstawione komponenty zintegrowano za pomocą SLOPE. Dla zwiększenia czytelności na diagramie oznaczono ontologie systemu SLOPE jako interfejsy komponentów. Powyższa



Rycina 8.1: Podukład analizy ortograficznej w układzie fonetycznej analizy tonalnej. Diagram komponentów UML.

konwencja stosowana jest w dalszej części pracy. Należy pamiętać, że w architekturze tablicowej SLOPE interfejsy komponentów (agentów) są jednorodne a kierunki przepływu danych między komponentami wynikają wyłącznie z rodzaju modyfikacji dokonywanych przez komponenty na współdzielonej anotacji. W kolejnych podsekcjach opisano komponenty przedstawione na rycinie 8.1.

8.1.1 Tokenizer

W niejszej pracy **tokenem** nazywamy maksymalnie długi (dopasowanie zachłanne) podciąg znaków sygnału ortograficznego pasujący do wyrażenia regularnego:

```
/(^\s*|\s+)\S+(\s*$)?/.
```
(8.1)

Zgodnie z proponowaną definicją dowolny sygnał ortograficzny można podzielić jednoznacznie na tokeny, których konkatenacja da wejściowy sygnał ortograficzny. W przypadku rozważa-

nych w niniejszej pracy języków naturalnych token obejmuje zwykle wystąpienie dokładnie jednego leksu.

Agent Tokenizer wyznacza lokalizacje tokenów w wejściowym sygnale ortograficznym. Tokenizer wstawia do anotacji segmenty etykietowane ontologiami klasy Token, zakotwiczone na osi czasowej wejściowego sygnału ortograficznego. Nazwą wstawianego Tokenu jest podciąg znaków sygnału ortograficznego dopasowany do wyrażenia 8.1. Tokenizer zaimplementowano w oparciu o standardową bibliotekę wyrażeń regularnych w języku Java.

8.1.2 Normalizer

Agent Normalizer tworzy ścieżkę leksów (etykieta Lex) dla każdego otrzymanego segmentu tokenu (etykieta Token). Agent Normalizer jest oparty na skrajnie uproszczonym, zdefiniowanym algorytmicznie leksykonie, do którego należą wszystkie ciągi małych liter rozszerzonego alfabetu łacińskiego. Przyjmuje się, że leksy mogą wystąpić w tokenach z opcjonalną kapitalizacją (np. kapitalizacja pierwszej litery w zdaniu). Znaki nieliterowe w tokenach są ignorowane, w szczególności zaś ścieżka leksów tokenu może być pusta.

W aktualnej implementacji Normalizera nie uwzględnia się szeregu zagadnień normalizacji tekstu polskiego takich, jak liczby, daty i skróty. Zagadnienia te podjęte zostały m.in. w pracy Jassem i inni (2006).

8.1.3 SentenceSplitter

Agent SentenceSplitter tworzy warstwę zdań (etykieta Sentence) dla danej warstwy tokenów (etykieta Token). Tworzona warstwa zdań jest oparta na kotwicach warstwy tokenów.

SentenceSplitter został zaimplementowany w oparciu o narzędzie GNU flex dostępne pod adresem http://flex.sourceforge.net. Wydruk 8.1 przedstawia uproszczoną wersję reguł podziału zdaniowego dla języka polskiego (składnia języka flex). Reguły te zawierają listę skrótów zakończonych kropką, które nie wystąpują bezpośrednio przed granicą zdaniową. Na podstawie reguł, program flex generuje kod źródłowy w języku C. Wygenerowany kod wstawia znak następnej linii w miejscu granicy zdaniowej.

Agent SentenceSplitter: 1) konkatenuje nazwy tokenów wejściowych (rozdzielając tokeny spacjami), 2) przekazuje wynik konkatenacji do kodu wygenerowanego programem flex a następnie 3) tworzy warstwę zdań na podstawie lokalizacji znaków następnej linii w strumieniu tekstowym wychodzącym z kodu wygenerowanego programem flex.

Listing 8.1: Uproszczone reguły podziału tekstu polskiego na zdania (język flex).

```
ABBREV  ([Aa]b|[Dd]s|[Dd]yr|[Kk]ier|[Pp]rof|[Tt]zw)

%s on

%%

[ ]{ABBREV}"."          ECHO; BEGIN(INITIAL);
[.!?]                   ECHO; BEGIN(on);
<on>[ ]*[[:upper:]]     putc('\n'); BEGIN(INITIAL); REJECT;
.                       ECHO; BEGIN(INITIAL);
```

8.1.4 Speller

Agent Speller dla każdego segmentu leksu tworzy ścieżkę liter (etykieta Letter) występujących w nazwie leksu.

8.1.5 Phonetizer

Agent Phonetizer tworzy warstwę głosek (etykieta Phone) na podstawie wejściowej warstwy liter (etykieta Letter), przy uwzględnieniu lokalizacji segmentów morfów (etykieta Morph), leksów (etykieta Lex) oraz zdań (etykieta Sentence). Tworzona warstwa głosek jest oparta na kotwicach warstwy liter.

Steffen-Batogowa (1975) oraz Steffen-Batóg i Nowakowski (1992) przedstawili formalne reguły transkrypcji fonematycznej oraz fonetycznej polskiego tekstu ortograficznego. Wypych (1999) zaproponował układ transkrypcji fonematycznej polskiego tekstu ortograficznego oparty na automatach skończenie-stanowych (FSA, *Finite State Automaton*).

Wypych i inni (2003) przedstawili rozszerzoną wersję układu Wypych (1999), w której zastosowano przetworniki skończenie-stanowe (FST, *Finite State Transducer*), reguły transkrypcji fonetycznej Batogowej oraz słownik wyjątków. W regułach wykorzystano także wyniki prac, które opublikowali Madejowa (1989) oraz Nowak (1991). W regułach uwzględniono dwa warianty regionalne wymowy polskiej: północno-wschodni oraz południowo-zachodni. Ponadto Wypych i inni (2003) przeprowadzili ewaluację reguł transkrypcji fonetycznej, w wyniku której wykonano dalsze rozszerzenia reguł oraz przygotowano zbiór ponad tysiąca leksemów o nieregularnej transkrypcji. Ewaluację wykonano na korpusie tekstów ortograficznych zawierającym ponad 20 tyś. leksów, w tym fragmenty Słownika Wyrazów Obcych, który opublikował Sobol (2002).[1]

W latach 2004-2006 Wypych kierował pracą zespołu lingwistów[2], w wyniku której zweryfikowano transkrypcje fonetyczne ponad 270 tyś. leksów polskich. Najliczniejszą grupę spośród zweryfikowanych leksów stanowiły nazwy własne (blisko 200 tyś.), których weryfikację oparto m.in. na pracy Łobacz (1999). W wyniku weryfikacji wygenerowano zbiór ponad 10 tyś. wyrażeń regularnych opisujących wyjątki od reguł transkrypcji.

Stosowany w agencie Phonetizer formalizm opisu reguł transkrypcji fonetycznej wywodzi się z formalizmu, który zaproponowała Steffen-Batogowa (1975). Ciągi znaków oznaczamy małymi literami greckimi. Literę ϵ rezerwujemy dla ciągu pustego (tj. ciągu o zerowej liczbie znaków). Przez $\alpha\beta$ oznaczamy ciąg powstający przez konkatenację ciągów α oraz β. Zbiory ciągów znaków będziemy oznaczać wielkimi literami alfabetu łacińskiego. Przez X^* oznaczamy domknięcie Kleene'ego zbioru X, tj. zbiór wszystkich skończonych ciągów znaków ze zbioru X (włączając ϵ).

Oznaczmy przez $\mathfrak{L}$ zbiór złożony z liter oraz znaku technicznego # (granica leksu lub zdania). Oznaczmy przez $\mathcal{G}$ zbiór głosek. **Tabelą transkrypcji** $T[1..m][1..n]$ nazywamy macierz $m \times n$, spełniającą następujące warunki:

[1] Korpus zebrała oraz ewaluację wykonała Emilia Szalkowska z Instytutu Językoznawstwa UAM.

[2] Weryfikacją transkrypcji fonetycznych zajmowali się: Emilia Szalkowska z Instytutu Językoznawstwa UAM, Michał Szczyszek z Instytutu Filologii Polskiej UAM oraz Karol Świetlik z wydawnictwa Lektor-Klett w Poznaniu.

1.
$$T[1][1] \in \mathcal{L}, \tag{8.2}$$

2.
$$T[i][1] \subset \mathcal{L}^* \text{dla } 2 \leqslant i \leqslant m, \tag{8.3}$$

3.
$$T[1][j] \subset \mathcal{L}^* \text{dla } 2 \leqslant j \leqslant n, \tag{8.4}$$

4.
$$T[i][j] \subset \mathcal{G}^* \text{dla } 2 \leq i \leqslant m \wedge 2 \leqslant j \leqslant n. \tag{8.5}$$

Niech będzie ustalona tabela transkrypcji $T[1..m][1..n]$. Dowolna para (i, j) taka, że $2 \leqslant i \leqslant m$ oraz $2 \leqslant j \leqslant n$, reprezentuje **regułę transkrypcji**:

$$T[1][1] \rightarrow T[i][j]/T[i][1]\_T[1][j]. \tag{8.6}$$

Reguła transkrypcji 8.6 stanowi, że wystąpienie litery $T[1][1]$ w kontekście $\alpha T[1][1]\beta$, gdzie $\alpha \in T[i][1]$ oraz $\beta \in T[1][j]$, należy transkrybować jako $T[i][j]$. W odróżnieniu od prac Steffen-Batogowej, w bieżącej pracy nie wymaga się by w każdym kontekście było możliwe zastosowanie co najwyżej jednej reguły. W przypadku przypasowania więcej niż jednej reguły w danym kontekście, priorytet reguły jest tym wyższy, im wyższa jest długość kontekstu pasującego do reguły. Tabele transkrypcji są ekonomiczną reprezentacją zbioru reguł transkrypcji, dostosowaną do opisu relacji o wysokiej regularności.

Do opisu relacji o niskiej regularności proponuje się zastosowanie odrębnej reprezentacji reguł transkrypcji. Wyjątkiem transkrypcji $E[1..k][2]$ nazywamy macierz $k \times 2$, spełniającą następujące warunki:

1.
$$E[i][1] \subset \mathcal{L} \text{dla } 1 \leqslant i \leqslant k, \tag{8.7}$$

2.
$$E[i][2] \subset \mathcal{G}^* \cup \{\lambda\}, \text{dla } 1 \leqslant i \leqslant k, \tag{8.8}$$

gdzie λ oznacza transkrypcję domyślną (przy użyciu tabel transkrypcji). Dla dowolnego i takiego, że $1 \leqslant i \leqslant k$, jeśli $E[i][2] \neq \lambda$, to $E[1..k][2]$ reprezentuje zbiór reguł transkrypcji:

$$\bigcup_{g \in E[i][1]} g \rightarrow E[i][2]/E[1..i-1][1]\_E[i+1..k][1]. \tag{8.9}$$

Reguły transkrypcji zamieniane są do postaci pojedynczego, trójstrumieniowego FST zgodnie z algorytmem 8.1. Kolejne strumienie reprezentują: 1) literę, której dotyczy reguła, 2) lewy kontekst reguły (w kierunku malejących indeksów), 2) prawy kontekst reguły (w kierunku rosnących indeksów). W algorytmie 8.1 wyrażenia dotyczące strumieni 2 oraz 3 są oznaczane linią poziomą odpowiednio pod oraz nad tekstem. W implementacji oznaczenia te zapisywane są w postaci dodatkowych bitów znaków, co umożliwia zastosowanie jednostrumieniowego FST. Dla zwięzłości zapisu podzbiorów zbioru $\mathcal{L}^*$ w tabelach transkrypcji używa się specjalizowanej algebry. Dla dowolnych $A, B \subset \mathcal{L}^*$ określa się działanie dodawania (zapis $A + B$) jako sumę zbiorów A i B oraz działanie mnożenia (zapis $A * B$) jako zbiór złożony ze wszystkich ciągów $\alpha\beta$ takich, że $\alpha \in A$, $\beta \in B$ (Steffen-Batóg i Nowakowski 1992, 149). W wyrażeniach zawierających ww. działania można stosować nawiasy klamrowe (definiowanie zbioru z elementów) oraz okrągłe (kolejność działań).

Algorytm 8.1 Translacja reguł transkrypcji fonematycznej (fonetycznej) z tabel i wyjątków do kodu źródłowego w języku C.

```
 1: Phonetizer.translateRules(𝒯:Set, ℰ:Set):String
Wejście 𝒯: tabele transkrypcji
Wejście ℰ: wyjątki transkrypcji
Wyjście kod źródłowy C
 2: FST F
 3: for all T ∈ 𝒯 do
 4:   for i = 2 to m do
 5:     for j = 2 to n do
 6:       String a ← T[i][1] * overline(T[1][j]) * T[1][1]
 7:       String r ←overline(Phonetizer.alpha2regex(a))
 8:       FST H ←FSA6.buildFST(r,T[i][j])
 9:       F ←FSA6.addOverwrite(F, H)
10:     end for
11:   end for
12: end for
13: for all E ∈ ℰ do
14:   for i = 1 to k do
15:     if E[i][2] ≠ λ then
16:       String a ← E[1..i − 1][1] * overline(E[i + 1..k][1]) * E[i][2]
17:       String r ←overline(Phonetizer.alpha2regex(a))
18:       FST H ←FSA6.buildFST(r,E[i][2])
19:       F ←FSA6.addOverwrite(F, H)
20:     end if
21:   end for
22: end for
23: FST O ← FSA6.optimize(F)
24: return FSA6.exportC(O)
```

Zmienne, symbole oraz algorytmy pomocnicze stosowane w algorytmie 8.1:

- m:Integer, n:Integer — liczba wierszy (m) oraz kolumn (n) w tabeli transkrypcji.
- k:Integer — liczba wierszy w wyjątku transkrypcji.
- Phonetizer.alpha2regex(s:String): String — translacja wyrażeń algebry $(+, *)$ znajdujących się w napisie s do odpowiadających im wyrażeń regularnych zgodnych z językiem Perl.
- FSA6.bulidFST(r:String, e:String):FST — dla danego wyrażenia regularnego r oraz napisu e, zwraca FST, który dla każdego napisu pasującego do r emituje e.
- FSA6.addOverwrite(F:FST, G:FST): FST — zwraca FST będący sumą F oraz G. Jeśli dla pewnego napisu s przetworniki F oraz G dają różne emisje f oraz odpowiednio g, to FST wynikowy dla s daje emisję g.
- FSA6.optimize(F:FST):FST — determinizuje oraz minimalizuje FST.
- FSA6.exportC(F:FST):String — zapisuje FST w postaci kodów źródłowych w języku C.

Algorytmy, których nazwy zawierają przedrostek FSA6 zaimplementowano przy użyciu biblioteki FSA6 (van Noord 2009) oraz interpretera języka Prolog (dialekt SICstus Prolog 4).

W komponencie Phonetizer stosuje się 87-głoskowy segmentalny system fonetyczny oparty na systemie, który zaproponowali Steffen-Batóg i Nowakowski (1992, 149). Przy opisie systemu wykorzystano ponadto prace, których autorami są: Jassem (1973), Steffen-Batogowa (1975), Wells (1997), Wypych i inni (2003) oraz Jassem (2003b). Proponowane rozszerzenia polskiego SAMPA (Wells 1997) na głoski oparto na dwóch założeniach: 1) niemodyfikowania istniejących oznaczeń SAMPA oraz 2) stosowania wyłącznie znaków dopuszczanych w popularnych systemach plików (NTFS, EXT, FAT).[3] W tabelach 8.1 i 8.2 znajdują się opisy przyjętych segmentalnych systemów: fonetycznego (kolumny z etykietą N) oraz fonologicznego (kolumny z etykietą B). W tabelach wyodrębiono allofony specyficzne dla języka polskiego (kolumny z etykietą S).

Na wydrukach 8.2 oraz 8.3 przedstawiono przykładową tabelę transkrypcji oraz niewielki podzbiór wyjątków transkrypcji. Oprócz działań '+' i '*' opisanych wcześniej, w kodach źródłowych tabel transkrypcji stosuje się działanie '-' interpretowane jako różnica zbiorów. Dowolny ciąg liter α dany jako argument działania interpretowany jest jako zbiór $\{\alpha\}$. Dla zwiększenia czytelności, w kodach źródłowych wybranym podzbiorom $\mathcal{L}^*$ nadaje się nazwy (por. X, S1, S2 na wydruku 8.2 oraz U na wydruku 8.3). W definicjach zbiorów przez wyliczenie stosuje się nawiasy klamrowe oraz przecinki w znaczeniu analogicznym do nawiasów okrągłych oraz znaku '+'. Znak '#' reprezentuje granicę leksów a znak '1' reprezentuje ϵ Podobnie jak w języku C++ napis '//' rozpoczyna komentarz. W kodach źródłowych wyjątków transkrypcji spacja oddziela wiersze a dwukropek kolumny wyjątku transkrypcji.

Listing 8.2: Kody źródłowe jednej z tabel transkrypcji dla litery 'i'.

```
X={a,ą,b,c,ć,d,e,ę,f,g,h,i,j,k,l,ł,m,n,ń,o,ó,p,q,r,s,ś,t,u,v,w,x,y,z,ź,ż,#}
S1=(X–{i,o,k})l+(X–m)il+(X–#){mi,po,k}l+(X–p)ol+(X–{t,p})r+(X–{#,a}){tr,pr}
S2=a{m,b}+ali(X–c)+o{d,l,z}+{y,ó,ę,ą}+o(X–(#+{d,l,n,z}))+u(X–#)
```

i	; em;	e(X–m);	S2;	alic;	al(X–i);	{a,o,u}#;	on;	a(X–{#,m,b})
p,b,f,w,v,m,t	; j ;	j	; j ;	j ;	j	; j	; j ;	j
d,h,cz,k,g,ż	; j ;	j	; j ;	j ;	j	; j	; j ;	j
c,n	; 1 ;	1	; 1 ;	1 ;	1	; 1	; 1 ;	1
(X–c)z+(X–#)s	; 1 ;	1	; 1 ;	1 ;	1	; 1	; 1 ;	1
#s	; 1 ;	1	; 1 ;	ij ;	ij	; 1	; 1 ;	1
#{mi,po}l	; j ;	i	; i ;	i ;	i	; j	; j ;	j
{#,a}{t,p}r	; ij ;	ij	; ij ;	;	;	ij	; ij ;	ij
#kl	; ;	ij	; ;	;	;	;	; ;	ij
S1	; j ;	j	; j ;	j ;	j	; j	; j ;	j

Listing 8.3: Przykładowy zbiór wyjątków transkrypcji.

8.1.6 MorphoAnalyser

Agent MorphoAnalyser tworzy segmenty morfów (etykieta Morph) na podstawie wejściowej ścieżki liter, przy uwzględnieniu lokalizacji leksów.

[3] Założeń tych nie spełniają wcześniejsze rozszerzenia SAMPA, które przedstawili Wypych i inni (2003).

ID	IPA	IPA	SAMPA	SAMPA	SAMPA	Slavonic	Unicode hex code	IPA phonetic features
	N	B	N	S	B	N	N	N
1.	i	i	i	i	i	i	0069	unrounded high front vowel
2.	ĩ	i	i~	i	i	ĩ	0069 0342	unrounded high front nasalized vowel
3.	ɨ	ɨ	I	I	I	y	0268	unrounded high central vowel
4.	ɨ̃	ɨ	I~	I	I	ỹ	0129 0335	unrounded high central nasalized vowel
5.	ɛ	e	e	e	e	e	025B	unrounded mid front vowel
6.	ɛ	e	e+	e+	e	ė	0065	unrounded mid front raised advanced vowel
7.	ɛ̃	e	e~	e	e	ẽ	025B 0342	unrounded mid front nasalized vowel
8.	ɛ̟̃	e	e+~	e	e	ẽ	025B 0342 031D	unrounded mid front advanced nasalized vowel
9.	a	a	a	a	a	a	0061	unrounded low central vowel
10.	a̟	a	a+	a+	a	ä	0061 031F	unrounded low central advanced vowel
11.	ã	a	a~	a	a	ã	0061 0342	unrounded low central nasalized vowel
12.	ã̟	a	a+~	a	a	ã̟	0061 031F 0342	unrounded low central advanced nasalized vowel
13.	ɔ	o	o	o	o	o	0254	rounded mid back vowel
14.	ɔ̟	o	o+	o	o	ȯ	0254 031F	rounded mid back advanced vowel
15.	ɔ̃	o	o~	o	o	õ	0254 0342	rounded mid back nasalized vowel
16.	ɔ̟̃	o	o+~	o	o	õ̟	0254 0342 031F	rounded mid back advanced nasalized vowel
17.	u	u	u	u	u	u	0075	rounded high back vowel
18.	u̟	u	u+	u	u	ü	0075 031F	rounded high back advanced vowel
19.	ũ	u	u~	u	u	ũ	0075 0342	rounded high back nasalized vowel
20.	ũ̟	u	u+~	u	u	ũ̟	0075 031F 0342	rounded high back advanced nasalized vowel
21.	j	j	j	j	j	i̯	006A	voiced palatal approximant consonant
22.	j̃	ɲ	j~	j~	n'	ń	006A 0342	voiced palatal approximant nasalized consonant
23.	w	w	w	w	w	u̯	0077	voiced bilabial velar approximant consonant
24.	w̥	w	w%	w%	w	u̯̥	0077 0325	voiceless bilabial velar approximant consonant
25.	wʲ	w	w,	w	w		0077 02B2	voiced bilabial velar approximant palatalized consonant
26.	w̃	ŋ	w~	w~	N	ŋ	0077 0342	voiced bilabial velar approximant nasalized consonant
27.	l	l	l	l	l	l	006C	voiced alveolar lateral-approximant consonant
28.	l̥	l	l%	l%	l	l̥	006C 0325	voiceless alveolar lateral-approximant consonant
29.	lʲ	l	l,	l	l	l’	006C 02B2	voiced alveolar lateral-approximant palatalized consonant
30.	r	r	r	r	r	r	0072	voiced alveolar trill consonant
31.	r̥	r	r%	r%	r	r̥	0072 0325	voiceless alveolar trill consonant
32.	rʲ	r	r,	r	r	r’	0072 02B2	voiced alveolar trill palatalized consonant
33.	m	m	m	m	m	m	006D	voiced bilabial nasal consonant
34.	m̥	m	m%	m%	m	m̥	006D 0325	voiceless bilabial nasal consonant
35.	mʲ	m	m,	m	m	m’	006D 02B2	voiced bilabial nasal palatalized consonant
36.	n	n	n	n	n	n	006E	voiced dental nasal consonant
37.	n̥	n	n%	n	n	n̥	006E 0325	voiceless dental nasal consonant
38.	nʲ	n	n,	n,	n	n’	006E 02B2	voiced dental nasal palatalized consonant
39.	n̪	n	n–	n–	n	n̪	006E 0331	voiced alveolar nasal retracted consonant
40.	ɲ	ɲ	n'	n'	n'	ń	0272	voiced palatal nasal consonant
41.	ɲ̥	ɲ	n'%	n'%	n'	ń̥	0272 0325	voiceless palatal nasal consonant
42.	ŋ	ŋ	N	N	N	ŋ	014B	voiced velar nasal consonant
43.	ŋ̊	ŋ	N%	N%	N	ŋ̥	014B 030A	voiceless velar nasal consonant
44.	ŋʲ	ŋ	N,	N	N	ŋ’	014B 02B2	voiced velar nasal palatalized consonant
45.	ŋ̊ʲ	ŋ	N,%	N	N	ŋ̥’	014B 030A 02B2	voiceless velar nasal palatalized consonant

Tabela 8.1: Segmentalny system fonetyczny i fonologiczny języka polskiego. Cz. 1.

ID	IPA	IPA	SAMPA	SAMPA	SAMPA	Slavonic	Unicode hex code	IPA phonetic features
	N	B	N	S	B	N	N	N
46.	v	v	v	v	v	v	0076	voiced labiodental fricative consonant
47.	vʲ	v	v,	v	v	v'	0076 02B2	voiced labiodental fricative palatalized consonant
48.	f	f	f	f	f	f	0066	voiceless labiodental fricative consonant
49.	fʲ	f	f,	f	f	f'	0066 02B2	voiceless labiodental fricative palatalized consonant
50.	z	z	z	z	z	z	007A	voiced dental fricative consonant
51.	zʲ	z	z,	z	z	z'	007A 02B2	voiced dental fricative palatalized consonant
52.	ʑ	ʑ	z'	z'	z'	ź	0291	voiced palatal fricative consonant
53.	s	s	s	s	s	s	0073	voiceless dental fricative consonant
54.	sʲ	s	s,	s	s	s'	0073 02B2	voiceless dental fricative palatalized consonant
55.	ɕ	ɕ	s'	s'	s'	ś	0255	voiceless palatal fricative consonant
56.	ʒ̪	ʒ	Z	Z	Z	ž	0292 031F	voiced alveolar fricative consonant
57.	ʒʲ	ʒ	Z,	Z	Z	ź	0292	voiced alveolar fricative palatalized consonant
58.	ʃ̪	ʃ	S	S	S	š	0283 031F	voiceless alveolar fricative consonant
59.	ʃʲ	ʃ	S,	S	S	ś	0283	voiceless alveolar fricative palatalized consonant
60.	ɣ	x	x	x	x	ɣ	0263	voiced velar fricative consonant
61.	ɣʲ	x	x,	x	x	ɣ'	0263 02B2	voiced velar fricative palatalized consonant
62.	x	x	x%	x%	x	x	0078	voiceless velar fricative consonant
63.	ç	x	x,%	x,%	x	x'	00E7	voiceless velar fricative palatalized consonant
64.	d͡z	d͡z	dz	dz	dz	ʒ	02A3	voiced dental postdental affricate consonant
65.	d͡zʲ	d͡z	dz,	dz	dz	ʒ'	02A3 02B2	voiced dental postdental affricate palatalized consonant
66.	d͡ʑ	d͡ʑ	dz'	dz'	dz'	ʑ	02A5	voiced palatal affricate consonant
67.	t͡s	t͡s	ts	ts	ts	c	02A6	voiceless dental postdental affricate consonant
68.	t͡sʲ	t͡s	ts,	ts	ts	c'	02A6 02B2	voiceless dental postdental affricate palatalized consonant
69.	t͡ɕ	t͡ɕ	ts'	ts'	ts'	ć	02A8	voiceless palatal affricate consonant
70.	d͡ʒ̪	d͡ʒ	dZ	dZ	dZ	ǯ	02A4 0331 031F	voiced alveolar affricate consonant
71.	d͡ʒʲ	d͡ʒ	dZ,	dZ	dZ	ʑ	02A4	voiced alveolar affricate palatalized consonant
72.	t͡ʃ̪	t͡ʃ	tS	tS	tS	č	02A7 0331 031F	voiceless alveolar affricate consonant
73.	t͡ʃʲ	t͡ʃ	tS,	tS	tS	č	02A7	voiceless alveolar affricate palatalized consonant
74.	b	b	b	b	b	b	0062	voiced bilabial plosive consonant
75.	bʲ	b	b,	b	b	b'	0062 02B2	voiced bilabial plosive palatalized consonant
76.	p	p	p	p	p	p	0070	voiceless bilabial plosive consonant
77.	pʲ	p	p,	p	p	p'	0070 02B2	voiceless bilabial plosive palatalized consonant
78.	d̪	d	d	d	d	d	0064 032A	voiced dental plosive consonant
79.	dʲ	d	d,	d	d	d'	0064 02B2	voiced dental plosive palatalized consonant
80.	d̻	d	d-	d-	d	d̻	0064 0331	voiced alveolar plosive consonant
81.	t̪	t	t	t	t	t	0074 032A	voiceless dental plosive retracted consonant
82.	tʲ	t	t,	t	t	t'	0074 02B2	voiceless dental plosive palatalized consonant
83.	t̻	t	t-	t-	t	t-	0074 0331	voiceless alveolar plosive retracted consonant
84.	g	g	g	g	g	g	0067	voiced velar plosive consonant
85.	ɟ	g	J	J	g	g'	025F	voiced palatal plosive consonant
86.	k	k	k	k	k	k	006B	voiceless velar plosive consonant
87.	c	k	c	c	k	k'	0063	voiceless palatal plosive consonant

Tabela 8.2: Segmentalny system fonetyczny i fonologiczny języka polskiego. Cz. 2.

Na porzeby bieżącej pracy stworzono prosty, regułowy układ analizy morfologicznej. **Regułą morfologiczną** nazywamy czwórkę uporządkowaną:

$$(L, M, R, C), \tag{8.10}$$

gdzie $L, M, R \subset \mathcal{G}^*$ oraz $C \in \{\text{Root}, \text{Affix}\}$. Reguła morfologiczna (L, M, C, R) stwierdza, że jeśli w wejściowej ścieżce liter pojawi się ciąg $\alpha\beta\gamma$, gdzie $\alpha \in L$, $\beta \in M$ oraz $\gamma \in R$, to β wyznacza granice allomorfu morefemu klasy C (por. rycina 6.4 na stronie 98). Podobnie jak w przypadku agenta Phonetizer opisanego w sekcji 8.1.5, zbiór reguł morfologicznych jest kompilowany do postaci minimalnego FST przy użyciu pakietu FSA6. W przypadku konfliktu reguł wybierane jest dopasowanie o największej liczbie liter w $\alpha\beta\gamma$.

Na wydruku 8.4 pokazano fragment kodów źródłowych reguł morfologicznych. W każdej linii jest zapisywana co najwyżej jedna reguła morfologiczna. Kolejne elementy reguły morfologicznej są oddzielane spacjami. W kodach źródłowych na wydruku 8.4 stosuje się konwencje opisane dla reguł transkrypcji (sekcja 8.1.5).

Listing 8.4: Kody źródłowe wybranych reguł morfologicznych.

8.1.7 PhonologicalSyllabifier

Agent PhonologicalSyllabifier tworzy kratę sylab fonologicznych (etykieta PhonologicalSyllable) na podstawie ścieżki głosek (etykieta Phone) oraz ścieżki leksów (etykieta Lex). PhonologicalSyllabifier jest oparty na regułach sylabizacji fonologicznej, które zaproponował Jassem (2003c). Wcześniej podobne reguły zastosował Jassem (1996a) do sylabizacji fonologicznej nazw własnych.

Regułą sylabizacji nazywamy parę uporządkowaną $(L, R) \in \mathcal{G}^* \times \mathcal{G}^*$. Reguła sylabizacji (L, R) stanowi, że jeśli ciąg etykiet ustalonej ścieżki segmentów głoskowych ma postać $\alpha\beta$, gdzie $\alpha \in L$, $\beta \in R$, to granica sylab fonologicznych przypada na granicy napisów α oraz β. Dopuszcza się, by zbiór reguł sylabizacji wskazywał alternatywne granice sylab (np. w przypadku kontekstów ambisylabicznych).

Reguły sylabizacji fonologicznej zapisywane są w formalizmie analogicznym do reguł transkrypcji oraz reguł morfologicznych, które opisano we wcześniejszych sekcjach. Reguły sylabizacji są kompilowane do postaci minimalnego FST za pomocą pakietu FSA6. Na wydruku 8.5 zamieszczono kod źródłowy przykładowych reguł sylabizacji fonologicznej. W każdej linii jest zapisywana co najwyżej jedna reguła a elementy składowe reguły są oddzielone spacją. W kodach źródłowych reguł obowiązują konwencje opisane w poprzednich sekcjach, w szczególności działania na zbiorach napisów.

Listing 8.5: Kody źródłowe wybranych reguł sylabizacji fonologicznej.

8.1.8 MorphoSyllabifier

Agent MorphoSyllabifier wyznacza sylabizację morfologiczną w anotacji kratowej zawierającej sylaby fonologiczne (etykieta PhonologicalSyllable), morfy (etykieta Morph) oraz leksy (etykieta Lex). Definicję sylabizacji morfologicznej wyjaśniono w sekcji 2.4.

Niech będzie ustalona anotacja (S, a). Przez $S_a^{\{\mathrm{T}\}}$ będziemy oznaczać zbiór wszystkich segmentów $s \in S$ takich, że $\overline{s}$ jest typu T. Sylabizacja morfologiczna jest wyznaczana z użyciem algorytmu 6.1 (por. strona 94) na segmentacji $S_a^{\{PhonologicalSyllable\}}$. Wartość funkcji kosztu cząstkowego $\check{f}_a(s_0, s_1)$ jest równa liczbie niespełnionych warunków z poniższej listy:

1.
$$s_0 = \varnothing, \tag{8.11}$$

2.
$$\exists_{l \in S_a^{\{Lex\}}} \; l \rhd s_0 \wedge \overrightarrow{l} = \overrightarrow{s_0}, \tag{8.12}$$

3.
$$\exists_{p \in S_a^{\{Prefix\}}} \; s_0 \rhd p \wedge \overrightarrow{p} = \overrightarrow{s_0}, \tag{8.13}$$

4.
$$\exists_{p \in S_a^{\{Postfix\}}} \; s_1 \rhd p \wedge \overleftarrow{p} = \overleftarrow{s_1}. \tag{8.14}$$

Przyjęta funkcja kosztu cząstkowego promuje ścieżki zawierające prawe granice leksów (wyrażenie 8.12), prawe granice prefiksów (wyrażenie 8.13) oraz lewe granice postfiksów (wyrażenie 8.14).

8.2 Podukład analizy akustycznej

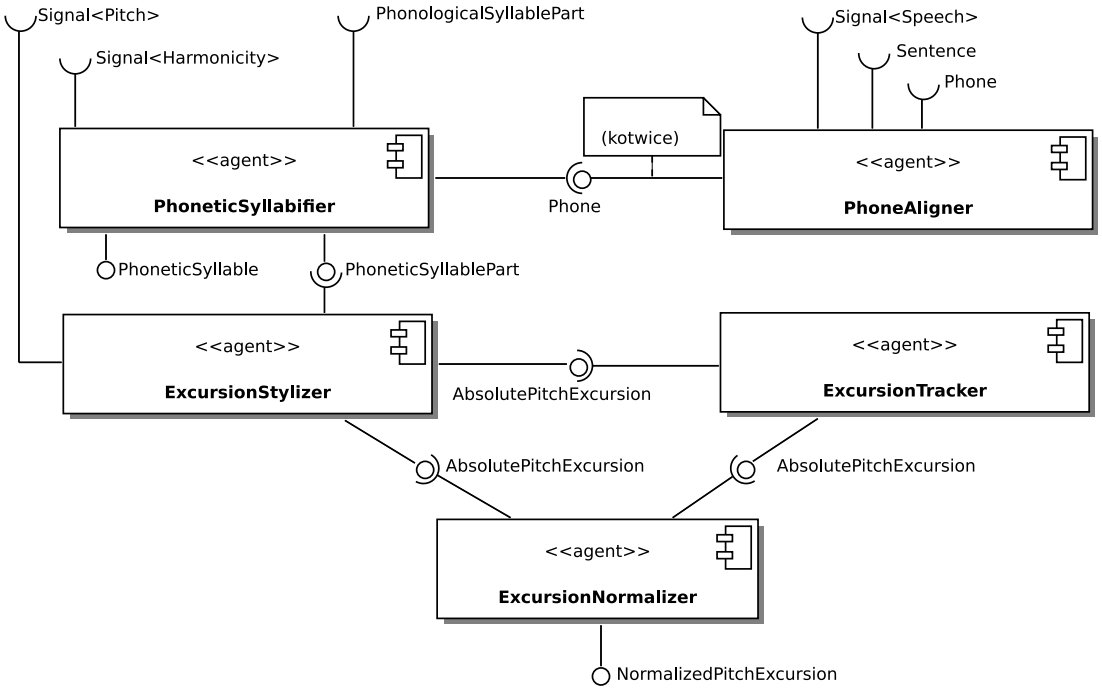
Podukład analizy akustycznej wykonuje fonetyczną analizę tonalną w oparciu o dane z układu sygnałowej analizy tonalnej (por. rozdział 7) oraz podukładu analizy ortograficznej (por. sekcja 8.1). Na rycinie 8.2 przedstawiono diagram komponentów podukładu analizy akustycznej. W kolejnych podsekcjach zawarto opisy poszczególnych komponentów.

8.2.1 PhoneAligner

Agent PhoneAligner ustala lokalizacje czasowe kotwic ścieżki głosek (etykieta Phone) obejmowanej przez dane zdanie (etykieta Sentence).

Agent PhoneAligner został zaimplementowany przy użyciu układu rozpoznawania mowy Sonic w wersji 2.0 beta 5 udostępnionego[4] przez Univeristy of Colorado w Boulder (Pellom i Hacioglu 2005). W układzie Sonic jest stosowana anotacja sygnałowa PMVDR, która w porównaniu do MFCC, korzystnie wyływa na skuteczność układu (WER) w warunkach obniżonego SNR w sygnale wejściowym (Yapanel i Hansen 2008). Sonic zbudowano w oparciu o generatywny paradygmat statystycznego rozpoznawania mowy, w którym występuje podział na model akustyczny (AM, *Acoustic Model*) oraz model językowy (LM, *Language Model*). Model akustyczny układu Sonic jest oparty na modelach CDGMM (*Continous-Density Gaussian Mixture Model*) oraz HMM. Model językowy układu Sonic jest oparty na n-gramach o dowolnej długości.

[4]Do roku 2006 University of Colorado udzielał darmowych licencji na wykorzystanie układu Sonic w niekomercyjnych pracach badawczych.



Rycina 8.2: Podukład analizy akustycznej w układzie fonetycznej analizy tonalnej. Diagram komponentów UML.

Podstawową jednostką modelu akustycznego w Sonic jest głoska. Każdej głosce przyporządkowuje się trójstanowy HMM o topologii Bakisa. Model akustyczny Sonic zawiera zarówno bezkontekstowe jak i kontkestowe (trifonowe) HMM głosek. Prawdopodobieństwo powtórzenia stanu HMM w kolejnej chwili jest modelowane rozkładem gamma. Skrajne stany trifonowych HMM są współdzielone w podzbiorach HMM wyznaczanych automatycznie przy użyciu algorytmu CART (*tied-state HMM*).

Na potrzeby niniejszej pracy przygotowano polski model akustyczny dla Sonic. Uczenie modelu akustycznego wykonano na korpusie PLT, który opublikowali Dziubalska-Kołaczyk i inni (2004). Korpus PLT zawiera 5932 jednozdaniowych wypowiedzi odczytanych w warunkach biurowych przez 113 osób w grupie wiekowej 20-30 lat. Teksty wypowiedzi w korpusie PLT dobrano w sposób zapewniających różnorodność kontekstów głoskowych.

Za zestaw głosek (*phoneset*) w polskim modelu akustycznym dla Sonic przyjęto zbiór allofonów specyficznych przedstawiony na stronach 118 oraz 119. Przy użyciu agenta Phonetizer (por. sekcja 8.1.5) wykonano fonetyzację tekstów w korpusie PLT. Następnie przygotowano skrypt Perl, który na podstawie tekstów ortograficznych oraz otrzymanych tekstów fonetycznych wygenerował słownik wymowy w formacie wymaganym przez Sonic. Zdania korpusu PLT podzielono w proporcji 9:1 odpowiednio na zbiór uczący oraz zbiór testowy. Estymację parametrów HMM wykonano za pomocją programu uczącego wchodzącego w skład pakietu Sonic. Program uczący Sonic naprzemiennie odszukuje ścieżeki Viterbiego zdań oraz estymuje parametry HMM fonemów algorytmem EM przy zastosowaniu algorytmu Bauma-Welcha (Pellom i Hacioglu 2005, 30).

Dla zweryfikowania poprawności otrzymanego otrzymanego modelu akustycznego przygotowano trigramowy model języka (por. Pellom 2005). Do wyuczenia modelu języka wykorzystano zbiór 57809 zdań z korpusu tekstów PLT, który zebrali Dziubalska-Kołaczyk i

inni (2004). Z korpusu tekstów PLT wykluczono zdania występujące w zbiorze testowym modelu akustycznego a następnie wykonano estymację modelu n-gramowego za pomocą pakietu *CMU-Cambridge Statistical Language Modeling Toolkit* (Clarkson i Rosenfeld 1997). W estymacji prawdopodobieństw n-gramów zastosowano techniki wygładzania, które zaproponowali Witten i Bell (1991) oraz Katz (1987). Skuteczność układu Sonic z opisanymi modelami polskimi na zbiorze testowym modelu akustycznego wyniosła WER=19.6%, co uznano za potwierdzenie poprawności wytrenowania modelu akustycznego.

W bieżącej wersji agenta PhoneAligner stosuje się bezkontekstowe modele akustyczne. W celu ustalenia granic głosek wykonywany jest algorytm Viterbiego (zaimplementowany w ramach układu Sonic) dla wejściowego sygnału mowy oraz zadanego ciągu głosek (*forced alignment*).

8.2.2 PhoneticSyllabifier

Agent PhoneticSyllabifier tworzy ścieżkę sylab fonetycznych (etykieta PhoneticSyllable) oraz ścieżkę części sylab fonetycznych (etykiety klas potomnych PhoneticSyllablePart) przyjmując na wejściu przebieg harmoniczności, ścieżkę sylab morfologicznych (etykieta PhonologicalSyllable) oraz części sylab fonologicznych (etykiaty klas potomnych PhonologicalSyllablePart). Zauważmy, że w wyniku działania agenta PhoneAligner (por. sekcja 8.2.1) kotwice segmentów wejściowych agenta PhoneticSyllabifier mają określoną lokalizację czasową w sygnale mowy (przebiegu harmoniczności).

Dla uproszczenia, w bieżącej implementacji zakłada się, że każdej sylabie fonologicznej odpowiada dokładnie jedna sylaba fonetyczna. Algorytm 8.2 przedstawia metodę wyznaczania granic części sylaby fonetycznej zastosowaną w agencie PhoneticSyllabifier. Funkcja pomocnicza $addSegment(A, b, f, a)$ stosowana w algorytmie 8.2 dodaje do anotacji A segment o kotwicy początkowej b, kotwicy końcowej f oraz etykiecie a.

Zastosowanie ośrodków sylab fonetycznych w dalszych etapach analizy pozwala ograniczyć propagację błędów z wcześniejszych etapów analizy, przede wszystkim sylabizacji fonologicznej oraz lokalizacji granic czasowych głosek. Jednocześnie, jak wielokrotnie wykazano, przebieg F_0 poza ośrodkami sylab fonetycznych ma marginalny wpływ na percepcję wysokości tonu (por. Dziubalska-Kołaczyk 2002; van Santen 2002).

8.2.3 ExcursionStylizer

Agent ExcursionStylizer dla danych: 1) sylaby fonologicznej s, 2) zakotwiczonego ośrodka sylaby fonetycznej o, 3) sygnału F_0 f oraz 4) przebiegu harmoniczności h tworzy dwusegmentową ścieżkę o początku w kotwicy $\overleftarrow{s}$ oraz końcu w kotwicy $\overrightarrow{s}$ etykietowaną ontologiami klasy StylizedPitchExcursion. W założeniach, obiekt klasy StylizedPitchExcursion reprezentuje wrażenia słuchowe odpowiadające sygnałowi etykietowanego segmentu odsłuchanemu w izolacji. Klasa StylizedPitchExcursion rozszerza interfejs PitchExcursion (por. diagram 6.3 na stronie 97) o pięć atrybutów rzeczywistych, które reprezentują *monotoniczny* przebieg wysokości tonu w granicach segmentu. Klasa StylizedPitchExcursion zawiera następujące atrybuty: PITCH (wysokość tonu), SLOPE (nachylenie przebiegu wysokości tonu), VOICINGDUR (czas trwania ramek dźwięcznych), HARMONICITY (harmoniczność) oraz DISPERSION (rozproszenie). Tworzenie wynikowej ścieżki segmentów agenta ExcursionStylizer obejmuje trzy etapy: 1) rekonstrukcję przebiegu wysokości tonu w granicach sylaby fonologicznej (algorytm 8.3),

Algorytm 8.2 Wyznaczanie granic części sylaby fonetycznej na podstawie sylabizacji fonologicznej oraz harmoniczności.

1: PhoneticSyllabifier.syllablePartition(A:Annotation,s:Segment,h:Signal):Annotation
Wejście A:anotacja kratowa zawierająca ścieżkę sylab fonologicznych oraz ścieżkę części sylab fonologicznych
Wejście s: segment sylaby fonologicznej należący do A
Wejście h: przebieg harmoniczności sygnału mowy
Wyjście Anotacja A rozszerzona o sylabę fonetyczną oraz części sylaby fonetycznej odpowiadającej s
2: $(S, a) \leftarrow A$
3: **for all** $u \in S : u \rhd s \wedge a(u) \in$ PhoneticSyllabeNucleus **do**
4: $m \leftarrow \operatorname{argmax}_{m \in [\overleftarrow{u}; \overrightarrow{u}]} h(m)$
5: $b \leftarrow \overleftarrow{s}$
6: $B \leftarrow \{i \in [b; m] : h(i) - h(m) < -6\text{dB}\}$
7: **if** $|B| > 0$ **then**
8: $b \leftarrow \max B$
9: addSegment($A, \overleftarrow{s}, b$,PhoneticSyllableOnset)
10: **end if**
11: $f \leftarrow \overrightarrow{s}$
12: $F \leftarrow \{i \in [m; f] : h(i) - h(m) < -6\text{dB}\}$
13: **if** $|F| > 0$ **then**
14: $f \leftarrow \min F$
15: addSegment($A, f, \overrightarrow{s}$,PhoneticSyllableCoda)
16: **end if**
17: addSegment(A, b, f,PhoneticSyllableNucleus)
18: **end for**
19: **return** A

2) wyznaczenie czasu kotwicy łączącej dwa segmenty ścieżki wynikowej (wzór 8.16) oraz 3) wyznaczenie wartości etykiet ścieżki wynikowej (wzory od 8.17 do 8.21).

Celem algorytmu 8.3 jest wyeliminowanie lokalnych błędów ekstrakcji F_0. Algorytm 8.3 realizuje suprasegmentalną sygnałową analizę tonalną (por. sekcja 3.3 na stronie 43), której zasięg ograniczono do pojedynczej sylaby fonetycznej.

Parametry stosowane w algorytmie 8.3:

- α:Real — liczba jednostek skali odpowiadająca podwojeniu F_0; $\alpha = 12$ (skala półtonowa).
- β:Real — maksymalna różnica F_0 między sąsiednimi ramkami sygnałowej anotacji tonalnej; 1ST$\leqslant \beta \leq$3ST.
- γ:Real — maksymalna korekta F_0; 2ST$\leqslant \gamma \leqslant$4ST

Przyjmijmy, że

$$f' = \text{ExcursionStylizer.pitchTracking}(f, h, o). \tag{8.15}$$

Algorytm 8.3 Wyznaczanie przebiegu wysokości tonu w granicach sylaby fonologicznej.

1: ExcursionStylizer.pitchTracking(f:Signal,h:Signal,o:Segment):Signal
Wejście f: przebieg F_0 (sygnał cyfrowy, skala półtonowa)
Wejście h: przebieg harmoniczności (sygnał cyfrowy, skala ilorazowa)
Wejście n: ośrodek sylaby fonetycznej
Wyjście przebieg wysokości tonu (skala półtonowa)
2: Signal d
3: **for** $i = 1$ **to** $|f| - 1$ **do**
4: $\quad d[i] \leftarrow \alpha * \left\lceil \frac{f[i]-f[i-1]}{\alpha} - 0.5 \right\rceil$
5: $\quad$ **if** $d[i] > \beta$ **then**
6: $\quad\quad d[i] \leftarrow 0$
7: $\quad$ **end if**
8: **end for**
9: Integer $m \leftarrow \underset{\overleftarrow{o} \leqslant i < \overrightarrow{o}}{\operatorname{argmax}} h(i)$
10: Signal $p \leftarrow f[m]$
11: **for** $i = m - 1$ **to** 0 **do**
12: $\quad p \leftarrow p - d[i+1]$
13: $\quad$ **if** $|f[i] - p| \leqslant \gamma \wedge h[i] > 0$ **then**
14: $\quad\quad p \leftarrow f[i]$
15: $\quad$ **else**
16: $\quad\quad f[i] \leftarrow p$
17: $\quad$ **end if**
18: **end for**
19: Signal $p \leftarrow f[m]$
20: **for** $i = m + 1$ **to** $|f| - 1$ **do**
21: $\quad p \leftarrow p + d[i]$
22: $\quad$ **if** $|f[i] - p| < \gamma \wedge h[i] > 0$ **then**
23: $\quad\quad p \leftarrow f[i]$
24: $\quad$ **else**
25: $\quad\quad f[i] \leftarrow p$
26: $\quad$ **end if**
27: **end for**
28: **return** f

Czas t kotwicy $\overrightarrow{s_1} = \overleftarrow{s_2}$ w ścieżce (s_1, s_2) tworzonej przez agenta ExcursionStylizer jest określony następująco:

$$t = \underset{k}{\operatorname{argmax}} \left| \sum_{i=1}^{k} f[i] - f[i-1] \right| + \left| \sum_{i=k+1}^{|f|-1} f[i] - f[i-1] \right|. \tag{8.16}$$

Przyjmijmy, że I jest zbiorem indeksów sygnału cyfrowego f w przedziale od czasu $\overleftarrow{s}$ (włączając) do czasu $\overrightarrow{s}$ (wyłączając). Wartości funkcji etykietującej $a : \mathbb{Z} \mapsto$ StylizedPitchExcurtion dla każdego segmentu s należącego do ścieżki wynikowej określamy jak poniżej.

Średnia ważona wysokość tonu:

$$\overline{s}.PITCH = \frac{\sum_{i \in I} f'[i]h[i]}{\sum_{i \in I} h[i]} \tag{8.17}$$

Średnia ważona zmiana wysokości tonu:

$$\overline{s}.SLOPE = \frac{\sum_{i \in I} (f'[i] - f'[i-1])h[i]}{\sum_{i \in I} h[i]}, \tag{8.18}$$

przy czym przyjmujemy, że $f'[j-1] = f'[j]$ dla $j = \min(I)$. Zakumulowany czas trwania ramek dźwięcznych:

$$\overline{s}.VOICINGDUR. = |\{i \in I : h[i] > 0\}| \tag{8.19}$$

Średnia harmoniczność:

$$\overline{s}.HARMONICITY = \frac{\sum_{i \in I} h[i]}{|\{i \in I : h[i] > 0\}|} \tag{8.20}$$

Zakumulowany czas trwania ramek skorygowanych:

$$\overline{s}.DISPERSION = |\{i \in I : f[i] \neq f'[i]\}| \tag{8.21}$$

Wyróżnikami proponowanej metody są: 1) przypisanie każdej sylabie fonetycznej dokładnie dwóch segmentów fonetycznej anotacji tonalnej, 2) niezakładanie ciągłości F_0 (poszczególne atrybuty mogą mieć wartość $\varnothing$), 3) określanie wysokości tonu za pomocą średnich ważonych F_0 względem harmoniczności, 4) wprowadzenie rozproszenia — miary stabilności częstotoliwości podstawowej (nasilenie drgania i/lub chrypliwości głosu), 5) wykonywanie suprasegmentalnej analizy tonalnej w zakresie sylaby, 6) określenie średniej harmoniczności segmentu (HARMONICITY).

8.2.4 ExcursionTracker

Agent ExcursionTracker dla danej ścieżki etykietowanej ontologiami StylizedPitchExcursion tworzy ścieżkę etykietowaną ontologiami TrackedPitchExcursion. W założeniach, obiekt klasy TrackedPitchExcursion reprezentuje wrażenia słuchowe odpowiadające sygnałowi etykietowanego segmentu odsłuchanemu w kontekście sygnałów sąsiednich segmentów ścieżki. Klasa TrackedPitchExcursion została wywiedziona z klasy StylizedPitchExcursion, przy czym wprowadzono w niej dodatkowe dwa atrybuty: rzeczywisty PITCHCORR (poprawka wysokości tonu) oraz boole'owski ELISSION (zanik). Atrybut PITCHCORR reprezentuje wielkość korekty atrybutu PITCH wprowadzoną w stosunku do wartości początkowej zawartej w StylizedPitchExcursion. Atrybut ELISION sygnalizuje zanik wpływu sygnału etykietowanego segmentu na percepcję przebiegu wysokości tonu ścieżki segmentów.

Agent ExcursionTracker działa w oparciu o algorytmy strumieniowe 8.4 oraz 8.5. **Istotnością percepcyjną** nazywamy dowolną wielkość pozytywnie skorelowaną z harmonicznością, głośnością i stabilnością ontologii StylizedPitchExcursion. Celem algorytmu 8.4 jest rekonstrukcja wartości atrybutów ontologii StylizedPitchExcursion mających niską istotność percepcyjną. Celem algorytmu 8.5 jest przyrostowa korekta grubych błędów (*gross error*) ekstrakcji F_0. Algorytmy 8.4 oraz 8.5 opracowano na podstawie wiedzy eksperciej oraz analizy korpusu mowy PoInt (Karpiński 2002).

Algorytm 8.4 Detekcja i korekta ontologii StylizedPitchExcursion mających niską istotność percepcyjną.

1: ExcursionTracker.elisionTracking(A:Annotation,p:Path,q:Path,m:Integer):Integer
Wejście $A = (S, a)$: anotacja wejściowa/wyjściowa
Wejście $p \in\bowtie S$: ścieżka wejściowa etykietowana ontologiami StylizedPitchExcursion
Wejście $q \in\bowtie S$: ścieżka wejściowa/wyjściowa etykietowana ontologiami TrackedPitchExcursion
Wejście m: indeks w ścieżkach wejściowych, od którego należy rozpocząć rekonstrukcję
Wyjście indeks w ścieżkach wejściowych, od którego należy rozpocząć kolejną rekonstrukcję
2: $k \leftarrow |q|$
3: **for** $i = k$ **to** $|p| - 1$ **do**
4: $q[i] \leftarrow$addSegment$(S, \overleftarrow{p[i]}, \overrightarrow{p[i]})$
5: TrackedPitchExcursion $e \leftarrow$TrackedPitchExcursion$(\overline{p[i]})$
6: **for all** $(v, h, d) \in E$ **do**
7: **if** $e.VOICINGDUR < v \wedge e.HARMONICITY < h \wedge e.DISPERSION < d$ **then**
8: $e.ELISION \leftarrow$**true**
9: $e.PITCH \leftarrow \varnothing$
10: $e.SLOPE \leftarrow 0$
11: **break**
12: **end if**
13: **end for**
14: $\overline{q[i]} \leftarrow e$
15: **end for**
16: **for** $i = m$ **to** $|q| - 1$ **do**
17: **if** $\overline{q[i]} = \varnothing$ **then**
18: **if** $i = 0$ **then**
19: $n \leftarrow \min\{j \in \{i, i+1, \ldots, |q|\} : q[j] \neq \varnothing\}$
20: **if** $n \neq \varnothing$ **then**
21: $r \leftarrow \overline{q[n]}$
22: **else**
23: **return** 0
24: **end if**
25: **else**
26: $r \leftarrow \overline{q[i-1]}$
27: **end if**
28: $\overline{q[i]}.PITCH \leftarrow r.PITCH$
29: **end if**
30: **end for**
31: **return** $|q|$

Algorytm 8.5 Detekcja i korekta ontologii StylizedPitchExcurison zawierających grube błędy przebiegu wysokości tonu.

ExcursionTracker.pitchTracking(A:Annotation,q:Path,c:Boolean,m:Integer):Integer
Wejście $A = (S, a)$: anotacja wejściowa/wyjściowa
Wejście $q \in \bowtie S$: ścieżka etykietowana ontologiami TrackedPitchExcursion
Wejście c: **true** $\iff$ przetwarzana ścieżka nie będzie już przedłużana
Wejście m: indeks w ścieżkach wejściowych, od którego należy rozpocząć modelowanie
Wyjście indeks ścieżki wejściowej, od którego należy rozpocząć następne modelowanie
$f \leftarrow F$
if $c =$ **true then**
  $f \leftarrow 0$
end if
if $|q| - m \leqslant f$ **then**
  return m
end if
if $m = 0$ **then**
  $k \leftarrow \underset{i}{\operatorname{argmax}}\, \overline{q[i]}.HARMONICITY$
  $M \leftarrow \{i \in \mathbb{Z} : \overline{q[i]}.HARMONICITY > \alpha * \overline{q[k]}.HARMONICITY\}$
  $m \leftarrow \underset{i \in M}{\operatorname{argmax}}\, \overline{q[i]}.PITCH$
  for $i = m - 1$ **downto** 0 **do**
    $d \leftarrow \overline{q[i+1]}.PITCH - \overline{q[i]}.PITCH$
    $\overline{q[i]}.PITCHCORR \leftarrow$ExcursionTracker.pitchOctaveCorrection(d)
    $d' \leftarrow d + \overline{q[i]}.PITCHCORR$
    $\overline{q[i]}.PITCH \leftarrow \overline{q[i+1]}.PITCH - d'$
  end for
  $m \leftarrow m + 1$
end if
for $i = m$ **to** $|q|$ **do**
  $d \leftarrow \overline{q[i+1]}.PITCH - \overline{q[i]}.PITCH$
  $\overline{q[i]}.PITCHCORR \leftarrow$ExcursionTracker.pitchOctaveCorrection(d)
  $d' \leftarrow d + \overline{q[i]}.PITCHCORR$
  $\overline{q[i+1]}.PITCH \leftarrow \overline{q[i]}.PITCH + d'$
end for
return $|q|$

Parametry oraz funkcje stosowane w algorytmach bieżącej sekcji:

- F — minimalna liczba segmentów potrzebna do wyznaczania segmentu referencyjnego w modelowaniu przebiegu wysokości tonu.
- $E \in \mathbb{R}^3$ — reguły detekcji segmentów o niskiej istotności percepcyjnej.
- α — minimalna względna średnia harmoniczność segmentów kandydujących względem segmentu o najwyższej średniej harmoniczności; $\alpha \in [0.3; 0.7]$.
- TrackedPitchExcursion(e:StylizedPitchExcursion):TrackedPitchExcursion — konstruktor obiektów klasy TrackedPitchExcursion na podstawie danej e, który inicjuje atrybuty TrackedPitchExcursion nieobecne w StylizedPitchExcursion wartościami 0 (lub false).

- addSegment(S:Segmentation,b:Anchor,f:Anchor):Segment — funkcja dodająca do S nowy segment oparty na kotwicach b i f oraz zwracająca dodany segment.

- ExcursionTracker.pitchOctaveCorrection(d:Real):Real — funkcja parzysta określająca poprawkę dla różnicy wysokości tonu dwóch sąsiednich segmentów w ścieżce.

8.2.5 ExcursionNormalizer

Agent ExcursionNormalizer dla każdego segmentu etykietowanego ontologią TrackedPitchExcursion tworzy nowy segment etykietowany ontologią NormalizedPitchExcursion. Klasa NormalizedPitchExcursion została wywiedziona z klasy TrackedPitchExcursion bez wprowadzania dodatkowych atrybutów. Zakłada się, że instancje klasy NormalizedPitchExcursion mają zmniejszoną (w stosunku do TrackedPitchExcursion) zależność od pozajęzykowych cech tonalnych.

Agent ExcursionNormalizer działa w oparciu o standaryzację statystyczną atrybutów skalarnych instancji TrackedPitchExcursion. Na etapie przygotowawczym dla k-tego atrybutu rzeczywistego ($k = 0$: PITCH, $k = 1$: SLOPE, itd.) oraz dla każdego identyfikatora mówcy v wyznaczane są: wektor średnich μ_v oraz wektor odchyleń standardowych σ_v na dostępnym sygnale mowy. W trakcie fonetycznej analizy tonalnej dla każdego etykietowanego segmentu wejściowego s jest tworzony etykietowany segment wyjściowy u w oparciu o następujące zależności:

$$\overleftarrow{u} = \overleftarrow{s} \wedge \overrightarrow{u} = \overrightarrow{s}, \tag{8.22}$$

$$\overline{u}[k] = \frac{\overline{s}[k] - \mu_v[k]}{\sigma_v[k]}, \tag{8.23}$$

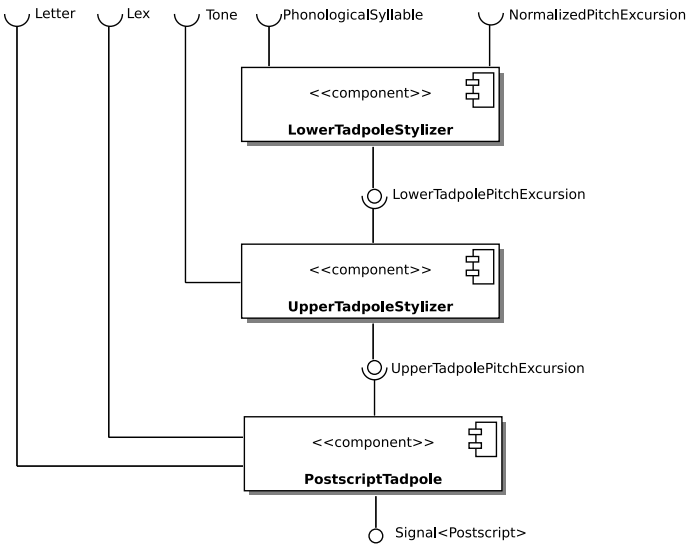
gdzie $\overline{w}[k]$ oznacza wartość k-tego atrybutu rzeczywistego w etykiecie segmentu w.

8.3 Podukład wizualizacji

Jednym z zastosowań proponowanego układu fonetycznej analizy tonalnej jest wizualizacja mowy. W rozdziale 4 przedstawiono szereg metod wizualizacji fonetycznych anotacji tonalnych. W bieżącej sekcji zajmiemy się niepodejmowanym dotąd szerzej tematem automatyzacji tonetycznej transkrypcji międzyliniowej (ITT — *Interlinear Tonetic Transcription*).

Przykład ITT pokazano na stronie 46 na rysunku 4.2. Na wstępie należy zaznaczyć, że ITT nie jest anotacją fonetyczną sensu stricte. Wielkość punktu reprezentującego sylabę w ITT zależy od wystąpienia akcentu realnego na sylabie. Etykiety fonologicznej anotacji tonalnej mają wpływ na wizualizację sylab w pobliżu granic melodii (Wypych 2006).

W niniejszej pracy ITT jest reprezentowana przez dwie warstwy anotacji: ścieżkę segmentów etykietowaną ontologiami TadpolePitchExcursion oraz ścieżkę segmentów etykietowanych ontologiami Tone (poziom fonologiczny, więcej w rozdziale 10). Klasa TadpolePitchExcursion została wywiedziona z klasy PitchExcursion w celu reprezentowania przebiegu wysokości tonu w granicach pojedynczej sylaby (graficznie reprezentowanego przez kropkę z „ogonkiem"). W klasie TadpolePitchExcursion zdefiniowano 3 atrybuty rzeczywiste: 1) PITCH (wysokość tonu), 2) DEC_CHANGE (zwalniająca zmiana wysokości tonu), 3) ACC_CHANGE (przyspieszająca zmiana wysokości tonu). Określenia „zwalniająca" oraz „przyspieszająca" odnoszą



Rycina 8.3: Podukład wizualizacji metodą ITT. Diagram komponentów UML.

się do niezgodności lub odpowiednio zgodności znaków pierwszej i drugiej pochodnej wyidealizowanego przebiegu wysokości tonu. Wartości atrybutów klasy TadpolePitchExcursion interpretowane są na skali standaryzowanej atrybutu PITCH w klasie NormalizedPitchExcursion (por. sekcja 8.2.5).

Wyróżniamy dwa kluczowe problemy automatyzacji ITT: 1) modelowanie percepcji przebiegu F_0 dla poszczególnych sylab, 2) uwzględnienie anotacji fonologicznej (analiza zstępująca). Jak pokazujemy poniżej, fonetyczna anotacja tonalna zaproponowana w sekcji 8.2 pozwala łatwo określić pozycję oraz zakres ikon ITT. Jednocześnie, stosowana przez nas architektura tablicowa (por. rozdział 6) umożliwia praktycznie bez dodatkowych kosztów implementacyjnych rozszerzanie algorytmów analizy indukcyjnej do algorytmów analizy indukcyjno–dedukcyjnej.

Diagram 8.3 przedstawia podukład generujący ITT dla danej ścieżki segmentów etykietowanych ontologiami NormalizedPitchExcursion. W kolejnych podsekcjach opisano agenty proponowanego podukładu.

8.3.1 LowerTadpoleStylizer

Agent LowerTadpoleStylizer dla wejściowej sylaby fonologicznej (etykieta PhonologicalSyllable) oraz obejmowanej przez nią ścieżki segmentów etykietowanych ontologiami NormalizedPitchExcursion tworzy segment etykietowany ontologią LowerTadpolePitchExcursion. Ontologia LowerTadpolePitchExcursion reprezentuje ikonę ITT otrzymaną wyłącznie za pomocą analizy indukcyjnej. Klasa LowerTadpolePitchExcursion została wywiedziona bezpośrednio, bez rozszerzeń z klasy TadpolePitchExcursion.

W algortmie 8.6 przedstawiono sposób wyliczania atrybutów ontologii LowerTadpolePitchExcursion. Przedstawiony algorytm powstał w oparciu o wiedzę ekspercką oraz szereg eksperymentów na korpusie mowy PoInt (Karpiński 2002). W algorytmie zastosowano m.in. pojęcie progu glissando, który opisał Mertens (2004) (por. też sekcja 4.5). Parametry α oraz

β określają próg glissando dla prędkości oraz czasu trwania zmiany wysokości tonu. Dla uproszczenia, w algorytmie 8.6 nie pokazano obsługi wybranych przypadków brzegowych.

Algorytm 8.6 Wyznaczanie segmentu oraz ontologii LowerTadpolePitchExcursion. Algorytm indukcyjny.

LowerTadpoleStylizer.restylize(e_1:NormalizedPitchExcursion, e_2:NormalizedPitchExcursion):LowerTadpolePitchExcursion
Wejście e_1: etykieta pierwszego segmentu w sylabie
Wejście e_2: etykieta drugiego segmentu w sylabie
Wyjście etykieta anotacji ITT odpowiadająca etykietom wejściowym
LowerTadpolePitchExcursion e
if e_1.SLOPE$*e_2$.SLOPE$\leqslant 0$ **then**
  if $|e_1.\text{SLOPE}| < \alpha \vee |e_1.\text{VOICINGDUR}| < \beta$ **then**
    e_1.HARMONICITY$\leftarrow e_1$.VOICINGDUR$\leftarrow e_1$.SLOPE$\leftarrow 0$
  end if
  if $|e_2.\text{SLOPE}| < \alpha \vee |e_2.\text{VOICINGDUR}| < \beta$ **then**
    e_2.HARMONICITY$\leftarrow e_2$.VOICINGDUR$\leftarrow e_2$.SLOPE$\leftarrow 0$
  end if
end if
if e_1.SLOPE$*e_2$.SLOPE$\geqslant 0$ **then**
  $v \leftarrow e_1$.VOICING$+e_2$.VOICING
  $a \leftarrow e_1$.HARMONICITY$*e_1$.SLOPE$+e_2$.HARMONICITY$*e_2$.SLOPE
  $a \leftarrow a/(e_1$.HARMONICITY$+e_2$.HARMONICITY$)$
  if $a > \alpha \wedge v > \beta$ **then**
    e.ACC_CHANGE$\leftarrow v * a$
  else
    e.ACC_CHANGE$\leftarrow 0$
  end if
else
  if e_1.SLOPE>0 **then**
    e.ACC_CHANGE$\leftarrow e_2$.SLOPE$*e_2$.VOICING
    e.DEC_CHANGE$\leftarrow e_1$.SLOPE$*e_1$.VOICING
  else
    e.ACC_CHANGE$\leftarrow e_1$.SLOPE$*e_1$.VOICING
    e.DEC_CHANGE$\leftarrow e_2$.SLOPE$*e_2$.VOICING
  end if
end if
if e.ACC_CHANGE$\neq 0 \vee e$.DEC_CHANGE$\neq 0$ **then**
  e.PITCH$\leftarrow e_1$.PITCH
else
  $b \leftarrow e_1$.HARMONICITY$/(e_1$.HARMONICITY$+e_2$.HARMONICITY$)$
  e.PITCH$\leftarrow b * e_1$.PITCH$+(1-b) * e_2$.PITCH
end if
return e

8.3.2 UpperTadpoleStylizer

Agent UpperTadpoleStylizer tworzy warstwę z etykietami UpperTadpolePitchExcursion na podstawie dwóch warstw z etykietami Tone oraz LowerTadpolePitchExcursion. Klasa UpperTadpolePitchExcursion została wywiedziona bezpośrednio, bez rozszerzeń z klasy TadpolePitchE-

xcursion. Ze względu na użycie ontologii Tone pochodzącej z poziomu fonologicznego o agencie UpperTadpoleStylizer mówimy, że jest oparty na algorytmie indukcyjno–dedukcyjnym.

Wstępny algorytm analizy indukcyjno–dedukcyjnej na potrzeby wizualizacji anotacji tonalnej zaprezentował Wypych (2006). Algorytm 8.7 przedstawia aktualny sposób tworzenia ścieżek ontologii UpperTadpolePitchExcursion w granicach segmentu Tone. Istotą algorytmu 8.7 jest eliminowanie z reprezentacji fonetycznej tych wartości, które nie są zgodne z wyidealizowanym przebiegiem oczekiwanym na podstawie reprezentacji fonologicznej. Algorytm 8.7 jest stosowany dla każdego segmentu Tone danego na wejściu agenta UpperTadpoleStylizer.

Algorytm 8.7 Wyznaczanie ścieżki etykietowanej ontologiami UpperTadpolePitchExcursion. Algorytm indukcyjno–dedukcyjny.

1: UpperTadpoleStylizer.restylize(A:Annotation,s:Segment,p:Path)
Wejście $A = (S, a)$: anotacja wejściowa/wyjściowa
Wejście $s \in S$: segment etykietowany ontologią Tone
Wejście $p \in \bowtie S$: ścieżka etykietowana ontologiami LowerTadpolePitchExcursion
Wyjście A zawiera ścieżkę etykietowaną ontologiami UpperTadpolePitchExcursion
2: **for** $i = 0$ **to** $|p| - 1$ **do**
3: UpperTadpolePitchExcursion u, w
4: **if** $i > 0$ **then**
5: $u_P \leftarrow u$.PITCH
6: $w_P \leftarrow w$.PITCH
7: **if** (isRising($\overline{s}$)$\wedge u_P < w_P$)$\vee$(isFalling($\overline{s}$)$\wedge u_P > w_P$)$\vee$(isLevel($\overline{s}$)$\wedge u_P \neq w_P$) **then**
8: u.PITCH= w.PITCH
9: **else**
10: u.PITCH= $p[i]$.PITCH
11: **end if**
12: **else**
13: u.PITCH= $\overline{p[0]}$.PITCH
14: **end if**
15: **if** ($i = 0 \wedge \neg$isWeakPrenuclear($\overline{s}$))$\vee$($i = |p| - 1 \wedge$isNuclear($\overline{s}$)) **then**
16: u.ACC_CHANGE = $\overline{p[i]}$.ACC_CHANGE
17: u.DEC_CHANGE = $\overline{p[i]}$.DEC_CHANGE
18: **else**
19: u.ACC_CHANGE = 0
20: u.DEC_CHANGE = 0
21: **end if**
22: addSegment($A, \overleftarrow{p[i]}, \overrightarrow{p[i]}, u$)
23: $w \leftarrow u$
24: **end for**

Parametry oraz funkcje stosowane w algorytmie 8.7:

- addSegment(A:Annotation,b:Anchor,f:Anchor,o:Ontology) — dodaje do anotacji A segment o kotwicach b i f oraz etykiecie o.
- isWeakPrenuclear(e:Tone):Boolean — prawdziwe wtedy i tylko wtedy gdy e jest tonem słabym.
- isNuclear(e:Tone):Boolean — prawdziwe wtedy i tylko wtedy gdy e jest tonem rdzennym.

- isRising(e:Tone):Boolean — prawdziwe wtedy i tylko wtedy gdy e jest tonem rosnącym.
- isFalling(e:Tone):Boolean — prawdziwe wtedy i tylko wtedy gdy e jest tonem opadającym.
- isLevel(e:Tone):Boolean — prawdziwe wtedy i tylko wtedy gdy e jest tonem równym.

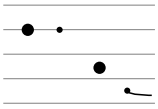
Warstwa etykietowana ontologiami Tone powstaje w wyniku fonologicznej analizy tonalnej (por. rozdział 9).

8.3.3 PostscriptTadpole

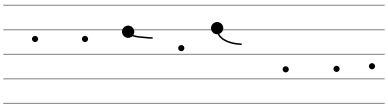
Agent PostscriptTadpole zapisuje ścieżkę segmentów etykietowanych ontologiami UpperTadpolePitchExcursion w postaci kodu źródłowego w języku PostScript.

Algorytm agenta PostscriptTadpole oparto na podstawianiu liniowo przeskalowanych wartości atrybutów ontologii UpperTadpolePitchExcursion we wcześniej przygotowanym szablonie kodu źródłowego PostScript. W stosunku do oryginału (por. rysunek 4.2 na stronie 46) zmieniono konwencje dotyczące linii odniesienia. W oryginalnej ITT występują dwie linie poziome oznaczające odpowiednio minimum oraz maksimum zakresu fonacji modelnej analizowanego głosu. W proponowanej transkrypcji rysowane jest pięć poziomych linii reprezentujących wartości μ, $\mu \pm \sigma$, $\mu \pm 2\sigma$ gdzie μ jest średnią a σ odchyleniem standardowym wysokości tonu mówcy.

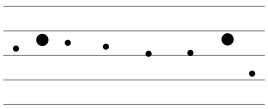
W dalszej części sekcji pokazano przykładowe ITT otrzymane za pomocą proponowanego układu. Analizowany materiał dźwiękowy pochodzi z korpusu PoInt (Karpiński 2002).



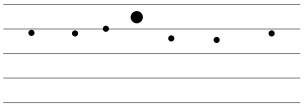
pozytywne



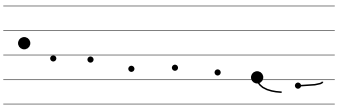
chociaż głowy nie dam za to



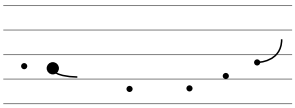
i role się odwróciły



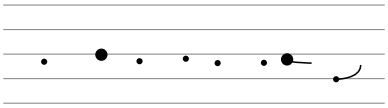
nie angażując się w to



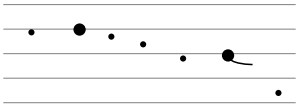
tutaj to już będzie gorzej



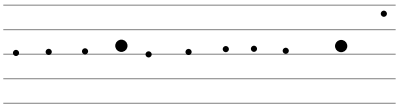
to nie wiem co to jest



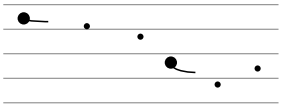
mam dosyć dużo znajomych



nie mają takich barier



i kiedy jadę do innego miasta



tam się spotkałyśmy

8.4 Wyniki

W bieżącej sekcji przedstawimy statystyki ontologii typu NormalizedPitchExcursion wykonane na podzbiorach korpusów mowy PoInt (Karpiński 2002) oraz Babel (Gubrynowicz 1999). Zgodnie z założeniami przyjętymi na początku rozdziału nie badamy odwracalności proponowanej anotacji fonetycznej. Wyliczenia oraz wizualizacje prezentowane w bieżącej sekcji wykonano za pomocą skryptów napisanych w języku R (R Development Core Team 2010). Skrypty uruchomiono w środowisku programowym R w wersji 2.11.0 z kwietnia 2010 roku.

Tabela 8.3: Korpusy do badania rozkładów ontologii StandarizedPitchExcursion.

	PoS1	BaS1	BaS1F	BaS1M
Korpus źródłowy	PoInt	Babel	BaS1	BaS1
Rodzaj mowy	spontaniczna	czytana	czytana	czytana
Liczba mówców	3	47	23	24
Liczba fraz intonacyjnych	437	4159	2128	2031
Liczba sylab fonologicznych	1908	18574	9497	9077
Liczba instancji ontologii	3816	37148	18994	18154
Wielkość próby	2000	2000	2000	2000

W tabeli 8.3 przedstawiono oznaczenia oraz podstawowe informacje na temat korpusów mowy stosowanych w bieżącej sekcji. Korpus PoS1 powstał poprzez wybór z korpusu PoInt nagrań wywiadów przeprowadzonych z trzema osobami o identyfikatorach: „mawa", „joju" oraz „pano" (2 kobiety, 1 mężczyzna) wykonanych w UAM w Poznaniu. Korpus BaS1 powstał poprzez wybór z korpusu Babel nagrań fraz liczbowych (1-5 cyfrowych) wypowiedzianych przez głosy o identyfikatorach: „am", „bk", „da", „dg", „dj", „dt", „ej", „ga", „gm", „gt", „ja", „jr", „ju", „jz", „ka", „kb", „kd", „ki", „kj", „km", „kp", „kr", „kw", „ky", „kz", „ma", „mj", „mm", „mr", „nk", „nw", „om", „pj", „pz", „ro", „sa", „sh", „sk", „sl", „sm", „ss", „tb", „wd", „we", „ww", „zj" oraz „zk" (23 kobiety, 24 mężczyzn) wykonanych w IPPT PAN w Warszawie. Korpusy BaS1F oraz BaS1M zawierają wszystkie nagrania odpowiednio głosów męskich oraz głosów kobiecych z korpusu BaS1. W korpusach mowy przyjęto częstotliwość próbkowania równą 16 kHz. Wszelkie dalsze analizy statystyczne wykonano na próbach losowych o wielkości 2000 przypadków.

8.4.1 Rozkłady jednowymiarowe

Na rycinie 8.4 pokazano jednowymiarowe empiryczne rozkłady gęstości dla pięciu zmiennych (atrybutów) ontologii StandarizedPitchExcursion na PoS1 oraz BaS1. Empiryczne rozkłady gęstości wykonano metodą KDE (*Kernel Density Estimation*) z użyciem okna Gaussa z szerokością pasma $\sigma = 0.2$. Testy Kołmogorowa–Smirnowa (K–S) dają podstawy do odrzucenia (przy $p = 0.01$) hipotez o równości rozkładów zmiennych VOICING, SLOPE, VOICINGDUR i DISPERSION w PoS1 oraz BaS1. Testy K–S nie dają podstaw do odrzucenia (przy $p = 0.01$) hipotez o równości rozkładów zmiennej HARMONICITY w PoS1 oraz BaS1.

Na rycinie 8.5 pokazano jednowymiarowe empiryczne rozkłady gęstości dla pięciu zmiennych (atrybutów) ontologii StandarizedPitchExcursion na BaS1F oraz BaS1M. Empiryczne rozkłady gęstości wykonano ponownie metodą KDE z użyciem okna Gaussa z szerokością pasma $\sigma = 0.2$. Testy K–S dają podstawy (przy $p = 0.01$) do odrzucenia hipotez o równości rokładów zmiennych PITCH w BaS1F oraz BaS1M. Testy K–S nie dają podstaw do odrzucenia (przy $p = 0.01$) hipotez o równości rozkładów zmiennych SLOPE, VOICINGDUR, HARMONICITY, DISPERSION na BaS1F oraz BaS1M.

Jak wynika z zaprezentowanych analiz statystycznych, w badanym materiale rozkłady jednowymiarowe wykazują większe zróżnicowanie między korpusami niż między płciami w korpusie BaS1.

Tabela 8.4: Macierz korelacji ontologii StandarizedPitchExcursion. Korpus PoS1.

	PITCH	SLOPE	VOICINGDUR	HARMONICITY	DISPERSION
PITCH	1.0000				
SLOPE	0.0502	1.0000			
VOICINGDUR	-0.0333	-0.0541	1.0000		
HARMONICITY	-0.0475	-0.0527	0.2149	1.0000	
DISPERSION	-0.0354	0.0106	-0.1708	-0.2306	1.0000

Tabela 8.5: Macierz korelacji ontologii StandarizedPitchExcursion. Korpus BaS1.

	PITCH	SLOPE	VOICINGDUR	HARMONICITY	DISPERSION
PITCH	1.0000				
SLOPE	0.1087	1.0000			
VOICINGDUR	-0.1085	-0.0499	1.0000		
HARMONICITY	-0.0161	-0.0696	0.3683	1.0000	
DISPERSION	-0.1350	0.0242	-0.2102	-0.0064	1.0000

8.4.2 Analiza zależności liniowych

W tabach 8.4 oraz 8.5 pokazano macierze korelacji zmiennych ontologii StandarizedPitchExcursion na korpusach PoS1 oraz BaS1. W bieżącej pracy przyjmujemy, że korelacja w zakresie $0.0 \leqslant |\rho| < 0.2$) charakteryzuje zmienne *nieskorelowane* a korelacja w zakresie $0.2 \leqslant |\rho| < 0.4$) zmienne *nieznacznie skorelowane*. W korpusie PoS1 nieznacznie skorelowane są zmienne VOICINGDUR i HARMONICITY ($\rho = 0.2149$) oraz HARMONICITY i DISPERSION ($\rho = -0.2306$). W korpusie BaS1 nieznacznie skorelowane są zmienne VOICINGDUR i HARMONICITY ($\rho = 0.3683$) oraz VOICINGDUR i DISPERSION ($\rho = -0.2102$). Pozostałe pary zmiennych w obu korpusach są nieskorelowane.

W tabeli 8.6 przedstawiono wyniki analizy głównych składowych (*Principial Component Analysis*, PCA) na korpusach PoS1 oraz BaS1. W kolumnach umieszczono odchylenia standardowe kolejnych pięciu składowych. W tabelach 8.7 oraz 8.8 podajemy wektory własne otrzymane w wyniku PCA na korpusach PoS1 oraz BaS1. Zauważmy, że $\sigma_5/\sigma_1 \approx 0.3837 \gg 0.1$ dla PoS1 oraz $\sigma_5/\sigma_1 \approx 0.6104 \gg 0.1$ dla BaS1. W związku z powyższym PCA nie daje podstaw do redukcji liczby wymiarów.

Na podstawie przytoczonych wyników przyjmujemy, że proponowana przez nas reprezentacja fonetyczna jest nieredundantna przynajmniej w zakresie zależności liniowych.

Tabela 8.6: Odchylenia standardowe w analizie głównych składowych ontologii StandarizedPitchExcursion. Korpusy PoS1 oraz BaS1.

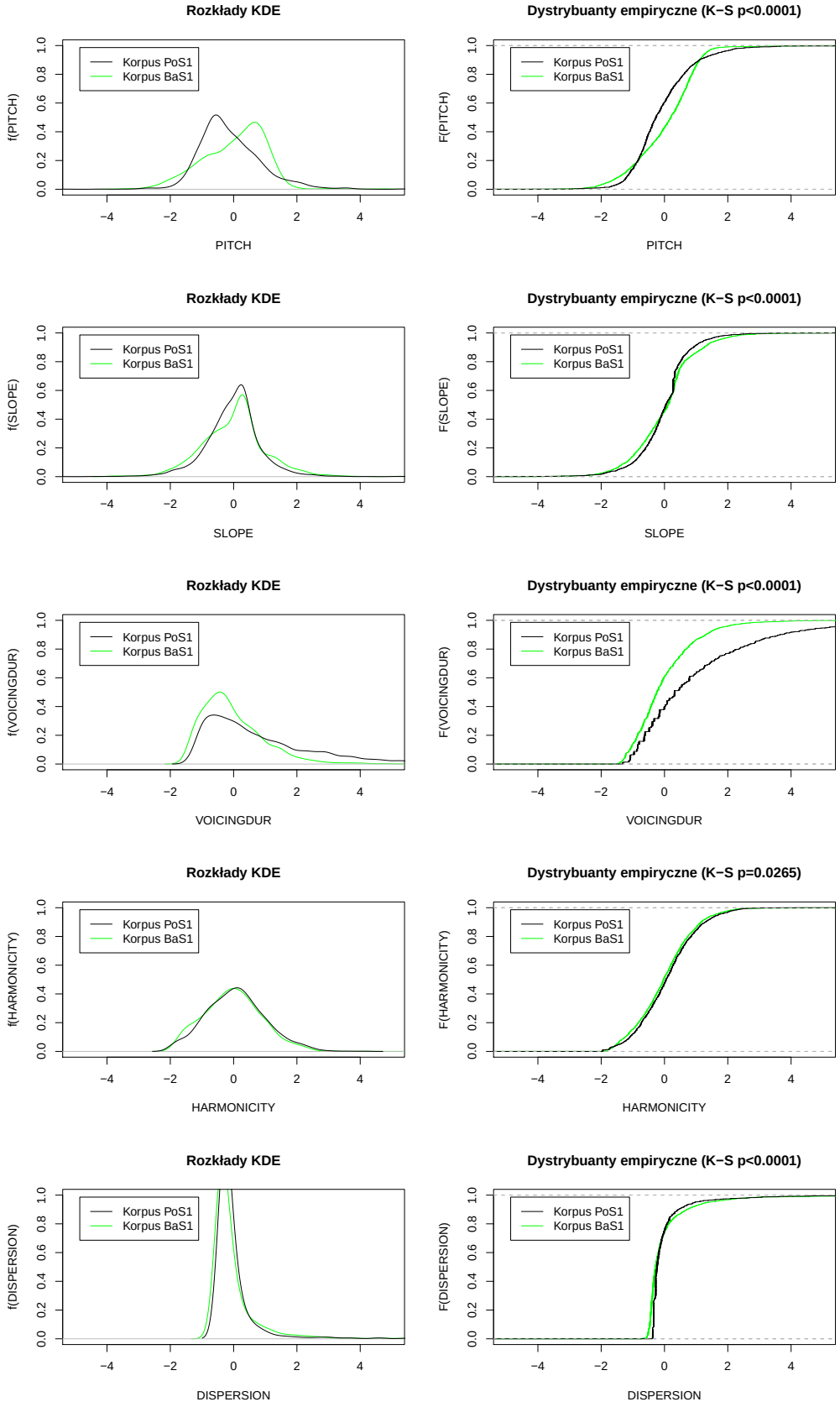
	σ_1	σ_2	σ_3	σ_4	σ_5
PoS1	2.0874	1.0466	0.9813	0.8864	0.8009
BaS1	1.1860	1.0597	0.9722	0.9154	0.7236

Tabela 8.7: Wektory własne w analizie głównych składowych ontologii StandarizedPitchExcursion (pominięto wartości <0.1). Korpus PoS1.

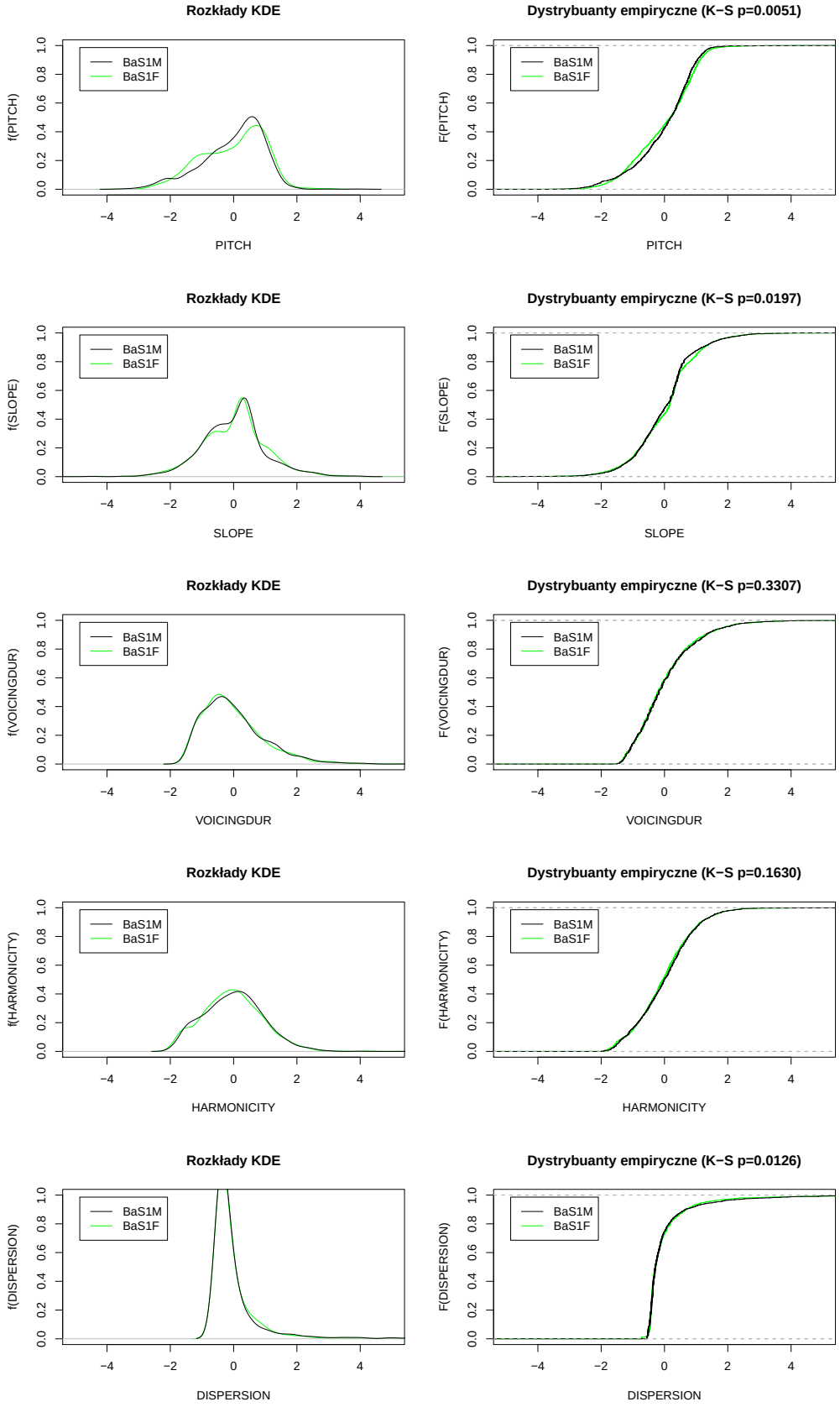
	Skł. 1	Skł. 2	Skł. 3	Skł. 4	Skł. 5
PITCH		0.979		0.166	-0.111
SLOPE		0.154	0.118	-0.974	-0.116
VOICINGDUR	-0.987		0.157		
HARMONICITY	-0.122		-0.698		-0.699
DISPERSION			0.688	0.155	-0.696

Tabela 8.8: Wektory własne w analizie głównych składowych ontologii StandarizedPitchExcursion (pominięto wartości <0.1). Korpusi BaS1.

	Skł. 1	Skł. 2	Skł. 3	Skł. 4	Skł. 5
PITCH	-0.175	0.723	-0.128	0.595	0.277
SLOPE	-0.237	0.418	0.792	-0.365	
VOICINGDUR	0.681	0.111	0.121	-0.225	0.677
HARMONICITY	0.621		0.331	0.474	-0.529
DISPERSION	-0.254	-0.538	0.482	0.487	0.419



Rycina 8.4: Empiryczne rozkłady jednowymiarowe ontologii NormalizedPitchExcursion na korpusach PoS1 oraz BaS1.



Rycina 8.5: Empiryczne rozkłady jednowymiarowe ontologii NormalizedPitchExcursion na korpusach BaS1F oraz BaS1M.

ROZDZIAŁ 9

Anotacja intonacyjna języka polskiego

Fonologiczną anotację tonalną proponowanego układu oparto na gramatyce intonacyjnej, którą zaproponował Jassem (2003a) (por. strona 73). W bieżącym rozdziale opisujemy kroki podjęte w celu zgomadzenia zbioru uczącego dla układu opisanego w rozdziale 10.

9.1 Korpus anotacji wzorcowych

Jassem (2003a) opisał metodę subiektywnej fonologicznej analizy tonalnej przez osobę biegle znającą mowę polską oraz wspomaganą układem fonetycznej analizy tonalnej.

W ramach prezentowanej pracy wykonano (wspólnie z Jassemem) subiektywną fonologiczną analizę tonalną korpusu PoS1 (por. strona 135) będącego podzbiorem korpusu PoInt (Karpiński 2002). Na podstawie wstępnych analizy zgodności anotacji zauważno znacznie rozbieżności w zakresie anotowanej lokalizacji melodii rdzennych rosnących i opadających. W związku z powyższym przyjęto założenie, że w korpusie PoS1 warianty melodii rdzennych rosnących (nurihi, nuriwi, nurilo) i opadających (nufahi, nufawi, nufalo) nie będą odróżniane oraz otrzymają wspólne etykiety, odpowiednio nuri i nufa.

Zaproponowany przez Jassema protokół anotacji obejmował następujące etapy:

1. określenie lokalizacji czasowych oraz transkrypcji ortograficznych fraz intonacyjnych z korpusu PoS1 (wspólnie przez obu anotatorów),
2. określenie anotacji intonacyjnych dla wszystkich fraz intonacyjnych (niezależnie przez każdego z anotatorów),
3. weryfikacja anotacji intonacyjnych (wspólnie przez obu anotatorów).

W pierwszym etapie zlokalizowano 454 frazy intonacyjne, co do których granic byli zgodni obaj anotatorzy. Liczba fraz, którym nadano identyczne anotacje w drugim etapie analizy wyniosła 297. Zgodność anotatorów liczona dla całych fraz wynosiła po drugim etapie 65.4%. W trzecim etapie anotatorzy przeprowadzili ponowną, wspólną analizę 157 fraz, którym nadano niezgodne anotacje w etapie drugim. W przypadku 114 fraz anotatorzy byli w stanie wypracować wspólnie akceptowaną anotację, co podniosło ostateczną zgodność anotatorów do 90.5%. Pozostałe 43 frazy o niezgodnych anotacjach usunięto z korpusu. We wszystkich

etapach prac analizę subiektywną (odsłuchową) wspomagano analizą obiektywną sygnałową oraz fonetyczną z użyciem układów przedstawionych w rozdziałach 7 oraz 8.

Na potrzeby analizy korpusu PoS1 opracowano format WTT, który umożliwia zapis fonologicznej anotacji tonalnej (segmentacja prosta) wraz z sygnałem ortograficznym. Format WTT opiera się na wstawianiu bezpośrednio przed literą pierwszej samogłoski melodii znacznika określonego wyrażeniem regularnym:

$$\texttt{/\{D?(\$Tone)\}/}, \tag{9.1}$$

gdzie `$Tone` jest alternatywą wszystkich identyfikatorów napisowych melodii podanych w tabeli 5.8 na stronie 74. Wystąpienie litery 'D' w znaczniku oznacza melodię odrzuconą na skutek braku zgody między anotatorami po trzecim etapie protokołu anotacji. Pozycja znacznika w sygnale ortograficznym pozawla na jednoznaczne określenie odpowiadającego mu segmentu fonologicznej anotacji tonalnej, niezależnie od lokalizacji granic międzysylabowych. Każdy segment frazy intonacyjenej zapisywany jest w oddzielnej linii. Od początku linii wpisywane są kolejno dwie liczby zmiennopozycyjne oznaczające granice czasowe segmentu (w sekundach). Następnie po spacji wstawiany jest napis reprezentujący transkrypcję ortograficzną oraz melodie frazy intonacyjnej jak opisano powyżej. Dodatkowo w pliku WTT spacje w transkrypcji ortograficznej zamienia się na znaki podkreślenia. Etykieta pusta (ϵ) jest reprezentowana w pliku WTT pojedynczym znakiem podkreślenia. Fragment pliku w opisanym formacie zamieszczono na wydruku 9.1. Format WTT jest wczytywany jako plik etykietowy przez aplikację WaveSurfer w wersji 1.8.5 z listopada 2005 (Sjölander i Beskow 2000), z której koszystano przy analizie korpusu PoS1.

Listing 9.1: Przykładowy fragment pliku WTT. Korpus PoS1, mówca joju.

```
22.5850000 23.3475000 j{wele}est_d{stfa}użo_l{nuri}asów
23.3475000 24.2425000 {wele}i_j{nule}ezior
24.2425000 24.5725000 _
24.5725000 25.3925000 w_kt{Dstfa}órych_są_jag{Dnule}ody
25.3925000 31.8200000 _
31.8200000 32.9700000 {weri}i_jest_na_pr{stfa}awdę_przyj{nufr}emnie
32.9700000 41.5975000 _
41.5975000 42.6725000 m{stfa}ogły_być_wykorzyst{nufa}ane
42.6725000 47.6350000 _
47.6350000 48.0925000 {wele}a_j{nuri}a
```

Ogółem w korpusie PoS1 oznaczono 1117 melodii, w tym 899 melodii akcentowanych (silnych lub rdzennych). Melodie w korpusie PoS1 objęły ogółem 3124 sylaby. Jak wynika z powyższych danych 28.8% sylab rozpoczyna melodię akcentowaną i tym samym ma akcent realny. W tabeli 9.1 przedstawiono rozkład empiryczny melodii w korpusie PoS1.

Na rycinie 9.1 pokazano empiryczne rozkłady długości melodii. Na rycinie 9.2 pokazano analogiczne rozkłady zbiorczo dla melodii nieakcentowanych oraz akcentowanych. Długość melodii mierzymy liczbą sylab fonologicznych obejmowanych przez segment melodii. Do każdego z rozkładów empirycznych dopasowano dwa rozkłady teoretyczne: geometryczny oraz Poissona. Rozkład geometryczny liczono formułą:

$$P(X = k) = (1 - p)^{k-1} p, \tag{9.2}$$

dla $k = 1, 2, \ldots, 13$. Rozkład Poissona liczono formułą:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{9.3}$$

Tabela 9.1: Rozkład empiryczny etykiet intonacyjnych Jassema w korpusie PoS1.

Etykieta	Liczba	Odsetek	Etykieta	Liczba	Odsetek
stfabe	210	18.8%	weleab	28	2.5%
nufa	190	17.0%	striab	26	2.3%
welebe	139	12.4%	stfrab	19	1.7%
nuri	135	12.1%	stfrbe	16	1.4%
stfaab	126	11.3%	stlebe	11	1.0%
nufr	66	5.9%	strfab	11	1.0%
weribe	51	4.6%	stribe	11	1.0%
nule	31	2.8%	strfbe	10	0.9%
stleab	30	2.7%	nurf	7	0.6%

Tabela 9.2: Dopasowanie rozkładów liczby sylab obejmowanych przez najczęstsze etykiety intonacyjne Jassema. Test χ^2 na korpusie PoS1.

Etykieta	Średnia długość	Geometryczny (p-wartość)	Poissona (p-wartość)
stfabe	3.7	<0.0001	<0.0001
nufa	2.3	<0.0001	<0.0001
welebe	1.7	0.9700	0.0006
nuri	1.9	<0.0001	<0.0001
stfaab	3.9	<0.0001	0.0498
nufr	5.2	0.0105	<0.0001
weribe	2.8	<0.0001	0.0123
nule	1.7	0.0043	0.1069
stleab	2.7	0.0129	0.4079
`/^we/`	1.8	0.9515	0.0015
`/^st|^nu/`	3.2	<0.0001	<0.0001

dla $k = 1, 2, \ldots, 13$. W obu przypadkach przyjęto estymatory MLE dla parametrów, tj. odwrotność średniej z próby dla p oraz średnią z próby dla λ.

W tabeli 9.2 przedstawiono p-wartości testów χ^2 dopasowania długości dziewięciu najczęstszych melodii do dwóch rozkładów teoretycznych: geometrycznego oraz Poissona. Wyrażenia regularne `/^we/` oraz `/^st|^nu/` reprezentują odpowiednio zbiór etykiet melodii nieakcentowanych oraz akcentowanych. Przedstawione p-wartości otrzymano w wyniku zastosowania testu χ^2 z dwunastoma punktami swobody. Jak wynika z tabeli 9.2 odpowiedź na pytanie, który z testowanych rozkładów lepiej pasuje do obserwacji zależy od rodzaju melodii. Z dużą pewnością można natomiast stwierdzić, że nie ma podstaw do odrzucenia hipotezy o geometryczności rozkładu długości melodii nieakcentowanych.

9.2 Subkategoryzacja melodii rdzennych

W korpusie anotacji wzorcowych (por. sekcja 9.1) zastosowano zbiorcze etykiety melodii rdzennych: nuri (dla rosnących) oraz nufa (dla opadających). W bieżącej sekcji podjęto problem automatycznej rekonstrukcji etykiet szczegółowych przewidywanych przez gramatykę

Jassema, tj. nurilo, nuriwi, nurihi, nufalo, nufawi oraz nufahi. W celu rekonstrukcji etykiet szczegółowych posłużono się statystycznym kryterium dystynktywności. (Kryteria dystynktywności opisano w sekcji 1.3.)

W bieżącej pracy statystyczne kryterium dystynktywności oparto na metodzie analizy skupień MCLUST (Fraley i Raftery 2002). Rozszerzona implementacja metody MCLUST została opublikowana jako pakiet środowiska R (Fraley i Raftery 2006). W niniejszej pracy użyto środowiska R w wersji 2.11 oraz pakietu MCLUST w wersji 3.4.1.

Istotą MCLUST jest maksymalizacja wartości oczekiwanej (EM) wieloskładnikowego ciągłego modelu gaussowskiego (CDGMM). Prawdopodobieństwo zbioru uczącego wyznaczone przy użyciu modelu CDGMM oznaczonego przez $\mathcal{M}$ ma postać:

$$P_{\mathcal{M}}(\boldsymbol{x}) = \prod_{i=1}^{n} \sum_{k=1}^{G} \tau_k \phi_k(\boldsymbol{x}_i | \mu_k, \Sigma_k), \tag{9.4}$$

gdzie $\boldsymbol{x}$ jest zbiorem uczącym, G liczbą składników $\mathcal{M}$, τ_k prawdopodobieństwem *a-priori* k-tego składnika oraz ϕ_k wielowymiarowym rozkładem normalnym z wektorem średnich μ_k oraz macierzą kowariancji Σ_k (Fraley i Raftery 2006, 52). Zakłada się, że ustalone w wyniku uczenia bez nadzoru składniki CDGMM (wielowymiarowe rozkłady normalne) odpowiadają skupieniom danych w zbiorze uczącym. Więcej na temat analizy skupień metodą EM piszą np. Krzyśko i inni (2008, 358).

W metodzie MCLUST macierz kowariancji jest poddawana diagonalizacji:

$$\Sigma_k = \lambda_k D_k A_k D_k^T, \tag{9.5}$$

gdzie D_k jest ortogonalną macierzą wektorów własnych, $\lambda_k \in \mathbb{R}$ a $\lambda_k A_k$ jest diagonalną macierzą wartości własnych (Fraley i Raftery 2006, 52). Modele CDGMM podzielono na rodziny o trzyliterowych oznaczeniach reprezentujących ograniczenia nakładane na czynniki λ_k (E – wspólne, V – niezależne), A_k (E – wspólne, V – niezależne) oraz D_k (I – identycznościowe, E – wspólne, V – niezależne). Np. model z rodziny VEI ma niezależnie wyznaczane współczynniki λ_k (V), wspólną dla każdego k macierz $A = A_k$ (E) oraz identycznościową macierz $I = D_k$ (I).

Metoda MCLUST dla zadanych: liczby skupień G, rodziny modeli F oraz zbioru uczącego $\boldsymbol{x}$ tworzy model $\mathcal{M}$ w następujących krokach:

1. Określenie początkowego podziału $\boldsymbol{x}$ na G części metodą grupowania hierarchicznego (HC – *Hierarchical Clustering*).
2. Powtarzane do momentu osiągnięcia zbieżności wartości liczonej wzorem 9.4:
   - 2.1 krok „E”: obliczenie macierzy Z o rozmiarze $n \times G$ reprezentującej nieostrą przynależność elementów zbioru uczącego do poszczególnych skupień,
   - 2.2 krok „M”: estymacja parametrów modelu na podstawie macierzy Z.

Dla każdego modelu $\mathcal{M}$ oraz zbioru uczącego $\boldsymbol{x}$ jest wyliczany współczynnik BIC (*Bayesian Information Criterion*) (Schwarz 1978) określony następująco:

$$BIC(\mathcal{M}, \boldsymbol{x}) = 2 \log P_{\mathcal{M}}(\boldsymbol{x}) - m \log(|\boldsymbol{x}|), \tag{9.6}$$

gdzie m jest liczbą wolnych parametrów w modelu $\mathcal{M}$. Współczynnik BIC jest miarą jakości modelu, w której oprócz zgodności modelu z danymi brana jest pod uwagę złożoność (liczba wolnych parametrów) modelu.

Metoda MCLUST jest domyślnie wykonywana dla $G \in \{1, 2, .., 9\}$ skupień oraz rodzin modeli $F \in \{$EII, VII, EEI, VEI, EVI, VVI, EEE, EEV, VEV, VVV$\}$ (w sumie 90 modeli). Za wynik analizy skupień przyjmowany jest model o maksymalnej mierze BIC spośród badanych modeli.

9.3 Uczenie i wyniki

Na korpusie PoS1 wykonano obiektywną fonetyczną analizę tonalną za pomocą układu opisanego w rozdziale 8. W otrzymanych anotacjach zebrano wszystkie ścieżki z etykietami LowerTadpolePitchExcursion obejmowane przez segmenty melodii o etykietach nuri oraz nufa. Powstałe zbiory ścieżek oznaczamy przez $PoS1_{nuri}$ oraz $PoS1_{nufa}$.

Określmy funkcję g, która dla danej ścieżki p etykietowanej ontologiami LowerTadpolePitchExcursion zwraca podzbiór liczb rzeczywistych określony następująco:

$$\begin{aligned} g(p) \;=\; & \bigcup_j \{\overline{p[j]}.\text{PITCH}\} \\ & \cup \bigcup_j \{\overline{p[j]}.\text{PITCH} + \overline{p[j]}.\text{ACC\_CHANGE}\} \\ & \cup \bigcup_j \{\overline{p[j]}.\text{PITCH} + \overline{p[j]}.\text{ACC\_CHANGE} + \overline{p[j]}.\text{DEC\_CHENGE}\}. \end{aligned}$$

Określmy funkcję h następująco:

$$h(p) = (\min(g(p)), \max(g(p)))\,. \tag{9.7}$$

Zbiory uczące $PoS1^h_{nuri}$ oraz $PoS1^h_{nufa}$ definiujemy korzystając z funkcji h:

$$PoS1^h_{\bullet} = \{h(p) : p \in PoS1_{\bullet}\}. \tag{9.8}$$

Ryciny 9.3 oraz 9.4 przedstawiają wyniki metody MCLUST na zbiorach uczących $PoS1^h_{nuri}$ oraz $PoS1^h_{nufa}$. Fragmenty zatytułowane „Klasyfikacja” oraz „Wykres gęstości modelu” dotyczą modelu o maksymalnej wartości BIC. W przypadku zbioru $PoS1^h_{nuri}$ maksymalną wartość $BIC \approx -734$ otrzymano dla 3-składnikowego modelu z rodziny EEE. W przypadku zbioru $PoS1^h_{nufa}$ maksymalną wartość $BIC \approx -876$ otrzymano dla 4-składnikowego modelu z rodziny EEV.

Jak można zauważyć na rycinie 9.4 jedno ze skupień o wartości średniej w przybliżeniu równej $(4, 3)$ zawiera zaledwie trzy elementy zbioru uczącego. Jednocześnie wykazana liczba skupień melodii rdzennych opadających (4) jest niezgodna z przewidywaniami modelu. W związku z powyższym przeprowadzono dodatkowy eksperyment. Przygotowano zbiór uczący $PoS1N^h_{nufa}$, który jest sumą zbioru $PoS1^h_{nufa}$ oraz 10 punktów wylosowanych z dwuwymiarowym rozkładem równomiernym w przedziałach $[-4; 4]$. Wprowadzenie punktów losowych ma zmniejszyć wpływ elementów nietypowych w zbiorze $PoS1^h_{nufa}$ na wyniki MCLUST (Fraley i Raftery 2006, 18). Oczekuje się, że wprowadzone punkty wraz elementami nietypowymi zostaną zaliczone do (dodatkowego) skupienia o znacznej wariancji. Rycina 9.5 przedstawia wyniki metody MCLUST na $PoS1N^h_{nufa}$. Tym razem maksymalną wartość $BIC \approx -1036$ otrzymano dla 3-składnikowego modelu z rodziny VVV. Należy przy tym dać dwa zastrzeżenia: 1) skupienie o największej wariancji reprezentuje zakłócenia próby, 2) wybór rodziny VVV wynika prawdopodobnie z dużej wariancji skupienia reprezentującego elementy nietypowe.

Tabela 9.3: Etykiety melodii rdzennych otrzymanych metodą MCLUST.

Przedrostek	i	$M[i]$	$B[0]$	$\boldsymbol{v}_i$	Kąt	Etykieta
`'nuri'`	0	$(2.04, -0.93)$	0.62	$(1.42, -1.56)$	1.53	`'nuriwi'`
`'nuri'`	1	$(0.10, -0.78)$	0.62	$(-0.52, -1.40)$	0.43	`'nurilo'`
`'nuri'`	2	$(2.18, 1.13)$	0.62	$(1.56, 0.50)$	2.67	`'nurihi'`
`'nufa'`	1	$(0.66, -0.84)$	-0.24	$(0.91, -0.59)$	1.78	`'nufawi'`
`'nufa'`	2	$(-0.18, -0.62)$	-0.24	$(0.06, -0.37)$	0.94	`'nufalo'`

Przyporządkowanie skupień do etykiet melodii w gramatyce intonacyjnej wykonujemy w oparciu o przesłanki geometryczne opisane poniżej. Oznaczmy przez M macierz o rozmiarach $m \times 2$, w której wiersz $M[i]$ reprezentuje wektor średnich rozkładu normalnego dla skupienia i. Naszym celem jest zdefiniowanie funkcji klasyfikacyjnej $g_M : \{0, 1, .., m-1\} \mapsto \{0, 1, 2\}$, która dla indeksu wiersza (skupienia) w macierzy M zwraca indeks przyrostka etykiety melodii w trójce uporządkowanej $L =$ (`'lo'`, `'wi'`, `'hi'`) (od wyrazów: „*low*", „*wide*" oraz „*high*").

Niech punkt A będzie środkiem ciężkości macierzy M określonym następująco:

$$A = \frac{1}{m} \sum_{i=0}^{m-1} M[i]. \tag{9.9}$$

Oznaczmy przez B punkt powstały w wyniku rzutowania punktu A na prostą $y = x$ w kierunku prostopadłym do tej prostej. Rozwiązując elementarny układ równań, który wynika z powyższego opisu, dostajemy:

$$B[0] = B[1] = \frac{A[0] + A[1]}{2}. \tag{9.10}$$

Ostatecznie przyjmujemy następującą definicję funkcji klasyfikacyjnej:

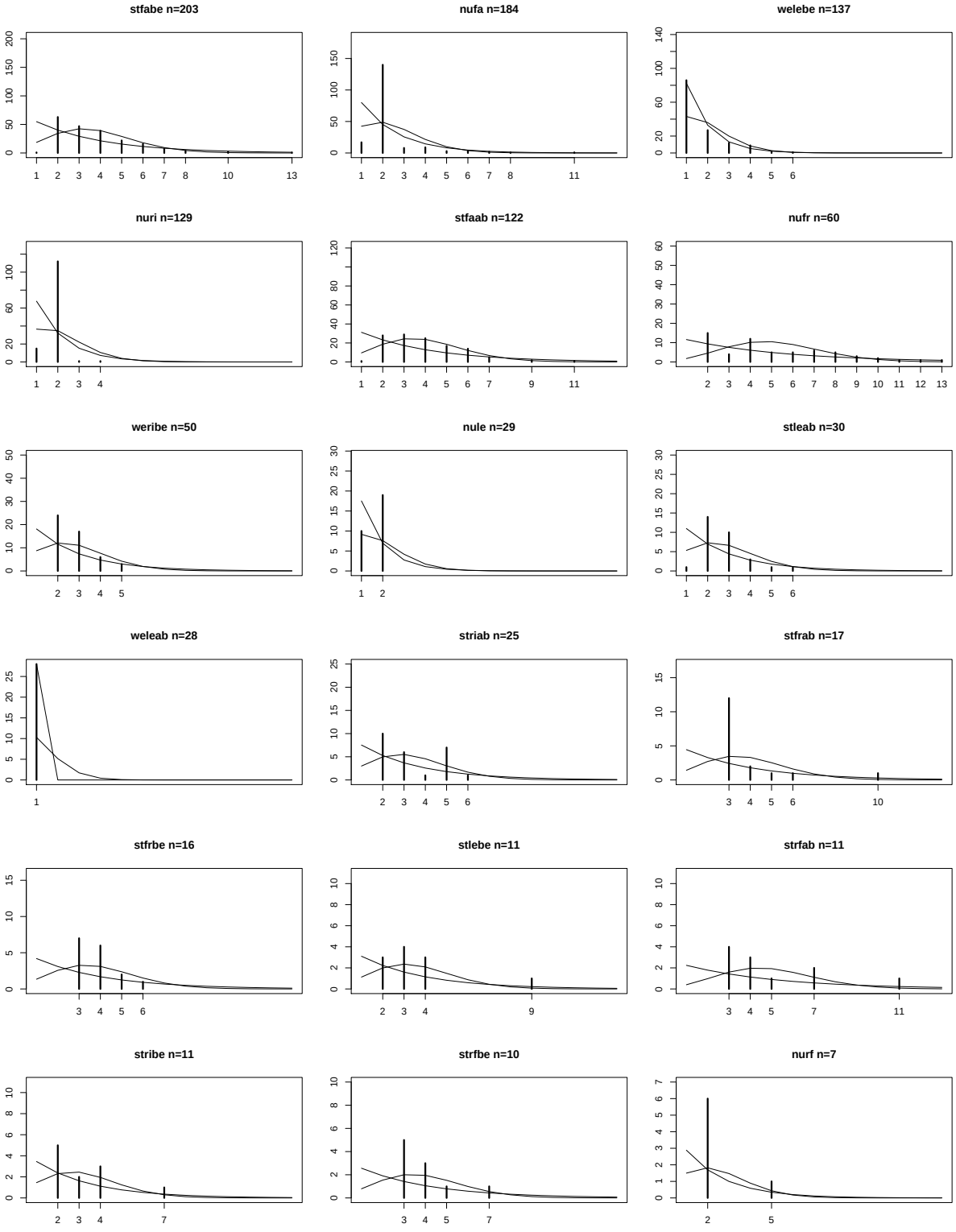
$$g_M(i) = \left\lceil \frac{3}{\pi} \arccos\left(\frac{\boldsymbol{r} \cdot \boldsymbol{v}_i}{||\boldsymbol{r}||||\boldsymbol{v}_i||} \right) \right\rceil, \tag{9.11}$$

gdzie $\boldsymbol{r} = (-1, -1)$ oraz $\boldsymbol{v}_i = M[i] - B$. Zgodnie z równaniem 9.11 wartość funkcji dla i-tego skupienia wynika w prosty sposób z kąta zawartego między wektorem referencyjnym $\boldsymbol{r}$ a wektorem łączącym punkt B ze środkiem i-tego skupienia.

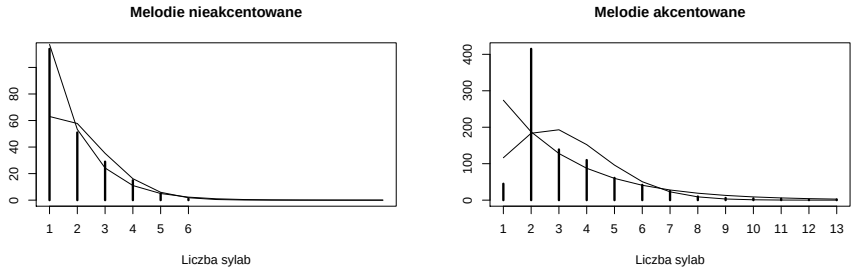
W tabeli 9.3 zawarto przybliżone pośrednie oraz końcowe wyniki przyporządkowania etykiet do skupień.

Na podstawie przeprowadzonych analiz stwierdzono występowanie w korpusie PoS1 trzech rodzajów melodii rosnących oraz dwóch rodzajów melodii opadających. Skupienia melodii rosnących są zgodne z przewidywaniami modelu Jassema (`'nurilo'`, `'nuriwi'`, `'nurihi'`). W przypadku melodii opadających wyniki potwierdzają istnienie skupienia `'nufalo'` oraz `'nufawi'`. Nie stwierdzono w badanym materiale dostatecznie licznej reprezentacji melodii zgodnych z fonetycznym opisem etykiety `'nufahi'`. Liczności poszczególnych melodii wyniosły: 78 (`'nurilo'`), 26 (`'nuriwi'`), 25 (`'nurihi'`), 67 (`'nufalo'`) oraz 110 (`'nufawi'`).

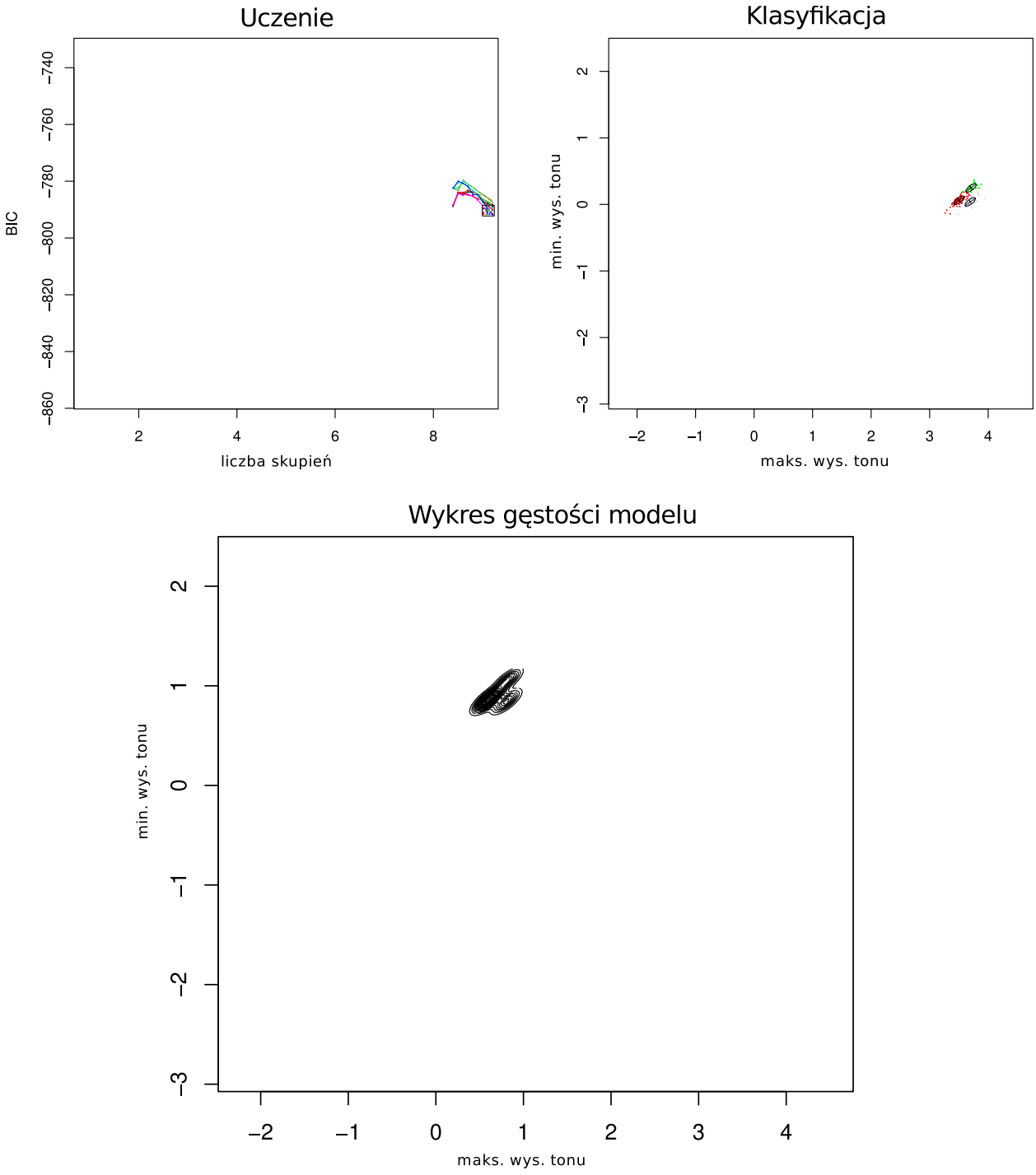
Na koniec sekcji określmy oznaczenia przyjmowane w dalszej części pracy. Niech $\mathcal{M}$ oznacza zbiór nazw melodii anotacji intonacyjnej Jassema wymienionych w tabeli 5.8 na stronie 74, w której melodie `'nuri'` oraz `'nufa'` zastąpiono podkategoriami wprowadzonymi w bieżącej sekcji. Przez $\mathcal{J} \subset \mathcal{M} \times \mathcal{M}$ będziemy oznaczać zbiór par melodii dopuszczanych przez gramatykę Jassema wynikający z wyrażenia regularnego 5.8 na stronie 73.



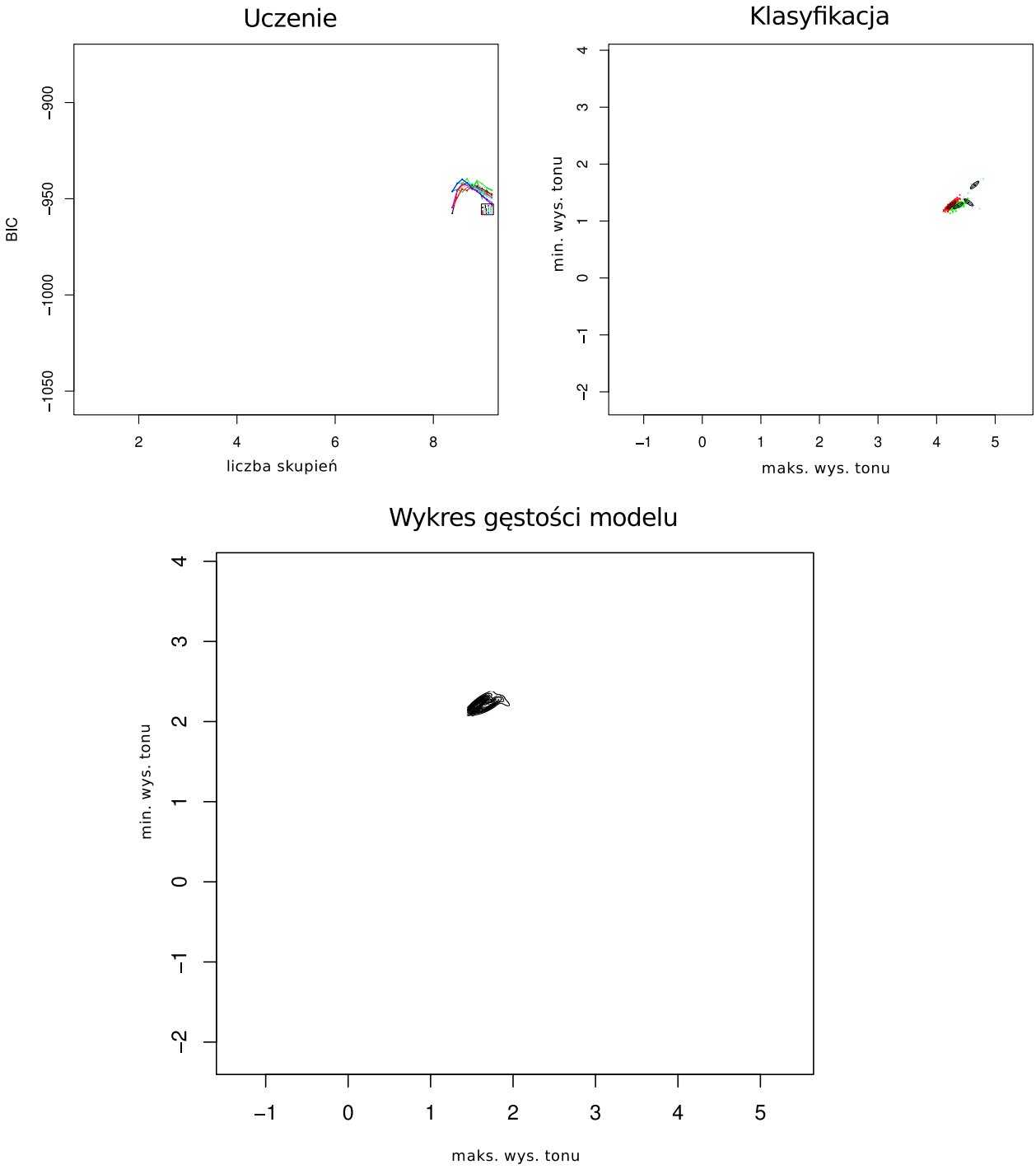
Rycina 9.1: Rozkłady empiryczne oraz teoretyczne (geometryczne i Poissona) długości melodii w korpusie PoS1. Wykresy dla poszczególnych melodii.



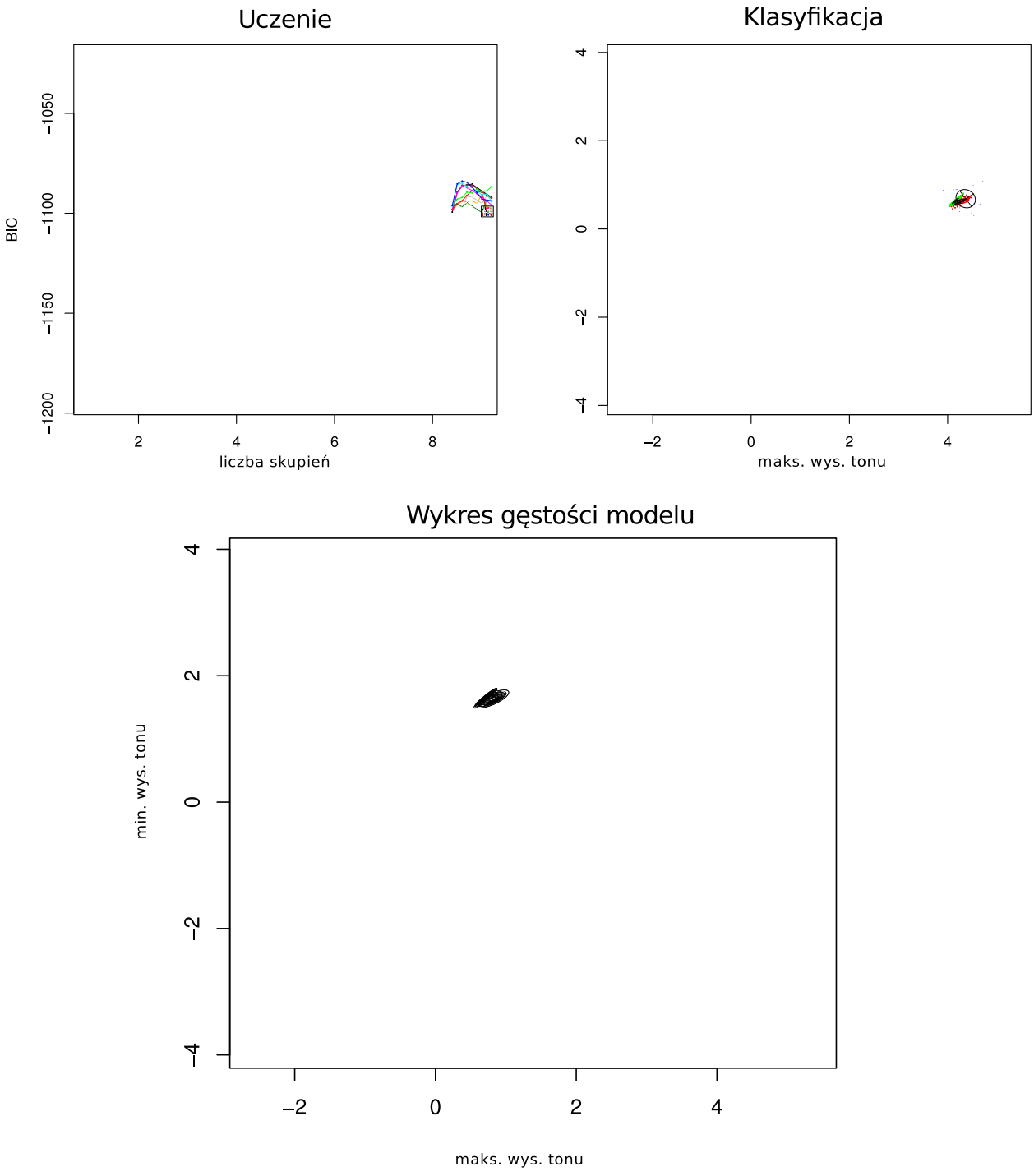
Rycina 9.2: Rozkłady empiryczne oraz teoretyczne (geometryczne i Poissona) długości melodii w korpusie PoS1. Wykresy zbiorcze dla melodii nieakcentowanych oraz akcentowanych.



Rycina 9.3: Wyniki metody MCLUST dla melodii rdzennych rosnących (zbiór uczący $PoS1^{h}_{nuri}$)



Rycina 9.4: Wyniki metody MCLUST dla melodii rdzennych opadających (zbiór uczący $PoS1^{h}_{nufa}$)



Rycina 9.5: Wyniki metody MCLUST dla melodii rdzennych opadających (zbiór uczący $PoS1N^{h}_{nufa}$)

ROZDZIAŁ 10

Układ fonologicznej analizy tonalnej

Zdecydowana większość opisanych w literaturze układów fonologicznej analizy tonalnej opiera się na bezgramatykowej detekcji tonalnych etykiet fonologicznych, która wykonywana jest w trybie off-line na korpusach mowy czytanej kilku mówców. W bieżącej pracy postawiono następujące wymagania dla tworzonego układu fonologicznej analizy tonalnej:

1. zastosowanie anotacji z gramatyką BT,

2. uczenie i pomiar skuteczności układu na licznej grupie mówców (>50),

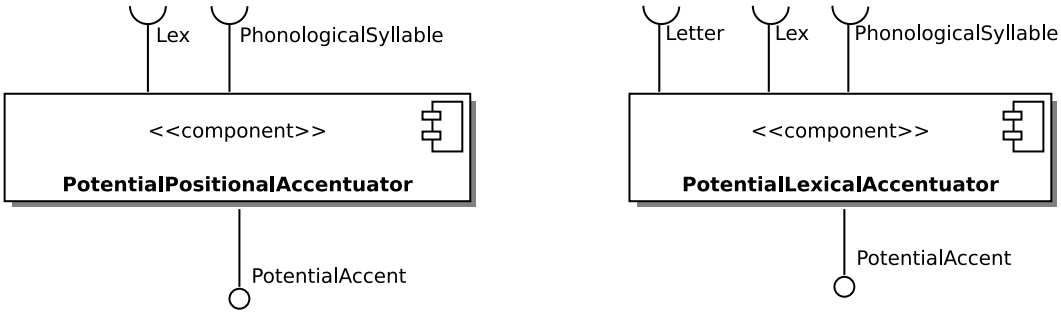
3. uczenie i pomiar skuteczności układu na mowie czytanej oraz spontanicznej,

Danymi wejściowymi proponowanego układu fonologicznej analizy tonalnej jest trójka uporządkowana: 1) sygnał akustyczny, 2) sygnał ortograficzny oraz 3) napisowy identyfikator mówcy. Przyjmuje się, że oba sygnały wejściowe są zapisem tej samej wypowiedzi. Sygnał ortograficzny pełni funkcję wspomagającą w analizie fonetycznej oraz przy określaniu lokalizacji akcentów potencjalnych. Napisowy identyfikator mówcy jest stosowany w normalizacji na etapie analizy fonetycznej (por. rozdział 8).

W sekcjach 10.1 oraz 10.2 opisano podukłady realizujące fonologiczną analizę tonalną w oparciu o dane dostępne w układzie fonetycznej analizy tonalnej, który opisano w rozdziale 8. W sekcji 10.3 przydstawiono procedurę uczenia oraz wyniki pomiarów skuteczności układu. Układ zintegrowano w środowisku SLOPE z zastosowaniem architektury tablicowej (por. sekcja 6.4).

10.1 Podukład analizy ortograficznej

Celem podukładu analizy ortograficznej w układzie fonologicznej analizy tonalnej jest detekcja akcentu potencjalnego. Ze względu na rodzaj melodii rozpoczynającej się na sylabie z akcentem potencjalnym wyróżniamy: 1) **akcent potencjalny główny** (występowanie melodii silnych i rdzennych) oraz 2) **akcent potencjalny poboczny** (występowanie melodii silnych ale nie rdzennych).



Rycina 10.1: Podukład analizy ortograficznej w układzie fonologicznej analizy tonalnej. Diagram komponentów UML.

W kracie anotacyjnej akcent potencjalny jest reprezentowany przez ontologie PotentialAccent należące do warstwy morfo–syntaktycznej zbioru ontologii SLOPE (por. rycina 6.4 na stronie 98). Układ działa w oparciu o dane pozyskane z sygnału ortograficznego za pośrednictwem agentów podukładu analizy ortograficznej w układzie fonetycznej analizy tonalnej (por. sekcja 8.1).

Na rycinie 10.1 przedstawiono diagram komponentów UML podukładu analizy ortograficznej. W kolejnych podsekcjach opisano komponenty przedstawione na rycinie 10.1.

10.1.1 PotentialPositionalAccentuator

Akcentem pozycyjnym nazywamy akcent potencjalny, w którego detekcji korzysta się z segmentacji oraz typów etykiet segmentów, lecz nie uwzględnia się wartości etykiet segmentów.

W języku polskim akcent potencjalny poboczny jest akcentem pozycyjnym. Akcent potencjalny poboczny przypada na sylabie inicjalnej każdego leksu, który objemuje cztery lub więcej sylab.

Powyższą regułę zaimplementowano w agencie PotentialPositionalAccentuator, który w granicach sylab z akcentem pobocznym wstawia segmenty etykietowane ontologiami SecondaryPotentialAccent.

10.1.2 PotentialLexicalAccentuator

Akcentem leksykalnym nazywamy akcent potencjalny, w którego detekcji korzysta się z segmentacji oraz wartości etykiet segmentów. W języku polskim akcent potencjalny główny jest akcentem pochodzenia leksykalnego. W zależności od wartości etykiet literowych obejmowanych przez leks akcent potencjalny główny może przypadać oksytonicznie (tj. na ostatniej sylabie, np. «osobodzień», «jury», «coupe», «eksmąż»), paroksytonicznie (tj. na przedostatniej sylabie, np. «niedziela», «permutacja», «papier») lub proparoksytonicznie (tj. na przedprzedostatniej sylabie, np. «liceum», «metryka», «znalazłby»). W języku polskim akcent potencjalny główny przypada najczęściej paroksytonicznie.

Agent PotentialLexicalAccentuator tworzy segmenty akcentu leksykalnego (etykieta PrimaryPotentialAccent) analizując wejściową ścieżkę liter, przy uwzględnieniu granic sylab fonologicznych oraz leksów. W agencie PotentialLexicalAccentuator zastosowano algorytm oparty na wiedzy słownikowej, którą zakodowano w postaci specjalizowanych reguł.

Oznaczmy przez $\mathfrak{L}$ zbiór złożony z liter oraz znaku technicznego # (granica leksu). **Regułą akcentuacji leksykalnej** nazywamy dowolną n-tkę uporządkowaną $(C, W, W_0, .., W_{n-3})$, gdzie $C \subset \mathfrak{L}^*$, $W \in \{0,1\}^*$ oraz $W_i \in \mathbb{Z} \times \{0,1\}^*$. Element zbioru $\{0,1\}^*$ (wektor binarny) nazywamy **wzorcem akcentuacyjnym**.

Niech $A = (S, a)$ będzie anotacją warstwową zawierającą warstwy liter, sylab oraz leksów. Przyjmijmy funkcję pomocniczą $h :\bowtie S \mapsto \mathfrak{L}^*$, która dla wejściowej ścieżki liter p zwraca napis powstający przez konkatenację liter w kolejności określonej przez p oraz wstawienie znaku # w miejscu każdej takiej kotwicy p, która jest jednocześnie lewą lub prawą kotwicą pewnego leksu w A. Niech będzie ustalona reguła akcentuacji leksykalnej $(C, W, W_0, .., W_{n-3})$. Jeśli istnieje ścieżka $p \in \bowtie S$ taka, że $h(p) \in C$, to wzorzec akcentuacyjny dla p wyznaczany jest funkcją w określoną następująco:

$$w(p) = \begin{cases} W_i[1] & \text{jeśli } \exists\limits_i W_i[0] = |r| \\ W & \text{w przec. przyp.,} \end{cases} \tag{10.1}$$

gdzie r jest ścieżką sylab fonologicznych taką, że $r \rhd p$. Znając wzorzec akcentuacyjny $w(p)$ oraz ścieżkę sylab fonologicznych r przypisujemy akcent fonologiczny każdemu i tylko takiemu $r[i]$, dla którego zachodzi $w(p)[i] = 1$. W przypadku kolizji reguł pierszeństwo daje się regule o dłuższej ścieżce p.

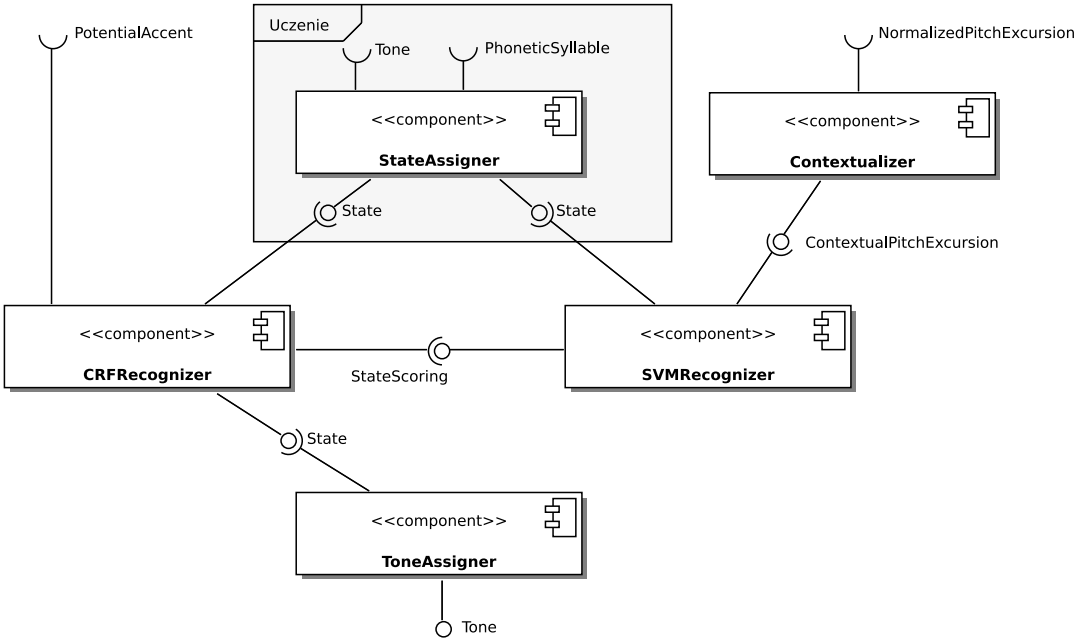
W niniejszej pracy zastosowano ponad 430 reguł akcentuacji leksykalnej, które zaproponował Szczyszek i Wypych (2007). Na wydruku 10.1 pokazano fragmenty kodów źródłowych reguł akcentuacji leksykalnej. W kodach źródłowych na wydruku 10.1 stosuje się konwencje opisane dla reguł transkrypcji fonetycznej (por. sekcja 8.1.5 na stronie 117). W każdej linii pliku źródłowego jest zapisywana co najwyżej jedna reguła akcentuacji leksykalnej. Część C reguły jest umieszczana przed dwukropkiem. Części W oraz W_i reguły są umieszczane w nawiasach kwadratowych oraz oddzielane przecinkami. Zapis części W zaczyna się od dwuznaku '->', po którym występuje wzorzec akcentuacyjny. Zapis części W_i składa się z liczby całkowitej $W_i[0]$, po której występuje dwuznak '->' oraz wzorzec akcentuacyjny.

Listing 10.1: Kody źródłowe wybranych reguł akcentuacji.

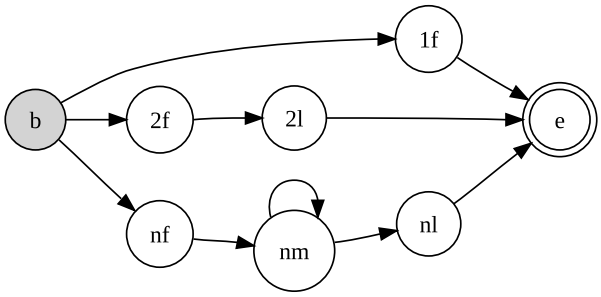
Zbiór reguł akcentuacji jest konwertowany do postaci FST, przy czym część C określa wyrażenie regularne na strumieniu wejściowym a części W oraz W_i zapisywane są w formie tekstowej jako emisja. Po zlokalizowaniu wzorca C w strumieniu wejściowym emisja jest interpretowana zgodnie z formułą 10.1. Zbiór FST odpowiadający regułom akcentuacji leksykalnej jest kompilowany do postaci minimalnego FST za pomocą pakietu FSA6 (van Noord 2009).

10.2 Podukład analizy akustycznej

Podukład analizy akustycznej wykonuje fonologiczną analizę tonalną w oparciu o dane z układu fonetycznej analizy tonalnej (por. rozdział 8) oraz podukładu analizy ortograficznej



Rycina 10.2: Podukład analizy akustycznej w układzie fonologicznej analizy tonalnej. Diagram komponentów UML.



Rycina 10.3: Topologia modelu grafowego pojedynczej melodii.

przedstawionego w poprzedniej sekcji. Na rycinie 10.2 przedstawiono diagram komponentów podukładu analizy akustycznej. W kolejnych podsekcjach zawarto opisy poszczególnych komponentów.

10.2.1 StateAssigner

Agent StateAssigner dla każdego wejściowego segmentu sylaby fonetycznej tworzy segment stanu (ontologia State) na podstawie anotacji wejściowej zawierającej warstwę sylab fonetycznych oraz warstwę melodii. Agent StateAssigner jest używany w trybie uczenia się agentów analizy akustycznej (por. sekcje 10.2.3 oraz 10.2.4).

Ze względu na zastosowanie etykiet intonacyjnych o zmiennej długości (wyrażonej liczbą sylab, por. sekcja 9.1), w bieżącej pracy proponuje się grafowy model przebiegu melodii pokazany na diagramie 10.3. W zależności od liczby sylab obejmowanych przez melodię zastosowanie ma jedno, dwu lub trzystanowa ścieżka modelu.

Przez $\mathcal{S}$ będziemy oznaczać zbiór dwuliterowych nazw stanów (tj. wszystkich poza stanami 'b' oraz 'e') podanych na rycinie 10.3. Przez $\mathcal{O}$ będziemy oznaczać zbiór wszystkich stanów modelu frazy intonacyjnej, tj. $\mathcal{O} = \mathcal{M} \times \mathcal{S}$, gdzie $\mathcal{M}$ określone jest na stronie 146. Obecnie $|\mathcal{O}| = 22 \cdot 6 = 132$. Nazwa stanu modelu frazy intonacyjnej składa się z nazwy melodii, podkreślenia oraz dwuliterowej nazwy stanu modelu melodii, np. `'nule_2f'`, `'nurihi_1f'` lub `'nurihi_nm'`.

10.2.2 Contextualizer

Agent Contextualizer wzbogaca oraz grupuje dane zawarte w ścieżce ontologii NormalizedPitchExcursion. W wyniku działania agenta Contextualizer do anotacji jest wprowadzana ścieżka etykietowana ontologiami ContextualPitchExcursion.

Przyjmijmy, że na wejściu agenta Contextualizer dane są: anotacja kratowa $A = (S, a)$ oraz ścieżka $p \in \bowtie S$ etykietowana ontologiami NormalizedPitchExcursion. Kotwice ścieżki wyjściowej q są określone następująco:

$$\overleftarrow{q[i]} = \overleftarrow{p[2i]} \tag{10.2}$$

oraz

$$\overrightarrow{q[i]} = \overrightarrow{p[2i+1]}. \tag{10.3}$$

Etykiety ścieżki wyjściowej mają typ ContextualPitchExcursion oraz są określone następująco:

$$\overline{q[i]} = \begin{bmatrix} p'[2i-1] - p'[2i-2] \\ p'[2i+0] - p'[2i-1] \\ p'[2i+0] \\ p'[2i+1] \\ p'[2i+2] - p'[2i+1] \\ p'[2i+3] - p'[2i+2] \end{bmatrix}, \tag{10.4}$$

gdzie ciąg wektorów p' jest określony wzorem:

$$p'[j] = \begin{bmatrix} \overline{p[j]}.PITCH \\ \overline{p[j]}.SLOPE \\ \overline{p[j]}.VOICINGDUR \\ \overline{p[j]}.HARMONICITY \\ \overline{p[j]}.DISPERSION \end{bmatrix} \tag{10.5}$$

Dla uproszczenia niniejszego opisu we wzorach 10.2–10.5 pominięto przypadki brzegowe.

Sygnał wynikowy agenta Contextualizer ma 30 wymiarów. W przeciwieństwie do szeregu wcześniejszych prac (por. np. Ananthakrishnan i Narayanan 2008; Rosenberg 2009) nie wykonujemy na bieżącym etapie preselekcji wymiarów na podstawie dodatkowych miar redundancji. Uzasadniamy to trzema faktami: 1) zastosowaniem anotacji fonetycznej, 2) zastosowaniem technik rozpoznawania wzorców dobrze dostosowanych do danych wielowymiarowych (por. sekcja 10.2.3) oraz 3) ryzykiem przedwczesnego odrzucenia potencjalnie przydatnych wymiarów.

10.2.3 SVMRecognizer

Agent SVMRecognizer wyznacza podobieństwo ontologii ContextualPitchExcursion do wzorców stanów modelu melodii, który opisano w sekcji 10.2.1. W agencie SVMRecognizer zastosowano

metodę wektorów nośnych[1] (SVM, *Support Vector Machines*). Agent ma dwa tryby działania: 1) uczenie się wzorców oraz 2) rozpoznawanie wzorców. W trybie uczenia się wzorców następuje estymacja parametrów SVM na podstawie zbioru par złożonych z ontologii: ContextualizedPitchExcursion (obserwacja) oraz State (pożądane wyniki rozpoznawania). W trybie rozpoznawania wzorców agent przyjmuje na wejściu ontologie ContextualizedPitchExcursion (obserwacje) a na wyjściu tworzy ontologię StateScoring określającą *podobieństwo* wejściowej obserwacji do każdego ze wzorców.

Vapnik (1995) zaproponował statystyczną teorię uczenia się, której głównym rezultatem aplikacyjnym jest metoda wektorów nośnych. Poniższy skrócony opis przygotowano na podstawie prac, które opublikowali Chen i inni (2005) oraz Hofmann i inni (2008). W języku polskim metodę wektorów nośnych opisali m.in. Krzyśko i inni (2008).

Niech $\mathcal{X}$ będzie zbiorem wszystkich możliwych obserwacji. W metodzie wektorów nośnych stosuje się funkcje klasyfikacyjne postaci:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i (\Phi(x) \cdot \Phi(x_i)) + b\right), \tag{10.6}$$

gdzie $(x_i, y_i) \in \mathcal{X} \times \{-1, 1\}$, $\alpha_i \in \mathbb{R}$ są parametrami a funkcja $\Phi : \mathcal{X} \mapsto \mathcal{H}$ przenosi obserwacje do **przestrzeni cech**. Przestrzenią cech jest nazywna dowolna przestrzeń liniowa $\mathcal{H}$, na której określono iloczyn skalarny.

Funkcja klasyfikacyjna podana we wzorze 10.6 dzieli przestrzeń cech hiperpłaszczyzną na dwie części odpowiadające etykietom -1 oraz 1. Istotą procedury uczącej jest dobór parametrów α_i w sposób maksymalizujący odległość przykładów w zbiorze uczącym od hiperpłaszczyzny dzielącej przestrzeń cech. W przypadku zbiorów liniowo separowalnych problem ten można utożsamić z następującym problemem programowania kwadratowego:

$$\underset{\alpha_1,\alpha_2,\ldots,\alpha_m}{\operatorname{argmax}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\Phi(x_i) \cdot \Phi(x_j)), \tag{10.7}$$

przy następujących ograniczeniach:

$$\alpha_i \geqslant 0, i = 1, 2, \ldots, m \tag{10.8}$$

oraz

$$\sum_{i=1}^{m} \alpha_i y_i = 0. \tag{10.9}$$

W przypadku, gdy $\alpha_i > 0$ wektor x_i jest nazywany **wektorem nośnym**. Uogólnienie wzoru 10.7 na przypadek zbiorów nieseparowalnych liniowo można znaleźć w dowolnej z polecanych prac na temat SVM (np. Hofmann i inni 2008, 21).

Funkcja Φ reprezentuje dowolne przekształcenie (w szczególności nieliniowe) elementów zbioru $\mathcal{X}$ do przestrzeni cech, mające na celu polepszenie separowalności liniowej obserwacji należących do różnych klas. Określmy funkcję $k : \mathcal{X}^2 \mapsto \mathbb{R}$ następująco:

$$k(x, x') = \Phi(x) \cdot \Phi(x'). \tag{10.10}$$

Funkcja k jest nazywana **funkcją jądra**. Znając funkcję jądra możemy zrezygnować z obliczania funkcji Φ zachowując w mocy wyrażenia 10.6 oraz 10.7. Jest to przydatne w sytuacjach, gdy postać Φ jest nieznana lub gdy wartość Φ nie jest możliwa do obliczenia (np. ze względu na nieskończoną liczbę wymiarów przestrzeni $\mathcal{H}$).

[1] W rozdziale przyjmujemy polską terminologię, którą zaproponowali Krzyśko i inni (2008).

W bieżącej pracy zastosowano liniową funkcję jądra postaci:

$$k(x, x') = x \cdot x', \tag{10.11}$$

oraz radialną funkcję bazową (RBF, *Radial Basis Function*) jądra postaci:

$$k(x, x') = e^{-\gamma||x-x'||^2}, \tag{10.12}$$

gdzie $\gamma > 0$ jest parametrem wolnym.

Agent SVMRecognizer zawiera $|\mathcal{O}|$ (obecnie 132) klasyfikatorów binarnych SVM. Zbiór ucznący klasyfikatora K_o, $o \in \mathcal{O}$ zbudowano w oparciu o zasadę „jeden przeciwko wszystkim" (*1 vs. all*). Zgodnie z tą zasadą klasyfikator K_o jest uczony odróżniania obserwacji etykietowanych jako o od obserwacji o etykietach $o' \in \mathcal{O}$ takich, że $o' \neq o$.

Dla każdego segmentu s etykietowanego ontologią klasy ContextualPitchExcursion, agent SVMRecognizer tworzy segment s' oparty na kotwicach $\overleftarrow{s}$ oraz $\overrightarrow{s}$. Należąca do klasy StateScoring etykieta segmentu s' reprezentuje funkcję $h : \mathcal{O} \mapsto \mathbb{R}$ taką, że:

$$h(o) = K_o(\overline{s}). \tag{10.13}$$

Implementację agenta SVMRecognizer oparto na bibliotece LIBSVM (Chang i Lin 2001) w wersji 3.0 z grudnia 2010.

10.2.4 CRFRecognizer

Agent CRFRecognizer dla danej warstwy StateScoring oraz segmentów PotentialAccent wyznacza warstwę State zgodnie z ograniczeniami nakładanymi przez model melodii opisany w sekcji 10.2.1. W agencie CRFRecognizer zastosowano metodę warunkowych pól losowych (CRF — *Conditional Random Field*), którą zaproponowali Lafferty i inni (2001). W dalszej części sekcji ontologie StateScoring oraz PotentialAccent będziemy nazywali **obserwacjami** a ontologie State **oznaczeniami**.

Agent CRFRecignizer ma dwa tryby działania: 1) uczenie się wzorców oraz 2) rozpoznawanie wzorców. W trybie uczenia się wzorców następuje estymacja parametrów CRF na podstawie obserwacji oraz oznaczeń. W trybie rozpoznawania, na podstawie obserwacji tworzona jest warstwa oznaczeń.

W ostatnich latach opublikowano kilka obiecujących rezultatów fonologicznej analizy tonalnej z użyciem CRF (por. Sridhar i inni 2008; Levow 2008). Hoefel i Elkan (2008), na przykładzie problemu OCR (*Optical Character Recognition*) pokazali, że łącząc SVM oraz CRF można uzyskać model o wyższej skuteczności niż każdy z modeli składowych z osobna. W bieżącej pracy proponujemy realizację indukcyjno–dedukcyjnej fonologicznej analizy tonalnej w oparciu o łączony model SVM/CRF. Szersze opisy CRF opublikowali m.in. Walach (2004) oraz Sutton i McCallum (2006).

Niech $\boldsymbol{X}$ oraz $\boldsymbol{Y}$ będą zmiennymi losowymi odpowiednio nad ciągami obserwacji oraz oznaczeń. Niech będzie dany graf $G = (V, E)$ taki, że $\boldsymbol{Y} = (\boldsymbol{Y}_v)_{v \in V}$. Parę $(\boldsymbol{X}, \boldsymbol{Y})$ nazywamy **warunkowym polem losowym** (CRF) w przypadku, gdy:

$$p(\boldsymbol{Y}_v | \boldsymbol{X}, \boldsymbol{Y}_w, w \neq v) = p(\boldsymbol{Y}_v | \boldsymbol{X}, \boldsymbol{Y}_w, w \sim v), \tag{10.14}$$

gdzie $w \sim v$ oznacza, że w oraz v są sąsiadami w G (Lafferty i inni 2001).

Przyjmijmy, że graf $G = (V, E)$ ma postać ścieżki, tj. $V = \{1, 2, \ldots, m\}$ oraz $E = \{(i, i+1)\}$ (założenie to przyjmujemy w dalszej części pracy). Zgodnie z twierdzeniem, które udowodnił Hammersley i Clifford (1971), zachodzi wtedy:

$$p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) \propto \exp\left(\sum_{e\in E,k} \lambda_k f_k(e, \boldsymbol{y}|_e, \boldsymbol{x}) + \sum_{v\in V,k} \mu_k g_k(v, \boldsymbol{y}|_v, \boldsymbol{x})\right), \tag{10.15}$$

gdzie $\boldsymbol{x}$ jest ciągiem obserwacji, $\boldsymbol{y}$ jest ciągiem oznaczeń a $\boldsymbol{y}|_s$ jest zbiorem elementów $\boldsymbol{y}$ związanych z wierzchołkami podgrafu s (Lafferty i inni 2001). Wektor $\boldsymbol{\theta} = (\lambda_1, \lambda_2, \ldots, \mu_1, \mu_2, \ldots)$ zawiera parametry modelu estymowane w wyniku uczenia. W literaturze na temat metod CRF funkcje f_k oraz g_k są nazywane **cechami**.

Lafferty i inni (2001) wykonują estymację parametrów (uczenie) CRF za pomocą algorytmu optymalizacyjnego IIS (*Improved Iterative Scaling*) (Della Pietra i inni 1997). Ze względu na wolną zbieżność algorytmu IIS w większości współczesnych implementacji CRF stosuje się quasi-Newtonowski algorytm optymalizacyjny LBFGS (oszczędna pamięciowo wersja algorytmu BFGS, którą zaproponował Byrd i inni (1994)). Vishwanathan i inni (2006) proponują, by w uczeniu CRF używać algorytmu gradientowego SGD (*Stochastic Gradient Descent*), w którym każdy krok optymalizacyjny jest wykonywany w oparciu o pojedyncze, wybierane losowo obserwacje. Algorytm SGD został opracowany przez Bottou (1991) na potrzeby uczenia sieci neuronowych. Jak pokazuje Bottou (2011) użycie SGD zamiast LBFGS pozwala skrócić czas uczenia CRF ponad siedmiokrotnie.

Niech $(\boldsymbol{X}, \boldsymbol{Y})$ będzie modelem CRF o ustalonych cechach oraz parametrach $\boldsymbol{\theta}$. Wtedy, dla każdego ciągu obserwacji $\boldsymbol{x}$, korzystając z algorytmu Viterbiego można efektywnie znaleźć ścieżkę oznaczeń $\boldsymbol{y}$, która daje maksymalną wartość prawej strony proporcji 10.15.

Zajmiemy się teraz problemem definicji zbioru cech na potrzeby układu analizy intonacyjnej. Przy przyjętych założeniach co do topologii grafu G oraz wprowadzając dodatkowe ograniczenia na argumenty cech proporcję 10.15 sprowadzamy do postaci:

$$p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) \propto \exp\left(\sum_{t=1}^{T}\sum_{k} \lambda_k f'_k(\boldsymbol{y}_{t-1}, \boldsymbol{y}_t) + \sum_{t=1}^{T}\sum_{k} \mu_k g'_k(\boldsymbol{y}_t, \boldsymbol{x}, t)\right). \tag{10.16}$$

Określmy zbiory $\mathcal{M}$ i $\mathcal{J}$ jak w sekcji 9.2 oraz zbiory $\mathcal{O}$ i $\mathcal{S}$ jak w sekcji 10.2.1.

Dla każdej pary $(m_1, m_2) \in \mathcal{J}$ tworzymy cechę binarną f'_{m_1,m_2} taką, że:

$$f'_{m_1,m_2}(\boldsymbol{y}_{t-1}, \boldsymbol{y}_t) = 1 \iff \mathrm{M}(\boldsymbol{y}_{t-1}) = m_1 \wedge \mathrm{M}(\boldsymbol{y}_t) = m_2 \wedge \mathrm{E}(\boldsymbol{y}_{t-1}) \wedge \mathrm{B}(\boldsymbol{y}_t), \tag{10.17}$$

gdzie $\mathrm{M} : \mathcal{O} \mapsto \mathcal{M}$ zwraca nazwę melodii dla danego oznaczenia a predykaty E oraz B zwracają prawdę wtedy i tylko wtedy, gdy w grafie na rycinie 10.3 stan związany z daną obserwacją jest połączony krawędzią z wierzchołkiem odpowiednio 'e' oraz 'b'.

Dodatkowo, dla każdej trójki $(m, s_1, s_2) \in \mathcal{M} \times \mathcal{S} \times \mathcal{S}$ określamy cechę binarną f'_{m,s_1,s_2} taką, że:

$$f'_{m,s_1,s_2}(\boldsymbol{y}_{t-1}, \boldsymbol{y}_t) = 1 \iff \mathrm{M}(\boldsymbol{y}_{t-1}) = m \wedge \mathrm{M}(\boldsymbol{y}_t) = m \wedge \mathrm{N}(s_1, s_2), \tag{10.18}$$

gdzie predykat N zwraca prawdę wtedy i tylko wtedy gdy wierzchołki stanów s_1 oraz s_2 są połączone krawędzią w grafie na rycinie 10.3.

Niech $A = (S, a)$ będzie anotacją wejściową agenta CRFRecognizer. Oznaczmy przez $p \in \bowtie S$ ścieżkę ontologii StateScoring.

W przypadku indukcyjnej analizy fonologicznej przyjmujemy, że $\boldsymbol{x}_t = \overline{p[t]}$ a następnie dla każdej pary $(o, i) \in \mathcal{O} \times \{-5, -4, \ldots, 5\}$ określamy cechę binarną $g'_{o,i}$ w sposób następujący:

$$g'_{o,i}(\boldsymbol{y}_t, \boldsymbol{x}, t) = 1 \iff \boldsymbol{y}_t = o \wedge \lceil \boldsymbol{x}_t(o) \rceil = i. \tag{10.19}$$

W przypadku indukcyjno–dedukcyjnej analizy fonologicznej stosujemy dodatkowo cechy związane z głównym oraz pobocznym akcentem potencjalnym określone następująco:

$$g'_{\mathrm{pri}}(\boldsymbol{y}_t, \boldsymbol{x}, t) = 1 \iff \exists_{s \in S}\, s \rhd p[i] \wedge \mathrm{pri}(\overline{s}) \tag{10.20}$$

$$g'_{\mathrm{sec}}(\boldsymbol{y}_t, \boldsymbol{x}, t) = 1 \iff \exists_{s \in S}\, s \rhd p[i] \wedge \mathrm{sec}(\overline{s}) \tag{10.21}$$

gdzie pri oraz sec są predykatami zwracającymi prawdę wtedy i tylko wtedy, gdy argument jest ontologią klasy odpowiednio PrimaryPotentialAccent lub SecondaryPotentialAccent (por. sekcja 10.1).

Agent CRFRecognizer został wykonany w oparciu o bibliotekę SGDCRF (Bottou 2011). Biblioteka SGDCRF jest implementacją modelu CRF z algorytmem estymacji SGD w języku C++.

10.2.5 ToneAssigner

Agent ToneAssigner dodaje do anotacji warstwę melodii (ontologie Tone) wynikającą bezpośrednio z wejściowej warstwy stanów (ontologie State).

10.3 Uczenie i wyniki

Uczenie i pomiary skuteczności przeprowadzono na trzech korpusach mowy oznaczonych PoS1, PoS2 oraz BaS1. Korpusy PoS1 oraz BaS1 opisano na stronie 134. Przez PoS2 oznaczono podzbiór korpusu PoInt (Karpiński 2002), zawierający przetranskrybowane nagrania dialogów spontanicznych 24 mówców (12 par) zrealizowanych zgodnie z protokołem „*Map Task*" (Bard i inni 1996). PoS2 zawiera 7775 fraz intonacyjnych (wliczając frazy z poprawną oraz niepoprawną stukturą intonacyjną) obejmujących łącznie 39226 sylab fonologicznych. We użytych korpusach przyjęto jednakową częstotliwość próbkowania równą 16 kHz.

10.3.1 Założenia dotyczące pomiaru skuteczności

Podstawy pomiaru skuteczności układów analizy intonacyjnej opisano w sekcji 5.2 na stronie 75. Pomiar skuteczności porponowanego układu przeprowadzono przy uwzględnieniu następujących warunków: 1) zakresu danych wejściowych, 2) modelu wzorców, 3) podziału zbioru melodii.

Rozpatrujemy dwa zakresy danych wejściowych:

- CPE: dane pochodzące z ontologii ContextualPitchExcursion,
- CPE/PA: dane pochodzące z ontologii ContextualPitchExcursion oraz PotentialAccent.

Wybór między zakresami CPE a CPE/PA skutkuje otrzymaniem odpowiednio indukcyjnego oraz indukcyjno–dedukcyjnego algorytmu analizy.

Rozpatrujemy cztery modele wzorców:

- SVMLIN: SVMRecognizer z liniową funkcją jądra,
- SVMRBF: SVMRecognizer z radialną funkcją jądra,
- SVMLIN/CRF: SVMRecognizer z liniową funkcją jądra oraz CRFRecognizer,
- SVMRBF/CRF: SVMRecognizer z radialną funkcją jądrową oraz CRFRecognizer.

W przypadku modelu bez CRF w nazwie agent CRFRecognizer jest zastępowany atrapą, która dla każdej ontologii StateScoring tworzy ontologię State reprezentującą stan o największej wartości funkcji StateScoring (por. sekcja 10.2.3).

W pomiarze skuteczności układu stosujemy miarę zgodności (por. wzór 5.9 na stronie 76) oraz uogólnione na potrzeby niniejszej pracy miary jakości układów *ekstrakcji dokumentów* (por. np. Jurafsky i Martin 2000, 578). Niech $\mathcal{V}$ oznacza zbiór wszystkich anotacji, które zawierają warstwę sylab. Oznaczmy przez A_n oraz A'_n ciągi takie, że dla każdego i zachodzi $A_i, A'_i \in \mathcal{V}$ oraz ścieżki sylab w A_i oraz w A'_i są równe (oczywiście etykiety odpowiadających sobie segmentów na ścieżkach mogą być różne). Przez p_i oraz p'_i będziemy oznaczać warstwy sylab odpowiednio w anotacji A_i oraz A'_i. Przyjmujemy, że dla każdego i anotacja A_i jest poprawna (wzorcowa) natomiast anotacja A'_i została otrzymana na wyjściu układu, którego skuteczność mierzymy.

Niech będzie dany dowolny skończony zbiór etykiet K zawierający $\varnothing$ oraz funkcja $\psi : \mathcal{V} \times \mathbb{S} \mapsto K$, określona dla każdej pary $(A = (S, a), s)$ takiej, że $s \in S$ jest segmentem sylaby. Funkcję ψ nazywamy **funkcją kategorii intonacyjnych**. Niech $K' = K \setminus \{\varnothing\}$. Wprowadzamy następujące oznaczenia:

$$TP = \sum_{k \in K'} \sum_i \sum_j \delta(\psi(A_i, p_i[j]), k)\delta(\psi(A'_i, p'_i[j]), k), \tag{10.22}$$

$$FP = \sum_{k \in K'} \sum_i \sum_j (1 - \delta(\psi(A_i, p_i[j]), k))\delta(\psi(A'_i, p'_i[j]), k), \tag{10.23}$$

$$FN = \sum_{k \in K'} \sum_i \sum_j \delta(\psi(A_i, p_i[j]), k)(1 - \delta(\psi(A'_i, p'_i[j]), k)). \tag{10.24}$$

Oznaczenia wprowadzone powyżej są skrótami od *True-Positive*, *False-Positive* (błąd typu I) oraz *False-Negative* (błąd typu II). Funkcja binarna δ zwraca liczbę 1 gdy jej argumenty są równe oraz 0 w przeciwnym przypadku. Przez **dokładność** (*precission*) rozumiemy wartość:

$$P = \frac{TP}{TP + FP}. \tag{10.25}$$

Przez **kompletność** (*recall*) rozumiemy wartość:

$$R = \frac{TP}{TP + FN}. \tag{10.26}$$

Przez **miarę F** (*F-measure*) rozumiemy wartość:

$$F = 2 \cdot \frac{P \cdot R}{P + R}. \tag{10.27}$$

Jak łatwo zauważyć wartości P, R oraz F mieszczą się w przedziale rzeczywistym $[0; 1]$ (bądź są nieokreślone).

Funkcję kategorii intonacyjnych określamy następująco:

$$\psi_k(A=(S,a),s) = \begin{cases} \phi_k(\overline{m}) & \text{jeśli } \underset{m\in S}{\exists} \text{tone}(\overline{m}) \wedge \overleftarrow{m} = \overleftarrow{s} \\ \varnothing & \text{w przec. przyp.,} \end{cases} \tag{10.28}$$

gdzie $k \in \{1, 3, 12, 22\}$ a predykat tone jest prawdziwy wtedy i tylko wtedy, gdy w argumencie dano etykietę klasy Tone.

Funkcje ϕ_k określamy następująco:

$$\phi_1(e) = \begin{cases} \texttt{'acc'} & \text{jeśli } e[0..1] = \texttt{'st'} \wedge e[0..1] = \texttt{'nu'} \\ \varnothing & \text{w przec. przyp.,} \end{cases} \tag{10.29}$$

$$\phi_3(e) = e[0..1], \tag{10.30}$$

$$\phi_{12}(e) = e[0..3], \tag{10.31}$$

$$\phi_{22}(e) = e. \tag{10.32}$$

Funkcja ψ_1 pozwala porównać wyniki proponowanego układu z układami segmentacji intonacyjnej. Funkcje ψ_3 i ψ_{12} pozwalają porównać wyniki proponowanego układu z układami parsingu intonacyjnego o porównywalnej liczbie etykiet.

10.3.2 Uczenie pod nadzorem

Uczenie pod nadzorem wykonano w oparciu o zbiór anotacji wzorcowych opisany w sekcji 9.1. W uczeniu i pomiarze skuteczności zastosowano metodę walidacji krzyżowej (*cross-validation*). Frazy intonacyjne zawarte w korpusie PoS1 podzielono losowo na 10 rozłącznych podzbiorów o równej (z dokładnością do jednego elementu) wielkości. Podzbiory te oznaczono przez PoS1_i, gdzie $i \in I = \{0, 1, \ldots, 9\}$. Przy podziale użyto losowania warstwowego z trzema warstwami, po jednej dla każdego mówcy w korpusie PoS1. Następnie, dla każdego $i \in I$ skonstruowano trójkę uporządkowaną:

$$Z_i = (\bigcup_{k\in I\setminus\{i,j\}} \text{PoS}_k, \text{PoS}_j, \text{PoS}_i), \tag{10.33}$$

gdzie $j = i + 1 \bmod |I|$. Trójkę Z_i będziemy nazywać **zadaniem**, a jej kolejne elementy interpretować jako zbiór uczący ($Z_i[0]$), zbiór walidacyjny ($Z_i[1]$) oraz zbiór testowy ($Z_i[2]$).

Dla każdego zadania Z_i, zakresu danych wejściowych oraz modelu rozpoznawania wzorców wyznaczono parametry SVM maksymalizujące miarę F na zbiorze walidacyjnym przy zastosowaniu metody *grid-search* (por. Hsu i inni 2010). Następnie dla każdego zakresu danych wejściowych, modelu rozpoznawania wzorców oraz funkcji kategorii intonacyjnych ψ_j wyznaczono zgodność oraz wartości średnie dokładności, kompletności i miary F na zbiorze kategorii intonacyjnych oddzielnie dla każdego zadania Z_i. W ostatnim etapie wyliczono (skrypt Perl) wartości średnie oraz przedziały ufności (95%) dla miar zgodności, średniej dokładności, średniej kompletności i średniej miary F na zbiorze zadań. Otrzymane wyniki zestawiono w tabeli 10.1.

Tabela 10.1: Skuteczność procentowa układu rozpoznającego struktury intonacyjne. Wyniki uczenia pod nadzorem na korpusie PoS1.

Model	Dane	ψ	Zgodność	Dokładność	Kompletność	Miara F
SVMLIN	CPE	ψ_1	76.2 ± 1.23	57.8 ± 2.67	54.7 ± 1.88	56.0 ± 1.01
SVMLIN/CRF	CPE	ψ_1	80.0 ± 0.70	64.0 ± 1.95	64.4 ± 1.93	64.0 ± 0.60
SVMRBF	CPE	ψ_1	78.7 ± 0.99	62.1 ± 2.47	60.7 ± 2.18	61.2 ± 0.84
SVMRBF/CRF	CPE	ψ_1	83.6 ± 0.70	69.6 ± 2.07	72.8 ± 1.78	71.1 ± 0.80
SVMLIN	CPE/PA	ψ_1	76.4 ± 0.79	56.9 ± 1.69	62.4 ± 1.65	59.4 ± 0.69
SVMLIN/CRF	CPE/PA	ψ_1	81.7 ± 0.78	66.7 ± 2.00	68.0 ± 1.95	67.2 ± 0.92
SVMRBF	CPE/PA	ψ_1	81.4 ± 0.75	66.7 ± 2.01	66.3 ± 1.39	66.4 ± 0.80
SVMRBF/CRF	CPE/PA	ψ_1	84.6 ± 0.84	71.4 ± 2.06	74.4 ± 1.04	72.8 ± 1.15
SVMLIN	CPE	ψ_3	69.4 ± 0.97	51.8 ± 2.09	44.4 ± 1.43	47.5 ± 0.96
SVMLIN/CRF	CPE	ψ_3	73.2 ± 0.94	56.6 ± 2.22	56.8 ± 0.97	56.4 ± 0.91
SVMRBF	CPE	ψ_3	71.5 ± 1.29	54.1 ± 2.73	51.1 ± 1.18	52.3 ± 1.41
SVMRBF/CRF	CPE	ψ_3	78.9 ± 0.68	66.1 ± 1.81	65.2 ± 0.78	65.4 ± 0.94
SVMLIN	CPE/PA	ψ_3	71.1 ± 0.99	53.7 ± 2.36	50.4 ± 1.05	51.8 ± 1.16
SVMLIN/CRF	CPE/PA	ψ_3	75.9 ± 0.94	61.6 ± 2.38	60.1 ± 0.85	60.6 ± 0.94
SVMRBF	CPE/PA	ψ_3	73.6 ± 0.91	56.7 ± 2.09	57.4 ± 1.07	56.8 ± 0.95
SVMRBF/CRF	CPE/PA	ψ_3	79.6 ± 1.11	66.8 ± 2.76	67.7 ± 0.96	67.1 ± 1.24
SVMLIN	CPE	ψ_{12}	65.9 ± 1.61	39.8 ± 2.06	41.2 ± 0.97	36.0 ± 1.39
SVMLIN/CRF	CPE	ψ_{12}	73.6 ± 1.00	49.6 ± 1.74	52.1 ± 1.16	47.4 ± 1.28
SVMRBF	CPE	ψ_{12}	70.5 ± 1.32	45.9 ± 2.32	46.5 ± 1.34	42.5 ± 1.31
SVMRBF/CRF	CPE	ψ_{12}	76.9 ± 0.65	53.2 ± 1.18	62.5 ± 1.98	54.1 ± 1.01
SVMLIN	CPE/PA	ψ_{12}	69.2 ± 1.21	43.9 ± 2.21	46.2 ± 1.84	41.3 ± 1.40
SVMLIN/CRF	CPE/PA	ψ_{12}	74.7 ± 1.02	50.5 ± 1.83	55.3 ± 1.59	49.4 ± 1.46
SVMRBF	CPE/PA	ψ_{12}	73.5 ± 1.06	49.5 ± 2.20	53.4 ± 1.63	47.8 ± 1.61
SVMRBF/CRF	CPE/PA	ψ_{12}	78.2 ± 0.81	54.6 ± 1.95	63.5 ± 1.42	55.3 ± 1.42
SVMLIN	CPE	ψ_{22}	66.8 ± 1.53	37.1 ± 2.24	40.1 ± 2.03	34.1 ± 1.23
SVMLIN/CRF	CPE	ψ_{22}	71.8 ± 0.46	42.6 ± 1.07	52.9 ± 1.32	42.7 ± 0.82
SVMRBF	CPE	ψ_{22}	69.5 ± 1.27	39.6 ± 2.23	46.8 ± 1.11	38.6 ± 1.71
SVMRBF/CRF	CPE	ψ_{22}	76.2 ± 1.22	47.4 ± 2.18	61.6 ± 1.46	49.5 ± 1.60
SVMLIN	CPE/PA	ψ_{22}	68.2 ± 1.22	37.1 ± 3.55	45.8 ± 0.93	36.6 ± 2.24
SVMLIN/CRF	CPE/PA	ψ_{22}	74.3 ± 1.15	46.1 ± 1.86	55.5 ± 2.11	46.4 ± 1.50
SVMRBF	CPE/PA	ψ_{22}	71.6 ± 1.13	41.8 ± 2.17	53.0 ± 1.36	42.7 ± 1.87
SVMRBF/CRF	CPE/PA	ψ_{22}	78.3 ± 0.93	50.5 ± 2.22	63.2 ± 1.65	52.6 ± 1.74

Ze względu na różnice m.in. w zakresie przyjętych anotacji fonologicznych, rodzaju mowy oraz liczby mówców, odniesienie otrzymanych wyników do wcześniejszych publikacji wymaga dużej ostrożności. Najmniej wątpliwości może budzić porównanie miar zgodności przy zastosowaniu kategorii intonacyjnych ψ_1. W takim przypadku proponowany układ może być porównany z układami segmentacji intonacyjnej (czyli układami określającymi lokalizację sylab akcentowanych melodycznie, por. strona 75). Problem segmentacji intonacyjnej dla języka polskiego był wcześniej podejmowany m.in. w pracach Demenko (1999) oraz Wagner (2008), które opisano w rozdziale 5. W obu przypadkach zastosowano sieci neuronowe uzyskując maksymalne zgodności na poziomie odpowiednio 82% oraz 81.95% a więc poniżej dolnej granicy przedziału ufności zgodności najlepszego z testowanych modeli, który osiągnął zgodność na poziomie $84.6\% \pm 0.84$ (model SVMRBF/CRF z danymi CPE/PA). Należy zauważyć, że cytowane prace dotyczą mowy czytanej lub powtarzanej pojedynczego mówcy. Prezentowane

przez nas wyniki dotyczą mowy spontanicznej trzech mówców.

Analizując wyniki zgodności można zauważyć (wielokrotnie wskazywaną w literaturze) niską czułość tej miary na degradację skuteczności układu w warunkach nierównomiernego rozkładu etykiet. Na nierównomierność rozkładu etykiet w przypadku miary zgodności mają m.in. wpływ sylaby nieakcentowane stanowiące większość sylab w mowie. Z powyższych względów dalsze omówienie skuteczności układu jest oparte na mierze F.

Na uwagę zwraca relatywnie niski uzysk związany z wprowadzeniem akcentu potencjalnego (dane CPE/PA w stosunku do danych CPE). W szczególności przedziały ufności miar F dla najskuteczniejszego z modeli (SVMRBF/CRF) z danymi CPE ($71.1\% \pm 0.80$) oraz z danymi CPE/PA ($72.8\% \pm 1.15$) nie są rozłączne. Jednym z powodów tej sytuacji może być zbyt niska ilość informacji pozyskiwanych w proponowanym układzie z sygnału ortograficznego. Wykorzystane reguły akcentuacji były tworzone z celem uzyskania maksymalnej kompletności (kosztem prezyzji). Przewidywano, że strategia taka będzie odpowiednia ze względu na wysoką zmienność (także błędy) obserwowaną w mowie spontanicznej. Innym powodem może być relatywnie duża zawartość informacyjna samego strumienia akustycznego w analizowanym materiale. Powyższe hipotezy zostaną sprawdzone w dalszym toku prac nad układem.

W prezentowanych badaniach modele SVMRBF osiągają konsekwentnie wyższą skuteczność od modeli SVMLIN. Jest to zgodne z wynikami uzyskiwanymi w modelowaniu metodą wektorów nośnych w szeregu innych dziedzinach (por. np. Hsu i inni 2010) ale niezgodne z np. z wynikami Rosenberga (2009, 127), który badał przydatność różnych technik rozpoznawania wzorców w detekcji granic fraz intonacyjnych w mowie.

Na koniec odniesiemy się do tezy zamieszczonej we wstępie niniejszej pracy mówiącej, że w układzie rozpoznawania struktur intonacyjnych model SVM/CRF osiągnie wyższą skuteczność w stosunku do modelu SVM. Jak wynika z wartości średnich i przedziałów ufności podanych w tabeli 10.1 miary F modelu SVMRBF/CRF istotnie przewyższą odpowiadające im miary F dla modelu SVMRBF dla dowolnej funkcji ψ_i.

10.3.3 Uczenie pod częściowym nadzorem

Przez **uczenie pod częściowym nadzorem** rozumiemy uczenie, w którym anotacje referencyjne są dostępne tylko dla części zbioru uczącego. Levow (2006), Jeon i Liu (2009) oraz Margolis i inni (2010a) pokazali, że uczenie pod częściowym nadzorem pozwala zwiększyć skuteczność układów analizy intonacyjnej na nowych korpusach mowy. W bieżącej sekcji opisano wstępny eksperyment polegający na uczeniu pod częściowym nadzorem układu analizy intonacyjnej opartego na najskuteczniejszym modelu frazy intonacyjnej otrzymanym w sekcji 10.3.2 (SVMRBF/CRF).

Z korpusów mowy PoS2 oraz BaS1 losowo wybrano podkorpusy $\mathrm{PoS2}_0$ oraz $\mathrm{BaS1}_0$ zawierające po 100 fraz intonacyjnych. Na korpusach $\mathrm{PoS2}_0$ oraz $\mathrm{BaS1}_0$ wykonano subiektywną analizę intonacyjną zgodnie z protokołem opisanym w sekcji 9.1. Wszystkie wystąpienia melodii rdzennych rosnących oraz opadających otrzymane w wyniku analizy subiektywnej przyporządkowano do jednej z klas `'nurilo'`, `'nuriwi'`, `'nurihi'`, `'nufalo'` oraz `'nufawi'` za pomocą klasyfikatorów uczonych na korpusie PoS1 (por. sekcja 9.2). Powstałe w ten sposób anotacje uznano za referencyjne a korpusy $\mathrm{PoS2}_0$ oraz $\mathrm{BaS1}_0$ przyjęto za zbiory testowe. Przyjęto ponadto następujące oznaczenia: $\mathrm{PoS2}_A = \mathrm{PoS2} \setminus \mathrm{PoS2}_0$ oraz $\mathrm{BaS2}_A = \mathrm{BaS2} \setminus \mathrm{BaS2}_0$.

Z pozostałych fraz korpusu PoS2 losowo wybrano ciągi zbiorów PoS2_i dla $i = 1, 2, \ldots, 7$, $j = 1, 2, \ldots, 10$ takie, że:

$$|\text{PoS2}_i| = 100 \cdot 2^{i-1} \tag{10.34}$$

oraz

$$\text{PoS2}_i \subset \text{PoS2}_{i+1}. \tag{10.35}$$

Analogicznie wybrano ciągi zbiorów BaS1_i dla $i = 1, 2, \ldots, 6$.

Dla każdego $i > 0$ układ analizy anotacyjnej oparty na modelu SVMRBF/CRF oraz uczony wcześniej pod nadzorem na korpusie PoS1 w ramach zadania Z_0 poddano dalszemu uczeniu pod częściowym nadzorem w następujących krokach:

1. pomiar średnich skuteczności oraz średnich miar dokładności, kompletności i miary F dla ostatnio uczonego układu na zbiorze PoS2_0.
2. anotacja zbioru PoS2_i układem uczonym na zbiorze PoS2_{i-1},
3. ponowne uczenie układu na zbiorze $\text{PoS1} \cup \text{PoS2}_i$.

Przedstawione kroki powtarzano do momentu zatrzymania wzrostu miary F na zbiorze PoS2_0 (nie więcej niż 20 iteracji). Analogiczną procedurę przeprowadzono dla zbioru BaS2. Wyniki opisanego eksperymentu zaprezentowano w tabelach 10.2 oraz 10.3.

Zdając sobie sprawę z niedoskonałości metod pomiaru (brak walidacji krzyżowej, brak przedziałów ufności) ograniczymy wnioski z bieżącego eksperymentu do minimum.

Zastosowanie modelu SVMRBF/CRF uczonego na korpusie PoS1 do analizy dowolnego z wymienionych korpusów wiąże się ze znacznym spadkiem skuteczności. Miara F (dla ψ_1) spada z $72.8\% \pm 1.15$ do 55.98% w przypadku zbioru testowego PoS2_0 oraz do 58.09% w przypadku BaS1_0. Obserwując pozostałe wyniki eksperymentu można zakładać, że spadek ten jest statystycznie istotny.

Zastosowanie opisanej procedury prowadzi do poprawy miary F już w przypadku zastosowania niewielkiego zbioru nieetykietowanego (100 fraz). Zwiększanie zbioru nieetykietowanego nie prowadzi do *znaczącej* dalszej poprawy skuteczności a po przkroczeniu 800 lub 1600 przykładów nieetykietowanych skuteczność zaczyna *niaznacznie* spadać.

Przypuszcza się, że jednym z powodów spadku skuteczności układu na nowych korpusach jest mała liczba mówców (3 osoby) w korpusie wykorzystywanym w uczeniu pod nadzorem. W ramach dalszych pracach nad opisanym układem planuje się: 1) anotowanie większego materiału w trybie pół-automatycznym z wykorzystaniem układu, 2) analizę przyczyn niewielkiej poprawy skuteczności po przejściu z danych CPE na dane CPE/PA oraz 3) dopracowanie eksperymentów uczenia pod częściowym nadzorem.

10.3.4 Wizualizacja wyników

W tabeli 10.4 przedstawiono proponowane ikony melodii, które są stosowane w wizualizacjach fonologicznej anotacji tonalnej w dalszej części rozdziału. Ikony nawiązują do szeregu prac Szkoły Brytyjskiej (por. np. Wells 2006, 262). Proponowane ikony uzyskano ze znaków ASCII przy zastosowaniu podstawowych znaczników formatujących (indeks górny oraz indeks dolny).

Tabela 10.2: Skuteczność procentowa układu rozpoznającego struktury intonacyjne opartego na modelu SVMRBF/CRF. Uczenie pod częściowym nadzorem na podzbiorach korpusu PoS2.

Podzbiór	Dane	ψ	Zgodność	Dokładność	Kompletność	Miara F
$\emptyset$	CPE	ψ_1	76.09	57.24	55.15	55.98
$\mathrm{PoS2}_1$	CPE	ψ_1	78.61	61.74	60.45	60.96
$\mathrm{PoS2}_2$	CPE	ψ_1	78.10	60.00	63.25	61.47
$\mathrm{PoS2}_3$	CPE	ψ_1	78.59	61.26	62.86	61.88
$\mathrm{PoS2}_4$	CPE	ψ_1	78.09	60.24	62.63	61.20
$\mathrm{PoS2}_5$	CPE	ψ_1	76.41	57.51	58.49	57.82
$\mathrm{PoS2}_6$	CPE	ψ_1	76.34	57.29	59.08	57.99
$\mathrm{PoS2}_7$	CPE	ψ_1	76.39	58.04	54.29	55.97
$\mathrm{PoS2}_A$	CPE	ψ_1	74.52	54.35	52.89	53.39
$\emptyset$	CPE/PA	ψ_1	75.64	55.98	58.11	56.83
$\mathrm{PoS2}_1$	CPE/PA	ψ_1	78.42	61.30	60.12	60.62
$\mathrm{PoS2}_2$	CPE/PA	ψ_1	78.00	60.33	61.91	60.88
$\mathrm{PoS2}_3$	CPE/PA	ψ_1	78.86	62.07	61.42	61.62
$\mathrm{PoS2}_4$	CPE/PA	ψ_1	78.40	60.94	62.34	61.48
$\mathrm{PoS2}_5$	CPE/PA	ψ_1	79.31	63.20	61.39	62.15
$\mathrm{PoS2}_6$	CPE/PA	ψ_1	78.19	60.26	63.27	61.62
$\mathrm{PoS2}_7$	CPE/PA	ψ_1	78.05	60.15	62.65	61.24
$\mathrm{PoS2}_A$	CPE/PA	ψ_1	76.71	58.24	57.57	57.75
$\emptyset$	CPE	ψ_3	68.42	48.71	46.04	47.09
$\mathrm{PoS2}_1$	CPE	ψ_3	72.39	56.20	51.54	53.49
$\mathrm{PoS2}_2$	CPE	ψ_3	72.26	55.34	53.07	53.94
$\mathrm{PoS2}_3$	CPE	ψ_3	72.65	56.58	51.96	53.87
$\mathrm{PoS2}_4$	CPE	ψ_3	72.52	56.30	52.29	53.92
$\mathrm{PoS2}_5$	CPE	ψ_3	70.74	53.89	48.64	50.81
$\mathrm{PoS2}_6$	CPE	ψ_3	70.67	54.02	47.32	50.08
$\mathrm{PoS2}_7$	CPE	ψ_3	69.43	51.86	44.41	47.53
$\mathrm{PoS2}_A$	CPE	ψ_3	67.17	46.77	41.90	43.89
$\emptyset$	CPE/PA	ψ_3	69.50	50.94	46.21	48.18
$\mathrm{PoS2}_1$	CPE/PA	ψ_3	70.49	52.13	51.38	51.51
$\mathrm{PoS2}_2$	CPE/PA	ψ_3	72.64	56.78	51.35	53.69
$\mathrm{PoS2}_3$	CPE/PA	ψ_3	72.37	56.42	53.44	54.64
$\mathrm{PoS2}_4$	CPE/PA	ψ_3	72.96	56.85	52.91	54.53
$\mathrm{PoS2}_5$	CPE/PA	ψ_3	72.01	54.87	54.19	54.22
$\mathrm{PoS2}_6$	CPE/PA	ψ_3	72.29	55.98	53.69	54.60
$\mathrm{PoS2}_7$	CPE/PA	ψ_3	72.56	56.01	52.05	53.77
$\mathrm{PoS2}_A$	CPE/PA	ψ_3	71.00	53.51	49.69	51.25

Tabela 10.3: Skuteczność procentowa układu rozpoznającego struktury intonacyjne opartego na modelu SVMRBF/CRF. Uczenie pod częściowym nadzorem na podzbiorach korpusu BaS1.

Podzbiór	Dane	Kategorie	Zgodność	Dokładność	Kompletność	Miara F
$\emptyset$	CPE	ψ_1	76.51	57.40	58.99	58.09
$BaS1_1$	CPE	ψ_1	78.50	61.22	61.67	61.32
$BaS1_2$	CPE	ψ_1	77.85	59.65	63.36	61.24
$BaS1_3$	CPE	ψ_1	78.37	60.48	63.46	61.82
$BaS1_4$	CPE	ψ_1	78.09	60.80	60.68	60.49
$BaS1_5$	CPE	ψ_1	77.49	58.66	64.00	61.12
$BaS1_6$	CPE	ψ_1	78.31	61.13	60.44	60.66
$BaS1_A$	CPE	ψ_1	78.23	61.14	60.82	60.71
$\emptyset$	CPE/PA	ψ_1	79.07	62.67	60.86	61.63
$BaS1_1$	CPE/PA	ψ_1	78.52	61.62	61.14	61.14
$BaS1_2$	CPE/PA	ψ_1	78.99	62.08	62.51	62.16
$BaS1_3$	CPE/PA	ψ_1	78.74	61.97	60.73	61.24
$BaS1_4$	CPE/PA	ψ_1	78.65	61.44	62.57	61.86
$BaS1_5$	CPE/PA	ψ_1	80.02	65.31	59.47	62.20
$BaS1_6$	CPE/PA	ψ_1	78.68	61.77	61.09	61.31
$BaS1_A$	CPE/PA	ψ_1	79.44	63.55	60.70	62.00
$\emptyset$	CPE	ψ_3	69.23	50.76	47.96	49.09
$BaS1_1$	CPE	ψ_3	71.63	53.94	53.45	53.42
$BaS1_2$	CPE	ψ_3	72.30	56.29	52.04	53.77
$BaS1_3$	CPE	ψ_3	71.91	54.93	55.31	54.84
$BaS1_4$	CPE	ψ_3	72.66	57.16	51.86	54.04
$BaS1_5$	CPE	ψ_3	72.45	55.69	52.25	53.73
$BaS1_6$	CPE	ψ_3	72.11	55.38	52.15	53.40
$BaS1_A$	CPE	ψ_3	71.04	52.86	52.56	52.54
$\emptyset$	CPE/PA	ψ_3	72.30	55.64	51.66	53.35
$BaS1_1$	CPE/PA	ψ_3	71.17	53.14	53.72	53.19
$BaS1_2$	CPE/PA	ψ_3	73.17	57.92	52.55	54.82
$BaS1_3$	CPE/PA	ψ_3	71.29	53.35	53.74	53.30
$BaS1_4$	CPE/PA	ψ_3	72.27	56.08	51.98	53.72
$BaS1_5$	CPE/PA	ψ_3	72.81	56.72	52.60	54.37
$BaS1_6$	CPE/PA	ψ_3	73.25	57.76	52.98	54.99
$BaS1_A$	CPE/PA	ψ_3	72.76	56.31	53.79	54.78

Tabela 10.4: Proponowane ikony melodii. Literę „a” dodano dla pokazania lokalizacji ikony względem tekstu.

Ikona	Nazwa	Ikona	Nazwa	Ikona	Nazwa
$\overset{(-)}{\text{a}}$	'welebe'	$\underline{\text{a}}$	'stlebe'	$=$a	'nule'
$\overset{(-)}{\text{a}}$	'weleab'	$\overline{\text{a}}$	'stleab'		
$\overset{(/)}{\text{a}}$	'weribe'	$\underset{/}{\text{a}}$	'stribe'	$\underset{/\!/}{\text{a}}$	'nurilo'
		$\overline{/\text{a}}$	'striab'	$\overline{/\!/\text{a}}$	'nurihi'
				$/\!/$a	'nuriwi'
		$\underset{\backslash}{\text{a}}$	'stfabe'	$\underset{\backslash\!\backslash}{\text{a}}$	'nufalo'
		$\overline{\backslash\text{a}}$	'stfaab'	$\overline{\backslash\!\backslash\text{a}}$	'nufahi'
				$\backslash\!\backslash$a	'nufawi'
		$\underset{/\backslash}{\text{a}}$	'strfbe'	$/\!/\backslash\!\backslash$a	'nurf'
		$\overline{/\backslash\text{a}}$	'strfab'		
		$\underset{\backslash/}{\text{a}}$	'stfrbe'	$\backslash\!\backslash/\!/$a	'nufr'
		$\overline{\backslash/\text{a}}$	'stfrab'		

W dalszej części sekcji pokazano przykładowe anotacje referencyjne (subiektywne) ze zbioru testowego PoS1_0 (poprzedzone symbolem ♡) oraz wyniki układu SVMRBF/CRF z danymi CPE/PA (poprzedzone symbolem □) na tym samym sygnale mowy. Rozpatrywane w przykładach sygnały nie należały do zbioru uczącego układu. Nad każdą parą przedstawiono tonetyczną transkrypcję międzyliniową wykonaną w trybie indukcyjno-dedukcyjnym przy zadanej anotacji referencyjnej. W tonetycznej transkrypcji liniowej wprowadzono dodatkową konwencję polegającą na odwzorowaniu sumy wartości atrybutu VOICINGDUR ontologii StandarizedPitchExcursion obejmowanych przez daną sylabę w postaci odcieni szarości.

1.

♡ $\backslash$i to chyba $\backslash\!\backslash$wszystko

□ $\backslash$i to chyba $\backslash\!\backslash$wszystko

Kolejno melodia „silna opadająca nad” oraz melodia „rdzenna opadająca pełna”. Anotacja referencyjna oraz anotacja otrzymana za pomocą układu są zgodne.

2.

♡ (–)im \bardziej kosz\marna jest \\sztuka
□ (–)im \bardziej kosz\marna jest \\sztuka

Kolejno melodia „słaba równa pod", „silna opadająca pod", „silna opadająca pod" oraz „rdzenna opadająca niska". Anotacja referencyjna oraz anotacja otrzymana za pomocą układu są zgodne.

3.

♡ \ale niespo=dzianka
□ \ale niespo\\dzianka

Kolejno melodia „silna opadająca nad" oraz melodia „rdzenna równa". W anotacji otrzymanej automatycznie melodia rdzenna została błędnie rozpoznana jako „rdzenna opadająca niska".

Więcej przykładów wizualizacji zamieszczono w dodatku B.

Podsumowanie

Praca zawiera trzy części. W części pierwszej (rozdziały 1-2) opisano proponowany model procesu komunikacji głosowej. W części drugiej (rozdziały 3-5) opisano ponad 80 reprezentacji, algorytmów oraz układów analizy cech tonalnych. W części trzeciej (rozdziały 6-10) przedstawiono układ rozpoznający struktury intonacyjne w sygnale mowy polskiej.

Najważniejsze oryginalne techniczne wyniki niniejszej pracy to:

- środowisko integracji programowych układów przetwarzania mowy i języka w oparciu o architekturę potokową oraz tablicową (rozdział 6),
- układ sygnałowej analizy tonalnej (ekstrakcji częstotliwości podstawowej) o zwiększonej tolerancji na zmianę parametrów akustycznych mówcy (rozdział 7),
- układ fonetycznej analizy tonalnej dla języka polskiego uwzględniający transkrypcję ortograficzną analizowanej mowy (rozdział 8),
- układ wizualizacji cech tonalnych dla języka polskiego z użyciem tonetycznej transkrypcji międzyliniowej (rozdział 8),
- układ rozpoznawania struktur intonacyjnych zgodnych z gramatyką polskiej frazy intonacyjnej Jassema z ikoniczną prezentacją wyników (rozdział 10).

Otrzymane wyniki umożliwiają m.in.:

- uwzględnienie intonacji w automatycznym rozpoznawaniu (ASR), rozumieniu (NLU) oraz tłumaczeniu (translacji) mowy,
- automatyczną anotację korpusów dla systemów syntezy mowy z tekstu (TTS),
- automatyczną indeksację oraz wyszukiwanie struktur intonacyjnych w obszernych korpusach mowy,
- automatyczną ocenę poprawności struktur intonacyjnych w mowie osób uczących się języków obcych oraz podlegających rehabilitacji mowy.

DODATEK A

Pseudokod wybranych algorytmów analizy tonalnej

Przedstawione w niniejszym dodatku algorytmy są autorską interpretacją opisów słownych, diagramów, pseudokodu i kodów źródłowych opublikowanych przez innych autorów. W algorytmach użyto oznaczeń i terminów wprowadzonych w pierwszej części pracy. Do zamieszczonych tutaj algorytmów odwołujemy się w drugiej części pracy.

A.1 IPO

Parametry oraz algorytmy pomocnicze użyte w algorytmie A.1:

- α:Integer — minimalna liczba segmentów F_0 obejmowanych przez segment anotacji IPO ($\alpha > 3$).
- $\beta(n$:Integer):Real — maksymalny dopuszczalny RMSE między n etyketami F_0 a linią regresji F_0, liczony w punktach lewych kotwic segmentów.
- $\gamma : Real$ — 50ms/T, gdzie T [ms] jest długością segmentu w anotacji F_0.
- IPO.beginIndex(V:Annotation): Integer — dla danej anotacji sekwencyjnej V, reprezentującej przebieg samogłoskowości, zwraca indeks segmentu (licząc od 0), od którego należy rozpocząć stylizację sygnału.
- RMSE(S:Segmentation,a_F:Function,a:Real,b:Real):Real — dla danej anotacji sekwencyjnej (S, a_F) oraz współczynników funkcji regresji liniowej (a, b) oblicza RMSE między etykietami segmentów oraz wartościami funkcji regresji liniowej w punktach lewych kotwic segmentów.
- addSegment(A:Annotation, t_0:Real, t_1:Real, e) — dla danej anotacji $A = (S, a)$ dodaje do segmentacji S segment, którego kotwice wskazują na punkty czasowe t_0, t_1 oraz ustala $a(\#s) = e$; algorytm zapewnia unikalność w S dla dodanych kotwic oraz segmentu.
- IPO.LR(S:Segmentation,f:Real,a_F:Function,a_V:Function): (a:Real,b:Real) — wylicza współczynniki (a, b) funkcji liniowej poprzez regresję liniową przebiegu (S, a_F) ważoną

Algorytm A.1 Fonetyczna analiza tonalna IPO (Hermes 2006, 36).

IPO.stylization(F:Annotation, V:Annotation):Annotation
Wejście F: przebieg F_0
Wejście V: przebieg samogłoskowości (opis w tekście)
Wyjście anotacja fonetyczna IPO
$(S_F, a_F) \leftarrow F$
$(S_V, a_V) \leftarrow V$
$k_0 \leftarrow$ IPO.beginIndex(V)
$f \leftarrow \overline{S_F[k_0]}$
Annotation A
while $k_0 < |S|$ **do**
  for $i = 1$ **to** 2 **do**
    Segmentation $S \leftarrow \{\}$
    $k_i \leftarrow k_{i-1}$
    repeat
      $S \leftarrow S \cup \{S_F[k_i]\}$
      (a_i, b_i)=IPO.LR(S, f, a_F, a_V)
      $e_i \leftarrow \mathrm{RMSE}(S, a_F, a_i, b_i)$
      $k_i \leftarrow k_i + 1$
    until $k_i = |S| \vee |S| \geqslant \alpha \vee e_i \geqslant \beta(|S|)$
  end for
  $e_{\min} \leftarrow e_1 + e_2$
  $j_{\min} \leftarrow k_1$
  for $j = k_0 + \gamma$ **to** $k_2 - \gamma$ **do**
    $S_L \leftarrow S_F[k_0..j]$
    $S_R \leftarrow S_F[j..k_2]$
    (a_1, b_1, a_2, b_2) =IPO.LR2(S_L, S_R, f, a_F, a_V)
    $e \leftarrow \mathrm{RMSE}(S_L, a_F, a_1, b_1)$+RMSE($S_R, a_F, a_2, b_2$)
    if $e < e_{\min}$ **then**
      $e_{\min} \leftarrow e$
      $j_{\min} \leftarrow j$
    end if
  end for
  addSegment($A, \overleftarrow{S_F[k_0]}, \overleftarrow{S_F[k_1]}, (a_1, b_1)$)
  $k_0 \leftarrow k_1$
  $f \leftarrow a_1 + b_1 \overleftarrow{S[j_{\min}]}$
end while
return A

przebiegiem (S, a_V). Funkcja wynikowa spełnia warunek $a + b(\overleftarrow{S[0]}) = f$. Wzory regresji ważonej określa się następująco:

$$a = \frac{1}{D}\left(\sum_i v_i \sum_i v_i t_i f_i - \sum_i v_i t_i \sum_i v_i f_i\right), \tag{A.1}$$

$$b = \frac{1}{D}\left(\sum_i v_i f_i \sum_i v_i t_i^2 - \sum_i v_i t_i \sum_i v_i t_i f_i\right), \tag{A.2}$$

gdzie $f_i = a_F(\#S[i])$, $v_i = a_V(\#S[i])$, $t_i = \overleftarrow{S[i]} - \overleftarrow{S[0]}$) oraz

$$D = \sum_i v_i \sum_i v_i t_i^2 - \left(\sum_i v_i t_i\right)^2, \tag{A.3}$$

por. Hermes (2006, 37).

- IPO.LR2(S_1:Segmentation, S_2:Segmentation, f:Real, a_F:Function, a_V:Function): (a_1: Real, b_1: Real, a_2:Real, b_2:Real) — wylicza współczynniki (a_1, b_1) oraz (a_2, b_2) dwóch funkcji liniowych poprzez regresję liniową przebiegów (S_1, a_F) oraz (S_2, a_F) ważoną przebiegami (S_1, a_V) oraz (S_1, a_V). Funkcje wynikowe spełnią dodatkowo warunek:

$$\underset{s \in S_1 \cap S_2}{\forall} a_1 + b_1 \overleftarrow{s} = a_2 + b_2 \overleftarrow{s} = f. \tag{A.4}$$

A.2 MOMEL

Algorytm A.2 Redukcja mikrointonacji w fonetycznej analizie tonalnej MOMEL (Hirst i inni 2000, 63).

1: MOMEL.macrointonation(F:Annotation): Annotation
Wejście F: ramkowa sygnałowa anotacja tonalna (przebieg F_0)
Wyjście przebieg F_0 o zmniejszonym udziale mikrointonacji
2: $(S, a) \leftarrow F$
3: $S[0] \leftarrow 0$
4: $S[|S| - 1] \leftarrow 0$
5: **if** $\overline{S[1]} = \varnothing$ **then**
6: $\quad \overline{S[1]} \leftarrow 0$
7: **end if**
8: **for** $i = 1$ **to** $|S| - 2$ **do**
9: $\quad$ **if** $\overline{S[i+1]} = \varnothing$ **then**
10: $\quad\quad \overline{S[i+1]} \leftarrow 0$
11: $\quad$ **end if**
12: $\quad$ **if** $\left(\overline{S[i]} > (1+\gamma)\overline{S[i-1]}\right) \wedge \left(\overline{S[i]} > (1+\gamma)\overline{S[i+1]}\right)$ **then**
13: $\quad\quad \overline{S[i]} \leftarrow 0$
14: $\quad$ **end if**
15: **end for**
16: **return** (S, a)

Algorytm A.3 Wyznaczenie etykiet kandydujących w fonetycznej analizie tonalnej MOMEL (Hirst i inni 2000, 63).

MOMEL.candidates(F:Annotation): Annotation
Wejście F: ramkowa sygnałowa anotacja tonalna (przebieg F_0)
Wyjście anotacja ramkowa z etykietami kandydującymi anotacji MOMEL
$(S, a) \leftarrow F$
Annotation C
for all $s \in S$ **do**
  $U \leftarrow \{u \in S : (\overleftarrow{u} > \overleftarrow{s} - A) \wedge (\overrightarrow{u} < \overrightarrow{s} + A)\}$
  for all $u \in U$ **do**
    if $\left(\overline{u} < F_0^{\min}\right) \vee (\overline{u} > F_0^{\max})$ **then**
      $\overline{u} \leftarrow \varnothing$
    end if
  end for
  repeat
    $(a, b, c) = \text{quadraticRegression}((U, a))$
    $k \leftarrow 0$
    for all $u \in U$ **do**
      if $\overline{u} + \Delta < a + b\overleftarrow{u} + c\overleftarrow{u}^2$ **then**
        $\overline{u} \leftarrow \varnothing$
        $k \leftarrow k + 1$
      end if
    end for
  until $k > 0$
  $t \leftarrow -b/2c$
  $h \leftarrow a + b\overleftarrow{u} + c\overleftarrow{u}^2$
  if $(t \geqslant \overleftarrow{s} - A) \wedge (t < \overrightarrow{s} + A) \wedge (h \geqslant F_0^{\max}) \wedge \left(h \leqslant F_0^{\min}\right)$ **then**
    addSegment($C, \overleftarrow{s}, \overrightarrow{s}, (t, h)$)
  else
    addSegment($C, \overleftarrow{s}, \overrightarrow{s}, \varnothing$)
  end if
end for
return C

Parametry oraz algorytmy pomocnicze stosowane w algorytmach bieżącej sekcji:

- γ — próg dopuszczalnej relatywnej odległości od mody w regresji modalnej; domyślnie 0.05.
- A — 1/2 czasu trwania okna regresji; domyślnie 150 ms.
- R — 1/2 czasu trwania okna w algorytmie segmentacji; domyślnie 100 ms.
- Δ — parametr mody.
- $F_0^{\min}/F_0^{\max}$ — wartość minimalna i maksymalna F_0; domyślnie $F_0^{\min} = 50Hz$ a $F_0^{\max}$ równe jest średniej ze zbioru 5
- quadraticRegression(Annotation A): (Real, Real, Real) dla danej anotacji A o etykietach skalarnych oblicza funkcję regresji kwadratowej etykiet względem położenia

Algorytm A.4 Wyznaczanie segmentacji w fonetycznej analizie tonalnej MOMEL Hirst i inni (2000, 64).

1: MOMEL.segmentation(C:Annotation): Segmentation
Wejście C: anotacja ramkowa z etykietami kandydującymi anotacji MOMEL
Wyjście segmentacja MOMEL
2: $(S, a) \leftarrow C$
3: Annotation D
4: **for all** $s \in S$ **do**
5: $U_0 \leftarrow \{u \in S : (\overleftarrow{u} > \overleftarrow{s} - R) \wedge (\overrightarrow{u} \leqslant \overrightarrow{s})\}$
6: $U_1 \leftarrow \{u \in S : (\overleftarrow{u} > \overleftarrow{s}) \wedge (\overrightarrow{u} < \overrightarrow{s} + R)\}$
7: $(\hat{t}_0, \hat{h}_0) \leftarrow \text{mean}((U_0, a))$
8: $(\hat{t}_1, \hat{h}_1) \leftarrow \text{mean}((U_1, a))$
9: addSegment($D, \overleftarrow{s}, \overrightarrow{s}, (|\hat{t}_1 - \hat{t}_0|, |\hat{h}_1 - \hat{h}_0|)$)
10: **end for**
11: $(\hat{t}, \hat{h}) \leftarrow \text{mean}(D)$
12: $\check{t} = 1/\hat{t}$
13: $\check{h} = 1/\hat{h}$
14: Annotation E
15: **for all** $d \in D$ **do**
16: $(t, h) \leftarrow \overline{d}$
17: addSegment($E, \overleftarrow{d}, \overrightarrow{d}, (t * \check{t} + h * \check{h})/(\check{t} + \check{h})$)
18: **end for**
19: $(\hat{e}) \leftarrow \text{mean}(E)$
20: Segmentation M
21: $b \leftarrow 0$
22: **for** $i = 1$ **to** $|E| - 2$ **do**
23: **if** $\left(\overline{S[i]} > \hat{e}\right) \wedge \left(\overline{S[i]} > \overline{S[i-1]}\right) \wedge \left(\overline{S[i]} > \overline{S[i+1]}\right)$ **then**
24: addSegment($M, b, \overleftarrow{S[i]}$)
25: $b \leftarrow \overleftarrow{S[i]}$
26: **end if**
27: **end for**
28: addSegment($M, b, \overrightarrow{S[|E| - 1]}$)
29: **return** M

czasowego segmentu; zwraca trójkę uporządkowaną (a, b, c) reprezentującą wynikową funkcję kwadratową $a + bt + ct^2$; wartości $\varnothing$ są ignorowane.

- addSegment(A:Annotation, t_0:Real, t_1:Real, e) — dla danej anotacji $A = (S, a)$ dodaje do segmentacji S segment, którego kotwice wskazują na punkty czasowe t_0, t_1 oraz ustala $\overline{s} = e$; algorytm zapewnia unikalność w S dla dodanych kotwic oraz segmentu.

- mean(A: Annotation):Real[] — dla danej anotacji A o etykietach w postaci wektora rzeczywistego zwraca wektor wartości średnich w poszczególnych wymiarach; wartości $\varnothing$ są ignorowane.

A.3 Fujisaki

Algorytm A.5 Fonetyczna analiza tonalna Fujisaki (Mixdorff 2000).

MixdorffFujisaki(x:Signal): Annotation
Wejście x: sygnał mowy obejmujący pojedynczą frazę intonacyjną
Wyjście anotacja Fujisaki
Annotation F
Signal v, e
$(F, v, e) =$ Rapp.extract(x)
$f_M \leftarrow$ Hirst.approxMOMEL(F)
$f_H =$ hiPass(f_M, 0.5Hz)
$A_a \leftarrow$ Mixdorff.accentCommands(f_H)
Mixdorff.optimize(A_a, f_H)
$f_L = f_M - f_H$
$A_p \leftarrow$ Mixdorff.phraseCommands(f_L)
Mixdorff.optimize(A_p, f_L)
Annotation $A \leftarrow A_p \cup A_a$
Real $b = \min(f_L)$
addSegment($A, 0, |f_M|, b$)
Mixdorff.optimize(A, f_M)
$w = v \cdot e$
Mixdorff.optimizeWeighted(A, f, w)
return A

Algorytm A.6 Inicjalizacja komend akcentowych w fonetycznej analizie tonalnej Fujisaki (Mixdorff 2000).

Mixdorff.accentCommands(f:Signal):Annotation
Real $t \leftarrow 0$
Annotation A
while $t < |f|$ **do**
  $t_0 \leftarrow$ nextLocalMin(f, t)
  $t_1 \leftarrow$ nextLocalMin(f, t_0)
  if $t_0 < t_1 - 200ms$ **then**
    $t_{\max} \leftarrow$ nextLocalMax(f, t_0)
    addSegment($A, t_0, t_1 - 200$ms, $f(t_{\max})$)
  end if
  $t \leftarrow t_1$
end while

Algorytmy pomocnicze stosowane w algorytmach A.5, A.6 oraz A.7:

- addSegment(A:Annotation, t_0:Real, t_1:Real, e) — dla danej anotacji $A = (S, a)$ dodaje do segmentacji S segment, którego kotwice wskazują na punkty czasowe b oraz f rozszerza funkcję a tak, aby $a(\#s) = e$; algorytm zapewnia unikalność dodanych kotwic oraz segmentu w S.

- nextLocalMin(f, t_0):Real — zwraca najmniejsze $t \geqslant t_0$ takie, że w punkcie t przypada minimum lokalne sygnału f.

Algorytm A.7 Inicjalizacja komend frazowych w fonetycznej analizie tonalnej Fujisaki (Mixdorff 2000).

```
1: Mixdorff.phraseCommands(f:Signal):Annotation
2: Real t ← 0
3: Annotation A
4: while t < |f| do
5:    t_0 ← nextLocalMin(f, t)
6:    t_1 ← nextLocalMax(f, t_0)
7:    if t_0 < t_1 then
8:       (S, a) ← A
9:       m ← f(t_1) − Σ_{s∈S} G_p(t_0 − ←s)
10:      addSegment(A, t_0, t_1, m)
11:   end if
12:   t ← t_1
13:   return A
14: end while
```

- nextLocalMax(f, t_0):Real — zwraca najmniejsze $t \geqslant t_0$ takie, że w punkcie t przypada maksimum lokalne sygnału f.

- Mixdorff.optimize(U:Annotation, f:Signal) — zmienia lokalizację kotwic oraz etykiet anotacji Fujisaki U tak, by zminimalizować odległość między sygnałem f oraz sygnałem f_U określonym we wzorze 4.6; algorytm oparty jest na minimalizacji gradientowej (por. opis w sekcji 7.2 na stronie 105).

- Mixdorff.optimizeWeighted(U:Annotation, f:Signal, w:Signal) — rozrzerzony wariant algorytmu Mixdorff.optimize, w którym odległość między sygnałami f_U oraz f obliczana jest z uwzględnieniem wagi w; w określa *istotność* fragmentów przebiegu F_0.

- Rapp.extract(x:Signal): (Annotation,Signal,Signal) — algorytm zwraca trójkę uporządkowaną F, v, e reprezentujących odpowienio F_0, harmoniczność oraz energię chwilową sygnału x z korpusu mowy, który zebrał Rapp (1998b).

- Hirst.approxMOMEL(F:Annotation):Signal — dla F reprezentującego przebieg F_0 zwraca sygnał powstający w wyniku aproksymacji przebiegu wejściowego metodą MOMEL (Hirst i Espesser 1993) (więcej o analizie MOMEL w sekcji 4.2).

A.4 Tilt

Parametry oraz algorytmy pomocnicze stosowane w algorytmach bieżącej sekcji:

- Tilt.medianSmoothing(F:Annotation) — 7-11 punktowe wygładzanie medianowe danej w argumencie anotacji sekwencyjnej o etykietach skalarnych.

- Tilt.linearFilling(F:Annotation) — przyjmijmy, że $(S, a) = F$ jest anotacją sekwencyjną o etykietach skalarnych; funkcja dla każdego s takiego, że $\overline{s} = \varnothing$ przypisuje $\overline{s}$ wynik interpolacji liniowej etykiet dwóch najbliższych czasowo segmentowi s segmentów $s_1, s_2 \in S$ takich, że $s_1 <^S s <^S s_2$, $\overline{s_1} \neq \varnothing$ oraz $\overline{s_2} \neq \varnothing$.

Algorytm A.8 Fonetyczna analiza tonalna RFC (Taylor 1995a, 7).

Tilt.RFC(F:Annotation, E:Annotation)
Wejście F: ramkowa sygnałowa anotacja tonalna (przebieg F_0)
Wejście E: nieciągła sekwencyjna fonologiczna anotacja tonalna (zdarzenia int.)
Wyjście E: nieciągła sekwencyjna fonetyczna anotacja tonalna (RFC)
Tilt.medianSmoothing(F)
Tilt.linearFilling(F)
$(S_F, a_F) \leftarrow F$
$(S_E, a_E) \leftarrow E$
for all $s_E \in S_E$ **do**
  $S_{left} \leftarrow \{s \in S_F : \overleftarrow{s_E} - 150\text{ms} \leqslant \overleftarrow{s} < \overleftarrow{s_E} + 0.2(\overrightarrow{s_E} - \overleftarrow{s_E})\}$
  $S_{right} \leftarrow \{s \in S_F : \overrightarrow{s_E} - 0.2(\overrightarrow{s_E} - \overleftarrow{s_E}) \leqslant \overleftarrow{s} < \overrightarrow{s_E} + 150\text{ms}\}$
  Segmentation $S \leftarrow \{s_F \in S_F : s_E \blacktriangleright s_F\}$
  Segment $p \leftarrow$ Tilt.pickPeak((S, a_F))
  if $p \neq \varnothing$ **then**
    $\overleftarrow{s_E} \leftarrow$ Tilt.maximizeRise($S_{left}, \{p\}, a_F$)
    $D_{rise} \leftarrow \overleftarrow{p} - \overleftarrow{s_E}$
    $\overrightarrow{s_E} \leftarrow$ Tilt.maximizeFall($\{p\}, S_{right}, a_F$)
    $D_{fall} \leftarrow \overrightarrow{s_E} - \overleftarrow{p}$
  else if $\overline{S[0]} < \overline{S[|S|-1]}$ **then**
    $\overleftarrow{s_E} \leftarrow$ Tilt.maximizeRise(S_{left}, S_{right}, a_F)
    $D_{rise} \leftarrow \overleftarrow{p} - \overleftarrow{s_E}$
    $D_{fall} \leftarrow 0$
  else
    $\overleftarrow{s_E} \leftarrow$ Tilt.maximizeFall(S_{left}, S_{right}, a_F)
    $D_{fall} \leftarrow \overrightarrow{s_E} - \overleftarrow{p}$
    $D_{rise} \leftarrow 0$
  end if
  $h \leftarrow \overline{S_F(\overleftarrow{S_E})}$
  $A_{rise} \leftarrow \overline{S_F(\overleftarrow{S_E} + D_{rise})} - \overline{S_F(\overleftarrow{S_E})}$
  $A_{fall} \leftarrow \overline{S_F(\overrightarrow{S_E})} - \overline{S_F(\overleftarrow{S_E} + D_{rise})}$
  $\overline{s_E} \leftarrow (A_{rise}, A_{fall}, D_{rise}, D_{fall}, h)$
end for

- Tilt.pickPeak(F): Segment — dla danej w argumencie sekwencyjnej anotacji $(S, a) = F$ o etykietach skalarnych zwraca pierwsze maksimum lokalne w porządku $<^S$ lub $\varnothing$ jeśli maksimum lokalne nie występuje.

- Tilt.maximizeRise(S_0:Segmentation, S_1:Segmentation, F:Annotation): (Segment,Segment) — dla danej anotacji sekwencyjnej $(S, a) = F$ o etykietach skalarnych oraz danych segmentacji $S_0 \in S$ i $S_1 \in S$ takich, że dowolny segment należący do S_0 poprzedza dowolny segment należący do S_1, funkcja Tilt.maximizeRise zwraca parę (s_0, s_1) minimalizującą na zbiorze $S_0 \times S_1$ dystans między przebiegiem etykiet ścieżki segmentalnej od s_0 do s_1 oraz próbkowanym w punktach kotwic segmentów ścieżki przebiegiem funkcji f określonej we wzorze 4.14 z parametrami $D_{fall} = A_{fall} = 0$. Przez dystans rozumiana jest metryka euklidesowa. Rozwiązanie uzyskiwane jest poprzez wyczerpujące przeszukiwanie zbioru rozwiązań.

- Tilt.maximizeFall(S_0:Segmentation, S_1:Segmentation, F:Annotation): (Segment,Segment) — funkcja analogiczna do Tilt.maximizeRise, przy czym zamiast $D_{fall} = A_{fall} = 0$ za-

kłada się $D_{rise} = A_{rise} = 0$.

A.5 Prosogram

W algorytmie A.9 przedstawiono Prosogram w wersji 2.4f (Mertens 2009). Dla zwiększenia czytelności nie objęto w prezentowanych algorytmach szeregu przypadków brzegowych oraz opcji dostępnych w programie Prosogram. Prosogram zaimplementowany jest jako skrypt programu Praat Boersma i Weenink (2008).

Algorytm A.9 Fonetyczna analiza tonalna Prosogram (Mertens 2009).

Prosogram(x: Signal, A: Annotation): Annotation
Wejście x: sygnał mowy
Wejście A: zakotwiczona anotacja prosta: 1) głoskowa, 2) sylabiczna albo 3) pusta
Wyjście prozogram
{Etap 1. Lokalizacja ośrodków fonetycznych sylab}
Annotation $L \leftarrow$ Prosogram.toLoudness(x)
Annotation $N \leftarrow$ Prosogram.syllabicNuclei(A, L)
{Etap 2. Wyznaczenie segmentacji}
Segmentation $S \leftarrow \emptyset$
Annotation $F0 \leftarrow$ Praat.extractF0(x)
$(S_N, a_N) \leftarrow N$
for all $n \in S_N : \overrightarrow{n} - \overleftarrow{n} > 100ms$ **do**
  Prosogram.addSubsegments($F0$, $\overleftarrow{n}$, $\overrightarrow{n}$, S)
end for
{Etap 3. Wyznaczenie etykiet}
$a \leftarrow \emptyset$
for $i = 0$ **to** $|S| - 1$ **do**
  $s \leftarrow S[i]$
  if $Prosogram.slopeSTs(F0, \overleftarrow{s}, \overrightarrow{s}) > G(\overrightarrow{s} - \overleftarrow{s})$ **then**
    if $i = 0$ **then**
      $l \leftarrow a_F(S_F(\overleftarrow{s}))$
    end if
    $r \leftarrow \overrightarrow{s}$
  else
    $m \leftarrow median(F0, s)$
    if $i = 0$ **then**
      $l \leftarrow m$
    end if
    $r \leftarrow m$
  end if
  $a \leftarrow a \cup \{\#S[i], (l, r)\}$
  $l \leftarrow r$
end for
return (S, a)

Algorytmy pomocnicze stosowane w algorytmach A.9 oraz A.10:

Algorytm A.10 Tworzenie segmentacji w fonetycznej analizie tonalnej Prosogram (Mertens 2009).

```
1: Prosogram.addSubegments(F: Annotation, t0: Real, t1: Real, S: Segmentation)
Wejście F: ramkowa, sygnałowa anotacja tonalna (przebieg F0)
Wejście s: segment
Wejście S: segmentacja, do której dodawane są segmenty wyjściowe
2: sub ← 0
3: if Prosogram.slopeSTs(F, t0, t1) > G(t1 − t0) then
4:    (SF, aF) ← F
5:    f0 ← aF(SF(t0))
6:    f1 ← aF(SF(t1))
7:    b ← (f1 − f0)/(t1 − t0)
8:    t ← argmax_{t0⩽t<t1} |aF(SF(t)) − f0 + b ∗ (t − t0)|
9:    if t − t0 > 35ms ∧ t1 − t > 35ms then
10:      g0 = Prosogram.slopeSTs(F0, t0, t)
11:      g1 = Prosogram.slopeSTs(F0, t, t1)
12:      if |g0 − g1| > DG then
13:         Prosogram.addSubsegments(F, t0, t, S)
14:         Prosogram.addSubsegments(F, t, t1, S)
15:         sub ← 1
16:      end if
17:   end if
18: end if
19: if sub = 0 then
20:   addSegment(S, t0, t1)
21: end if
```

- addSegment(Segmentation S, Real b, Real f) — dodaje do segmentacji S segment, którego kotwice wskazują na punkty czasowe b oraz f; algorytm zapewnia unikalność dodanych kotwic oraz segmentu w S.

- median(Annotation A, Segment s) — wylicza medianę rzeczywistych etykiet anotacji A przypisanych segmentom obejmowanym czasowo przez segment s.

- Prosogram.toLoudness(Signal x): Annotation — algorytm zwracający przebieg donośności sygnału x; wylicza przebieg donośności w sygnale x. W najprostrzym wariancie implementacji Prosogram.toLoudness jest stosowana funkcja programu Praat o nazwie „To Intensity...". „To Intensity..." zwraca anotację ramkową o czasie trwania segmentu 32 ms z krokiem 5 ms lub 10 ms, w której etykieta segmentu równa jest sumie kwadratów próbek sygnału segmentu (stosowane jest okno Gaussa). W bardziej rozbudowanych, percepcyjnych wariantach algorytmu Prosogram.toLoudness stosowana jest filtracja sygnału x filtrem środkowoprzespustowym (częstotliwości odcięcia 400Hz oraz 3500Hz) oraz (alternatywnie) przetworzenie sygnału do postaci cochleagramu (por. punkt 3.1.5 na stronie 38).

- Praat.extractF0(Signal x): Annotation — ekstrakcja F_0 programu Praat. Zastosowana metoda ekstrakcji F_0 jest oparta na periodogramie autokorelacyjnym oraz globalinej minimalizacji kosztu (Boersma 1993). Domyślny zakres F_0 wynosi 60 Hz do 450 Hz, krok segmentu analizy wynosi 5ms lub 10ms.

- Prosogram.slopeSTs($F0$:Annotation, t_0:Real, t_1:Real): Real — wyznacza średni wzrost

(spadek) częstotliwości podstawowej w przedziale czasowym $[t_0; t_1)$. Niech f_{max} [Hz] oraz f_{min} [Hz] oznaczają odpowiednio maksymalną oraz minimalną wartość F_0 zapisaną w etykietach anotacji $F0$ dla przedziału czasowego $[t_0; t_1)$. Wartością Prosogram.slopeSTs jest:

$$12\frac{log_2\frac{f_{max}}{f_{min}}}{t_1 - t_0}.$$

- Prosogram.syllabicNuclei(Annotation A, Annotation L):Annotation — dla danej anotacji sekwencyjnej A, której segmenty wskazują dopuszczalne lokalizacje ośrodków fonetycznych sylab oraz anotacji prostej L reprezentującej przebieg donośności algorytm Prosogram.syllabicNuclei zwraca anotację, której segmenty wskazują lokalizacje ośrodków fonetycznych sylab. Przebieg algorytmu zależy od rodzaju anotacji wejściowej A (segmentacja głoskowa, sylabiczna albo pusta). Za ośrodek fonetyczny uznawany jest przedział czasowy w którym następuje spadek o 3 lub 6 decybeli od maksymalnej intensywności (znalezionej w granicach samogłoski albo sylaby, jeśli dana jest segmentacja głoskowa albo sylabiczna). Przyjmowane jest ponadto, że minimalny czas trwania ośrodka wynosi 150 ms.

A.6 UHCF0C

Algorytm $A.11$ przedstawia przebieg UHCF0C.

Algorytm A.11 Tworzenie fonetycznego systemu tonalnego metodą UHCF0C (Lolive i inni 2007).

UHCF0C($\mathcal{V}^T$, $\mathcal{V}^V$, n)
Wejście $\mathcal{V}^T$: zbiór uczący (opis w tekście)
Wejście $\mathcal{V}^V$: zbiór walidujący (opis w tekście)
Wejście n: liczba HMM do podziału w każdym kroku algorytmu
Wyjście $\mathcal{M} = M_1, \ldots, M_{|\mathcal{M}|}$: zbiór wynikowych niejawnych modeli Markowa
$\mathcal{M} \leftarrow \{M_1\}$: zbiór niejawnych modeli Markowa
$a^T_{\mathcal{M}} \leftarrow$ViterbiAlignment$(\mathcal{M}, \mathcal{V}^T)$
$e_{prev} = +\infty$
$\epsilon = .0001$
converged=**false**
repeat
  for $i = 1$ **to** $|\mathcal{M}|$ **do**
    $M_i \leftarrow$ BaumWelshTraining$(M_i, \mathcal{V}^T, a^T_{\mathcal{M}})$
  end for
  $a^V_{\mathcal{M}} \leftarrow$ ViterbiClassification$(\mathcal{M}, \mathcal{V}^V)$
  $e_{cur} \leftarrow$ ViterbiRMS$(\mathcal{M}, \mathcal{V}^V, a^V_{\mathcal{M}})$
  if $e_{prev} - e_{cur} < \epsilon$ **then**
    converged = **true**
  else
    $\mathcal{M} \leftarrow$ HMMSplitting$(\mathcal{M}, n)$
    $a^T_{\mathcal{M}} \leftarrow$ ViterbiClassification$(\mathcal{M}, \mathcal{V}^T)$
  end if
  $e_{prev} = e_{cur}$
until converged=**true**
return $\mathcal{M}$

Parametry oraz algorytmy pomocnicze stosowane w algorytmie A.11:

- ϵ — próg zbieżności błędu dla warunku zatrzymania.
- BaumWelshTraining(M_i:HMM,$\mathcal{V}^T$:Set,$a^T_{\mathcal{M}}$:Function) — uczy M_i zmodyfikowanym algorytmem Bauma-Welsha (Huang i inni 2001, 389) wybierając ze zbioru $\mathcal{V}^T$ przebiegi etykietowane w anotacjach $a^T_{\mathcal{M}}$ liczbą i. Algorytm Bauma-Welsha zmodyfikowano tak, by stany q_{1i}, q_{2i} oraz q_{3i} (por. rys. 5.6) przypadały odpowiednio na nagłosie, ośrodku oraz wygłosie sylaby fonologicznej zgodnie z anotacjami zawartymi w $\mathcal{V}^T$.
- ViterbiClassification($\mathcal{M}$:Set,$\mathcal{V}$:Set):Function przyporządkowuje każdą sylabę opisaną (pośrednio) anotacjami struktur sylabicznych w zbiorze $\mathcal{V}$ do jednego z niejawnych modeli Markowa zawartych w zbiorze $\mathcal{M}$. O przyporządkowaniu do modelu M decyduje wartość prawdopodobieństwa ścieżki Viterbiego w modelu M (maksymalizacja). Funkcja wynikowa przyporządkowuje parze uporządkowanej $(A_F, A_S) \in \mathcal{V}$ anotację sylabiczną o etykietach ze zbioru $\{1, \ldots, |\mathcal{M}|\}$.
- ViterbiRMS($\mathcal{M}$:Set,$\mathcal{V}$:Set,$a_{\mathcal{M}}$:Function):Real — dla każdej sylaby w $\mathcal{V}$ wyznacza ścieżkę Viterbiego dla HMM ze zbioru $\mathcal{M}$ wskazywanego przez funkcję $a_{\mathcal{M}}$. Następnie wyznacza na zbiorze $\mathcal{V}$ błąd średniokwadratowy między przebiegiem F_0 dostępnym w anotacjach $\mathcal{V}$ oraz wartościami średnimi emisji stanów HMM w ścieżkach Viterbiego.
- HMMSplitting($\mathcal{M}$:Set, n:Integer):Set — dla danego zbioru HMM $\mathcal{M}$ tworzy wynikowy zbiór HMM o rozmiarze $|\mathcal{M}| + n$ poprzez podzielenie każdego z n elementów zbioru $\mathcal{M}$ na dwa, przy czym jeśli $\mu_i j$ oraz $\sigma_i j$ są średnią oraz odchyleniem standardowym dzielonego modelu, to za średnie rozkładu dwóch modeli po podziale przyjmuje się $\mu_{ij} \pm 0.001\sigma_{ij}$.

DODATEK B

Wizualizacje struktur intonacyjnych

W niniejszym dodatku przedstawiono wizualizacje wybranych struktur intonacyjnych otrzymanych subiektywnie (por. rozdział 9) oraz automatycznie na wyjściu układu SVM/CRF uczonego pod nadzorem i działającego w trybie indukcyjno–dedukcyjnym (por. sekcja 10.3.2). Sygnały analizowane automatycznie nie należały do zbioru uczącego układu analizy. Zastosowaną metodę wizualizacji opisano w sekcji 10.3.4 na stronie 165.

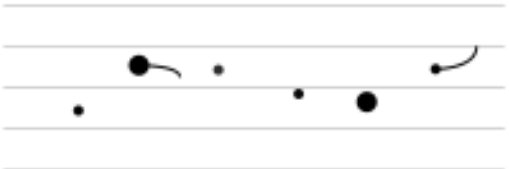
B.1 Głos „joju" z korpusu PoInt

1.

♡ \tyle że /każdy był z innej ‖pary

□ \tyle że /każdy był z innej ‖pary

Kolejno melodia „silna opadająca nad", melodia „silna rosnąca nad" oraz „rdzenna opadająca niska".



2.

♡ (–)w zes\tawie po‖dobnym

□ (–)w zes\tawie po‖dobnym

Kolejno melodia „słaba równa pod”, melodia „silna opadająca nad” oraz „rdzenna rosnąca niska”. W anotacji automatycznej melodia rdzenna została błędnie rozpoznana jako „rdzenna rosnąca pełna”.

3.

♡ \każdy od\innej//pary

□ \każdy od\innej//pary

Kolejno melodia „silna opadająca pod”, melodia „silna opadająca pod” oraz „rdzenna rosnąca wysoka”. W anotacji automatycznej melodia rdzenna została błędnie sklasyfikowana jako „rdzenna rosnąca pełna”. (Ostatnia sylaba nie jest widoczna w transkrypcji tonatycznej ze względu na niską harmoniczność.)

4.

♡ (–)za\łożył na\siebie te//dwa

□ (–)za\łożył na siebie te//dwa

Kolejno melodia „słaba równa pod”, „silna opadająca pod”, „silna opadająca nad” oraz „rdzenna rosnąca niska”. W anotacji automatycznej nie został rozpoznany akcent «siebie» ale została zachowana poprawność strukturalna poprzez zmianę klasyfikacji akcentu «założył».

5.

♡ \\//nie wiem dlaczego

□ \\//nie wiem dlaczego

Melodia „rdzenna opadająco–rosnąca”.

B.2 Głos „mawa” z korpusu PoInt

1.
♡ (/)no muszę coś z tym\\zrobić
□ (/)no muszę coś z tym\\zrobić

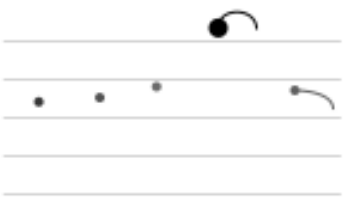
Kolejno melodia „słaba rosnąca pod” oraz „rdzenna opadająca pełna”.

2.
♡ \poszedł do domu je//zmienić
□ \poszedł do domu je//zmienić

Kolejno melodia „silna opadająca nad” oraz „rdzenna rosnąca pełna”.

3.
♡ /\długo nie przy//chodzi
□ \długo nie przy//chodzi

Kolejno melodia „silna rosnąco–opadająca nad” oraz „rdzenna rosnąca niska”. W anotacji automatycznej błędnie sklasyfikowany akcent «długo» jako „silny opadający nad”.



4.
♡ (/)że on je\\zmienił
□ (/)że on je//\\zmienił

Kolejno melodia „słaba rosnąca pod” oraz „rdzenna opadająca pełna”. W anotacji automatycznej błędnie sklasyfikowany akcent rdzenny jako „rosnąco–opadający”.

5.
♡ \kojarzą mi się\z pełnym//domem
□ \kojarzą mi się\z pełnym//domem

Kolejno melodia „silna opadająca pod”, „silna opadająca pod” oraz „rdzenna rosnąca pełna”.

B.3 Głos „pano” z korpusu PoInt

1.
♡ \widać po nim//wiosnę
□ \widać po nim//wiosnę

Kolejno melodia „silna opadająca pod” oraz „rdzenna rosnąca pełna”.

2.
♡ (–)ok\ropne wido//\\wiska
□ (–)ok\ropne wido\\wiska

Kolejno melodia „słaba równa pod”, „silna opadająca pod” oraz „rdzenna rosnąco–opadająca”. W anotacji automatycznej melodia rdzenna błędnie została rozpoznana jako „rdzenna opadająca pełna”.

3.

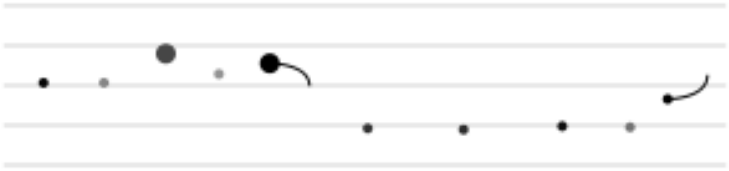
♡ $_{(-)}$który tu jest$^{/}$poka\\//zany
□ $_{(-)}$który tu jest^{-}poka\\zany

Kolejno melodia „słaba równa pod", „silna rosnąca nad" oraz „rdzenna opadająco–rosnąca". W anotacji automatycznej błędnie rozpoznano melodię „silną równą nad" oraz „rdzenną opadającą pełną".

4.

♡ $^{(-)}$op\isać\\obraz
□ $_{(-)}$op\isać\\obraz

Kolejno melodia „słaba równa nad", „silna opadająca pod" oraz „rdzenna opadająca niska". W anotacji automatycznej błędnie rozpoznana „słaba równa pod" oraz „rdzenna opadająca pełna".



5.

♡ $_{(-)}$no tu\kilka \\//planów jest właściwie
□ $_{(-)}$no tu\kilka \\//planów jest właściwie

Kolejno melodia „słaba równa pod", „silna opadająca pod" oraz „rdzenna opadająco–rosnąca".

Bibliografia

Ahmadi S. i Spanias A.S., 1999. Cepstrum-based pitch detection using a new statistical V/UV classification algorithm. *IEEE Transactions on Speech Audio Processing*, tom 7:333–338.

Alain P., Barbot N., Barreaud V., Blin L., Boeffard O., Charonnat L., Choumane A., Delhay A., Maguer S.L., Lolive D., Moudenc T. i Vidal G., 2009. A multi-agent platform for multimodal pervasive applications. W *2009 NEM Summit – Towards Future Media Internet*. St Malo's Palais du Grand Large, France.

Allen J., Hunnicutt M.S. i Klatt D., 1987. *From text to speech. The MITalk system.* Cambridge University Press, Cambridge.

Ananthakrishnan S. i Narayanan S., 2005. An automatic prosody recognizer using a coupled multi-stream acoustic model and a syntactic-prosodic language model. W *Proceedings of International Conference on Acoustics Speech and Signal Processing*. IEEE.

Ananthakrishnan S. i Narayanan S., 2008. Automatic prosody labeling using acoustic, lexical, and syntactic evidence. *IEEE Transactions on Speech, Audio and Language Processing*, tom 16.

Ananthakrishnan S. i Narayanan S., 2009. Unsupervised adaptation of categorical prosody models for prosody labeling and speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, tom 17:138–149.

Atal B. i Hanauer S.L., 1971. Speech analysis and synthesis by linear prediction of the peech wave. *Journal of the Acoustical Society of America*, tom 50:637–655.

Auran C., Bouzon C. i Hirst D., 2004. The Aix-MARSEC project: An evolutive database of spoken British English. W PROSODY'04.

Bagshaw P., 1993. An investigation of acoustic events related to sentential stress and pitch accents. *Speech Communication*, tom 13:333–342.

Bagshaw P., Hiller S. i Jack M., 1993. Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching. W EUROSPEECH'93, str. 1003–1006.

Bard E.G., Sotilloa C., Andersonb A.H., Thompsona H.S. i Taylor M.M., 1996. The DCIEM map task corpus: Spontaneous dialogue under sleep deprivation and drug treatment. *Speech Communication*, tom 20:71–84.

Barry W.J. i van Dommelen W.A., 2005. *The Integration of Phonetic Knowledge in Speech Technology*. Springer, Dordrecht, The Netherlands.

Batliner A. i Möbius B., 2005. *Prosodic Models, Automatic Speech Understanding, and Speech Synthesis: Towards the Common Ground*. W Barry i van Dommelen (2005).

Batóg T., 1967. *The Axiomatic Method in Phonology*. Routledge and Kegan Paul, London.

Batóg T., 1994. *Studies in Axiomatic Foundations of Phonology*. Wydawnictwo Naukowe UAM, Poznań.

Beckman M.E., Hirschberg J. i Shattuck-Hufnagel S., 2005. *The original ToBI system and the evolution of the ToBI framework*. W Jun (2005).

Benesty J., Sondhi M.M. i Huang Y., red., 2008. *Springer Handbook of Speech Processing*. Springer-Verlag.

Białasiewicz J.T., 2000. *Falki i aproksymacje*. Wydawnictwa Naukowo-Techniczne, Warszawa.

Bird S. i Liberman M., 2001. A formal framework for linguistic annotation. *Speech Communication*, tom 33:23–60.

Bird S., Ma X. i Lee H., 2007. AGLIB Java 1.0. [2010-07-20].
URL {http://agtk.sourceforge.net/}

Bleeck S., Ives T. i Patterson R., 2004. Aim-mat: the auditory image model in MATLAB. *Acta Acoustica*, tom 90.

Bloch B., 1948. A set of postulates for phonemic analysis. *Language*, tom 24.

Boersma P., 1993. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proceedings of the Institute of Phonetic Sciences*, tom 17:97–110.

Boersma P., 1998. *Functional Phonology. Formalizing the interactions between articulatory and perceptual drives.* Holland Academic Graphics, The Hague.

Boersma P. i Weenink D., 1996. *Praat, a system for doing phonetics by computer, version 3.4.* Institute of Phonetic Sciences of the University of Amsterdam, Report 132.

Boersma P. i Weenink D., 2008. Praat: doing phonetics by computer (version 5.0.26). [2008-06-24].
URL {http://www.praat.org/}

Bogert B., Healy M. i Tukey J., 1963. The quefrency analysis of time series for echos. W *Proceedings of the Symposium on Time Series Analysis*, str. 209–243. Wiley, New York.

Botinis A., Kouroupetroglou G. i Carayiannis G., red., 1997. *Intonation: Theory, Models and Applications*. European Speech Communication Association, Athens, Greece.

Bottou L., 1991. Stochastic gradient learning in neural networks. W *Proceedings of Neuro-Nîmes 91*. Nimes, France.

Bottou L., 2011. Stochastic gradient descent examples on toy problems.
URL {http://leon.bottou.org/projects/sgd}

Braunschweiler N., 2006. Prosodizer – automatic prosodic annotations of speech synthesis databases. W PROSODY'06.

Brenier J.M., Cer D.M. i Jurafsky D., 2005. The detection of emphatic words using acoustic and lexical features. W INTERSPEECH'05.

Breuer S., Wagner P., Abresch J., Bröggelwirth J., Rohde H. i Stöber K., 2005. Bonn open synthesis system (boss) 3. documentation and user manual.

Brown J., 1992. Musical fundamental frequency tracking using a pattern recognition method. *Journal of the Acoustical Society of America*, tom 92:1394–1402.

Brugos A., Shattuck-Hufnagel S. i Veilleux N., 2006. Transcribing prosodic structure of spoken utterances with ToBI. [2010-03-26].
URL {http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-911January--IAP--2006/LectureNotes/index.htm}

Byrd R., Nocedal J. i Schnabel R., 1994. Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, tom 63(4):129–156.

Campbell N., 1992. Syllable-based segmental duration. W *In Talking Machines: Theories, Models and Designs*, pod redakcją C.B.G. Bailly i T.R. Sawallis. Elsevier Science Publishers, Amsterdam.

Campbell N., 2006. On the structure of spoken language. W PROSODY'06.

Carletta J., 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, tom 22(2).

Carlson R., Elenius K. i Swerts M., 2004. Perceptual judgments of pitch range. W PROSODY'04.

Cassidy S. i Harrington J., 2001. Multi-level annotation in the Emu speech database management system. *Speech Communication*, tom 33:61–77.

Chan D., Fourcin A., Gibbon D., Granstrom B., Huckvale M., Kokkinakis G., Kvale K., Lamel L., Lindberg B., Moreno A., Mouropoulos J., Senia F., Trancoso I., Veld C. i Zeiliger J., 1995. EUROM – a spoken language resource for the EU. W EUROSPEECH'95, str. 867–870.

Chang C. i Lin C.J., 2001. LIBSVM: a library for support vector machines.
URL {http://www.csie.ntu.edu.tw/~cjlin/libsvm}

Chen P.H., Lin C.J. i Schölkopf B., 2005. A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry*, tom 21.

Chen Y., Lai M., Chu M., Soong F.K., Zhao Y. i Hu F., 2006. Automatic accent annotation with limited manually labeled data. W PROSODY'06.

Chisaki Y., Nakashima H., Shiroshita S., Usagawa T. i Ebata M., 2003. A pitch detection method based on continuous wavelet transform for harmonic signal. *Acoustical Science and Technology*, tom 24:7–16.

Clarkson P. i Rosenfeld R., 1997. Statistical language modeling using the cmu-cambridge toolkit from proceedings esca eurospeech 1997. W EUROSPEECH'97.

Cohen M., Grossberg S. i Wyse L., 1995. *A spectral network model of pitch perception. Technical Report.* Boston University.

Cooley J.W. i Tukey J.W., 1965. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, tom 297(19).

Corkill D.D., 2003. Collaborating software: Blackboard and multi-agent systems and the future. W *Proceedings of the International Lisp Conference*. New York.

Cormen T., Rivest C.L.R. i Stein C., 2004. *Wprowadzenie do algorytmów*. WNT, Warszawa.

Cosi P., Pasquin S. i Zovato E., 1998. Auditory modeling techniques for robust pitch extraction and noise reduction. W Mannell i Robert-Ribes (1998). [2008-01-05]. URL {http://andosl.anu.edu.au/icslp98/main.html}

Cruttenden A., red., 1997. *Intonation*. Cambridge University Press.

Crystal D., 1969. *Prosodic systems and intonation in English*. CUP, Cambridge.

Cunningham H., 2000. *Software Architecture for Language Engineering*. Rozprawa doktorska, University of Sheffield.

Cunningham H., Maynard D., Bontcheva K. i Tablan V., 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. W *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

d'Allesandro C. i Mertens P., 1995. Automatic pitch stylization using a model of tonal perception. *Computer Speech and Language*, tom 9(3):257–288.

de Cheveigné A., 2005. Pitch perception models. W *Pitch. Neural Coding and Perception*, rozdz. "6". Springer.

de Cheveigné A. i Kawahara H., 2002. YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, tom 111:1917–1930.

Della Pietra S., Della Pietra V. i Lafferty J., 1997. Inducing features of random fields. *IEEE Transactions on Patterns Analysis and Machine Intelligence*, tom 19:380–393.

Demenko G., 1999. *Analiza cech suprasegmentalnych języka polskiego na potrzeby technologii mowy*. Wydawnictwo Naukowe UAM, Poznań.

Demenko G., Grocholewski S., Wagner A. i Szymański M., 2006. Prosody annotation for corpus based speech synthesis. W *Proceedings of the 11th Australian International Conference on Speech Science and Technology*. Auckland, New Zeland.

Demenko G. i Jassem W., 1999. Modelling intonational phrase structure with Artificial Neural Networks. W EUROSPEECH'99.

Demenko G. i Wagner A., 2006. The stylization of intonation contours. W PROSODY'06.

D'Imperio M., 2002. Italian intonation: An overview and some questions. *Probus*, tom 14(1):37–69.

Dogil G., 1995. Stress patterns in West Slavic languages. *Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung*, tom 2(2):61–88.

Duch W., Korbowicz J., Rutkowski L. i Tadeusiewicz R., red., 2000. *Sieci neuronowe*. Akademicka Oficyna Wydawnicza EXIT, Warszawa.

Dudley H., 1935. Signal transmission. united states patent 2,151,091.

Duifhuis H., Willems L. i Sluyter R., 1979. Pitch in speech: A hearing theory approach. *Journal of the Acoustical Society of America*, tom 65.

Durand P., Durand-Deska A., Gubrynowicz R. i Marek B., 2002. Polish: Prosodic aspects of „Czy" questions. W PROSODY'02.

Dutoit T., 1997. *An Introduction to Text-to-Speech Synthesis*. Kluwer Academic Publishers, Dordrecht.

Dutoit T., 2008. *Corpus-Based Speech Synthesis*. W Benesty i inni (2008).

Dziubalska-Kołaczyk K., 2002. *Beats-and-binding phonology*. Beats-and-binding phonology. Peter Lang Verlag, Frankfurt/Main.

Dziubalska-Kołaczyk K., Cole R.A., Krynicki G., Wypych M., Pellom B., Ma J., Struemph T., Sobkowiak W. i Bogacka A., 2004. The use of metalinguistic knowledge in a polish literacy tutor. W *GlobE 2004*. Peter Lang, Warsaw.

Erman L., Hayes-Roth F., Lesser V. i Reddy D.R., 1980. The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, tom 12(2).

EUROSPEECH'03, 2003. *Proceedings of Eurospeech*. Geneva.

EUROSPEECH'93, 1993. *Eurospeech'93*. Berlin.

EUROSPEECH'95, 1995. *Proceedings of Eurospeech*. Madrid.

EUROSPEECH'97, 1997. *Proceedings of Eurospeech*. Rhodos.

EUROSPEECH'99, 1999. *Proceedings of Eurospeech*. Budapest, Hungary.

Fach M. i Wokurek W., 1995. Pitch accent classification of fundamental frequency contours by Hidden Markov Models. W EUROSPEECH'95.

Fant G., 1960. *Acoustic theory od speech production*. Mouton, The Hague, Netherlands.

Fastl H. i Zwicker E., 2007. *Psychoacoustics*. Springer-Verlag, Berlin Heidelberg.

Fourcin A.J., 1974. Laryngographic examination of vocal fold vibration. W *Ventilatory and Phonatory Control Systems*, pod redakcją B. Wyke, str. 315–333. Oxford University Press.

Fowler M., 2004. Inversion of control containers and the dependency injection pattern. [2009-12-01].
URL `{http://martinfowler.com/articles/injection.html}`

Fraley C. i Raftery A.E., 2002. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, tom 97:611–631.

Fraley C. i Raftery A.E., 2006. MCLUST version 3 for r: Normal mixture modeling and model-based clustering. technical report n. 504.

Francuzik K., Karpiński M., Kleśta J. i Szalkowska E., 2005. Nuclear melody in Polish semi-spontaneous and read speech: Evidence from Polish intonational database 'PoInt'. *Studia Phonetica Posnaniensia*, tom 7.

Frydrychowicz S., 1999. *Proces mówienia.* Wydawnictwo Naukowe UAM.

Fujisaki H., 2000. The physiological and physical mechanisms for controlling the tonal features of speech in various languages. W PROSODY'00.

Fujisaki H., 2004. Information, prosody and modeling with emphasis on tonal features of speech. W PROSODY'04.

Fujisaki H. i Hirose K., 1984. Analysis of voice fundamental frequency contours for declarative sentences of Japanese. *J. Acoust. Soc. Japan*, tom 5(4):233–242.

Fujisaki H. i Nagashima S., 1969. A model for the synthesis of pitch contours of connected speech.

Gerhard D., 2003. Pitch extraction and fundamental frequency: History and current techniques.

Gibbon D., 2006. *Time Types and Time Trees.* W Sudhoff i inni (2006).

Gold B., 1999. *Speech and Audio Signal Processing. Processing and Perception of Speech and Music.* John Wiley and Sons, Inc., New York.

Grabe E. i Karpiński M., 2003. Universal and language-specific aspects of intonation in English and Polish. W ICPhS'03.

Grabe E., Post B. i Nolan F., 2000. Modelling intonational variation in English. the IViE system. W PROSODY'00.

Greenbaum S. i Svartvik J., 1990. The London corpus of spoken English: Description and research. *Lund Studies in English*, tom 82.

Grice M., Leech G., Weisser M. i Wilson A., 2000. Representation and annotation of dialogue. W *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation*, pod redakcją D. Gibbon, I. Mertins i R. Moore. Kluwer.

Grice M., Reyelt M., Benzmüller R., Mayer J. i Batliner A., 1996. Consistency in transcription and labelling of German intonation with GToBI. W ICSLP'96.

Grützmacher M. i Lottermoser W., 1937. Über ein verfahren zur trägheitsfreien aufzeichnung von melodiekurven. tom 2:242–248.

Gubrynowicz R., 1999. *Projekt i realizacja bazy danych mowy polskiej w programie BABEL*, str. 257–276. Tom 3 z serii Jassem i inni (1999).

Gubrynowicz R., Mikiel W. i Żarnecki P., 1980. Acoustical analysis for evaluation of laryngeal dysfunction in case of vocal chords paralysis. *Speech analysis and synthesis*, tom 5.

Gussenhoven C., 1994. *The phonology of tone and intonation.* Cambridge University Press, Cambridge.

Gussenhoven C., 2002. Intonation and interpretation: Phonetics and phonology. W PROSODY'02.

Gussenhoven C., 2004. *The Phonology of Tone and Intonation*. Cambridge University Press, Cambridge.

Halliday M.A.K., 1967. *Intonation and Grammar in British English*. Oxford University Press, London.

Hammersley J. i Clifford P., 1971. Markov fields on finige graphs and lattices.

Hasegawa-Johnson M., Chen K., Cole J., Borys S., suk Kim S., Cohen A., Zhang T., yoon Choi J., Kim H., Yoon T. i Chavarria R., 2005. Simultaneous recognition of words and prosody in the boston university radio speech corpus. *Speech Communication*, tom 46:418–439.

Hedelin P., 1984. A glottal LPC vocoder. W *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. Sydney.

Hermansky H., 1990. Perceptual linear predictive (plp) analysis of speech. *Journal of the Acoustical Society of America*, tom 87(4).

Hermes D.J., 1988. Measurement of pitch by subharmonic summation. *Journal of the Acoustical Society of America*, tom 83:257–264.

Hermes D.J., 2006. Stylization of pitch contours. W Sudhoff i inni (2006), str. 29–62.

Hertz S., 1988. The Delta programming language: an integrated approach to nonlinear phonology, phonetics and speech synthesis. *UCLA Working Papers in Phonetics*, tom 69.

Hess W., 1983. *Pitch Determination of Speech Signals*. Springer-Verlag, Berlin Heidelberg.

Hess W.J., 2008. *Pitch and Voicing Determiniation of Speech with an Extension Toward Music Signals*. W Benesty i inni (2008).

Hirschberg J. i Nakatani C., 1998. Using machine learning to identify intonational segments. W *Proceedings of the AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*. Palo Alto.

Hirst D., 2000. Optimising the INTSINT coding of F0 targets for multi-lingual speech synthesis. W PROSODY'00.

Hirst D., 2007. A praat plugin for Momel and INTSINT with improved algorithms for modelling and coding intonation. W ICPhS'07.

Hirst D. i Buzon C., 2005. The effect of stress and boundaries on segmental duration in a corpus of authentic speech (British English). W INTERSPEECH'05.

Hirst D., Cristo A.D. i Espesser R., 2000. Levels of representation and levels of analysis for the description of intonation systems. W *Prosody: Theory and Experiment*, pod redakcją M. Horne, rozdz. 3. Kluwer Academic Publishers.

Hirst D. i Espesser R., 1993. Automatic modelling of fundamental frequency using a quadratic spline function. W *Travaux de l'Institut de Phonetique d'Aix*, str. 75–85.

Hirst D.J. i Cristo A.D., 1998. A survey of intonation systems. W *Intonation Systems: a Survey of Twenty Languages*, pod redakcją D.J. Hirst i A.D. Cristo. Cambridge University Press, Cambridge.

Hoefel G. i Elkan C., 2008. Learning a two-stage svm/crf sequence classifier. W *ACM 17th Conference on Information and Knowledge Management*. Napa Valley, California.

Hofmann T., Schölkpf B. i Smola A.J., 2008. Kernel methods in machine learning. *The Annals of Statistics*, tom 36(3).

Hoschek W., 2004. Colt. [2010-09-09].
URL {http://acs.lbl.gov/software/colt}

Hsu C.W., Chang C.C. i Lin C.J., 2010. A practical guide to support vector classification. *Bioinformatics*, tom 1.

Huang X., Acero A. i Hon H.W., 2001. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall PTR, New Jersey.

Huckvale M., Brookes D., Dworkin L., Johnson M., Pearce D. i Whitaker L., 1987. The spar speech filing system. W *European Conference on Speech Technology*. Edinburgh.

ICASSP'02, 2002. *Proceedings of International Conference on Acoustics Speech and Signal Processing*. IEEE.

ICASSP'82, 1982. *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*. Paris.

ICPhS'03, 2003. *15th International Congress of Phonetic Sciences*. Barcelona.

ICPhS'07, 2007. *16th International Congress of Phonetic Sciences*. Saarbrücken.

ICSLP'00, 2000. *6th International Conference of Spoken Language Processing*. Beijing, China.

ICSLP'94, 1994. *Proceedings of International Conference of Spoken Language Processing*. Yokohama, Japan.

ICSLP'96, 1996. *Proceedings of International Conference of Spoken Language Processing*. Philadelphia.

Inanoglu Z. i Young S., 2005. Intonation modelling and adaptation for emotional prosody generation. W *1st Intnl Conf on Affective Computing and Intelligent Interaction*. Springer-Verlag GmbH, Beijing.

INTERSPEECH'05, 2005. *Proceedings of Interspeech 2005*. Lisbon, Portugal.

INTERSPEECH'07, 2007. *Proceedings of Interspeech 2007*. Antwerp, Belgium.

INTERSPEECH'09, 2009. *Proceedings of Interspeech*. Brighton, UK.

IPA, 2005. Reproduction of the international phonetic alphabet. [2009-01-10].
URL {http://www.langsci.ucl.ac.uk/ipa/ipachart.html}

IPDS K., 1995. The Kiel corpus of spontaneous speech.

Ishi C.T., 2004. Analysis of autocorrelation-based parameters for creaky voice detection. W PROSODY'04.

Ito K., Speer S.R. i Beckman M.E., 2004. Informational status and pitch accent distribution in spontaneous dialogues in English. W PROSODY'04.

Jassem K., 1996a. A phonemic transcription and syllable division rule engine. W *Onomastica-Copernicus Research Coloquium*. Edinburgh.

Jassem K., 2002a. Transfer w systemie POLENG-3. W *Speech and Language Technology*, pod redakcją G. Demenko, M. Karpiński i K. Jassem, tom 6. Polskie Towarzystwo Fonetyczne.

Jassem K., Graliński F., Wagner A. i Wypych M., 2006. Text normalization as a special case of machine translation. W *Proceedings of the XXII International Multiconference on Computer Science and Information Technology*, tom 1. Wisła, Poland.

Jassem W., 1956. Węzłowe zagadnienia fonematyki. *Biuletyn Polskiego Towarzystwa Językoznawczego*, tom 15:13–30.

Jassem W., 1962. *Akcent języka polskiego*. Wydawnictwo Polskiej Akademii Nauk.

Jassem W., 1973. *Podstawy fonetyki akustycznej*. Państwowe Wydawnictwo Naukowe, Warszawa.

Jassem W., 1974. *Mowa a nauka o łączności*. Państwowe Wydawnictwo Naukowe, Warszawa.

Jassem W., 1983. *The Phonology of Modern English*. Państwowe Wydawnictwo Naukowe, Warszawa.

Jassem W., 1984. Isochrony in English speech. W *Intonation, Accent and Rhythm*, pod redakcją D. Gibbon i H. Richter. de Gruyter, Berlin.

Jassem W., 1996b. A quantitative analysis of Standard British-English nuclear tones. *Journal of Quantitative Liguistics*, tom 3:239–243.

Jassem W., 1999. *English Stress, Accent and Intonation revisited*, str. 33–50. Tom 3 z serii Jassem i inni (1999).

Jassem W., 2002b. Classification and organization of data in intonation research. W *Phonetics and its Applications*, pod redakcją A. Braun i H. Masthoff. Franz Steiner Verlag, Wiesbaden.

Jassem W., 2003a. Gramatyka intonacyjna języka polskiego.

Jassem W., 2003b. Polish. *Journal of the International Phonetic Association*, tom 34(1):103–107.

Jassem W., 2003c. Reguły podziału tekstu fonetycznego na sylaby.

Jassem W., 2007. Kryteria dystynktywności w fonetyce. [komunikacja prywatna].

Jassem W., Basztura C., Demenko G. i Jassem K., red., 1999. *Speech and Language Technology*, tom 3. Polskie Towarzystwo Fonetyczne, Poznań.

Jassem W. i Kudela-Dobrogowska K., 1980. Speaker-independent intonation curves. W *The Melody of Language*, pod redakcją L.R. Waugh i C.H. Schooneveld, rozdz. 10. University Park Press, Baltimore.

Jensen U., Moore R., Dalsgaard P. i Lindberg B., 1993. Modelling of intonation contours at the sentence level using CHMMS and the 1961 O'Connor and Arnold scheme. W EUROSPEECH'93.

Jeon J.H. i Liu Y., 2009. Semi-supervised learning for automatic prosodic event detection. W *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, str. 540–548. ACL, Singapore.

Jun S.A., red., 2005. *Prosodic Typology. The Phonology of Intonation and Phrasing.* Oxford University Press.

Jurafsky D. i Martin J.H., 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall PTR, Upper Saddle River, NJ, USA.

Karpiński M., 2002. The corpus of the Polish intonational database: Technical specifications. *Investigationes Linguisticae*, tom VIII.

Karpiński M., 2006. *Struktura i intonacja polskiego dialogu zdaniowego.* Wydawnictwo Naukowe UAM, Poznań.

Kasi K. i Zahorian S., 2002. Yet Another Algorithm for Pitch Tracking. W ICASSP'02.

Katz S.M., 1987. Estimation of probabilities from sparse data for the language model component of a speech recogniser. *IEEE Transactions on Acoustics, Speech and Signal Processing*, tom 35(3):400–401.

Kiessling A., Kompe R., Batliner A., Niemann H. i Nöth E., 1994. Automatic labeling of phrase accents in German. W ICSLP'94.

Klabbers E., Stöber K., Veldhuis R., Wagner P. i Breuer S., 2001. Speech synthesis development made easy: The Bonn Open Synthesis System.

Klatt D. i Klatt L., 1990. Analysis, synthesis and perception of voice quality variations among male and female talkers. *Journal of the Acoustical Society of America*, tom 87:820–856.

Klatt D.H., 1979. Synthesis by rule of segmental durations in English sentences. W *Frontiers of Speech Communication Research*, pod redakcją B. Lindblom i S. Ohman. Academic Press, New York.

Klinghardt H. i Klemm G., 1920. *Übungen im englischen Tonfall.*

Klüter A., Ndiaye A. i Kirchmann H., 2000. *Verbmobil From a Software Engineering Point of View: System Design and Software Integration.* Springer, Berlin.

Kochanski G., Grabe E., Coleman J. i Rosner B.S., 2005. Loudness predicts prominence: Fundamental frequency lends little. *Journal of the Acoustical Society of America*, tom 118(2):1038–1054. [2008-06-10].
URL `{http://kochanski.org/gpk/apers/2005/04pnp.pdf}`

Kohler K., 1991. A model of German intonation. *Arbeitsberechte des Instituts für Phonetik der Uniwersität Kiel*, tom 25:115–185.

Kohler K., 1997. *Modelling Prosody in Spontaneous Speech.* W Sagisaka i inni (1997).

Kohler K., 2006. *Paradigms in Experimental Prosodic Analysis*, str. 123–152. W Sudhoff i inni (2006).

Kohler K.J., 2003. Neglected categories in the modelling of prosody: pitch timing and non-pitch accents. W ICPhS'03.

Kohonen T., 2001. *Self-Organizing Maps.* Springer, Berlin.

Kornacki J. i Ćwik J., 2008. *Statystyczne systemy uczące się.* Akademicka Oficyna Wydawnicza EXIT, Warszawa.

Krzyśko M., Wołyński W., Górecki T. i Szkorybut M., 2008. *Systemy uczące się.* Wydawnictwa Naukowo-Techniczne, Warszawa.

Kubzdela H., 1976. An analogue fundamental frequency extractor. *Speech analysis and synthesis*, tom 4.

Kumar V., Sridhar R., Nenkova A., Narayanan S. i Jurafsky D., 2008. Detecting prominence in conversational speech: pitch accent, givenness and focus. W PROSODY'08.

Ladd R., 1996. *Intonational Phonology.* Cambridge University Press, Cambridge.

Lafferty J., McCallum A. i Pereira F., 2001. Contitional random fields: Probabilistic models for segmenting and labeling sequance data. W *The Eighteenth International Conference on Machine Learning.* Berkshires.

Leemann A. i Siebenhaar B., 2008. Swiss Alpine and Midland intonation. W PROSODY'08.

Lehiste I., 1976. Suprasegmental features of speech. W *Contemporary Issues in Experimental Phonetics*, pod redakcją N. Lass, str. 256–289. Academic Press, New York.

Levinson S.E., 2005. *Mathematical Models for Speech Technology.* John Wiley and Sons Ltd., Chichester.

Levow G., 2006. Unsupervised and semi-supervised learning of tone and pitch accent. W *Proceedings of HLT-NAACL.*

Levow G.A., 2008. Automatic prosodic labeling with conditional random fields. W *The Third International Joint Conference on Natural Language Processing.* Hyderabad, India.

Lolive D., Barbot N. i Boeffard O., 2007. Unsupervised HMM classification of F0 curves. W INTERSPEECH'07.

Lyon R., 1982. A computational model of filtering, detection, and compression in the cochlea. W ICASSP'82.

Łobacz P., 1999. Problems of automatic phonematic transcription of contemporary polish proper names. *Seria Językoznawcza*, tom 22.

Madejowa M., 1989. Zasady współczesnej wymowy polskiej. *Biuletyn Audiofonologii*, tom 2-4.

Maia R., Toda T., Zen H., Nankaku Y. i Tokuda K., 2007. An excitation model for HMM-based speech synthesis based on residual modeling. W *Sixth ISCA Tutorial and Research Workshop on Speech Synthesis (SSW6)*, pod redakcją P. Wagner, J. Abresch, S. Breuer i W. Hess. Bonn, Germany.

Mallat S.G., 1999. *A wavelet tour of signal processing.* Academic Press.

Mannell R.H. i Robert-Ribes J., red., 1998. *Proceedings of International Conference of Spoken Language Processing.* Sydney, Australia. [2008-01-05].
URL {http://andosl.anu.edu.au/icslp98/main.html}

Marasek K., 1997. EGG and voice quality. [2006-04-01].
URL {http://www.ims.uni-stuttgart.de/phonetik/EGG/frmstt.htm}

Marasek K. i Gubrynowicz R., 2005. Multilevel annotation in SpeeCon Polish speech database. W *Intelligent media technology for communicative intelligence. Second International Workshop, IMTCI 2004*, pod redakcją L. Bolc, Z. Michalewicz i T. Nishida. Springer, Berlin.

Margolis A., Livescu K. i Ostendorf M., 2010a. Domain adaptation with unlabeled data for dialog act tagging. W *Workshop on Domain Adaptation for Natural Language Processing*.

Margolis A., Ostendorf M. i Livescu K., 2010b. Cross-genre training for automatic prosody classification. W *Proceedings of Speech Prosody 2010*. Chicago, IL.

Martin P., 1982. Comparison of pitch detection by cepstrum and spectral comb analysis. W ICASSP'82.

Martin P., 1987. A logarithmic spectral comb method for fundamental frequency analysis.

Matthews G.G., 2000. *Neurobiologia*. Wydawnictwo Lekarskie PZWL.

Mayers N.C., 1995. Traits: a new and useful template technique. *C++ Report*, tom 6.

Medan Y., Yair E. i Chazan D., 1991. Super resolution pitch determination of speech signals. *IEEE Transactions on Speech Processing*, tom 39(1):40–48.

Meddis R. i Hewitt M.J., 1991. Virtual pitch and phase sensitivity of a computer model of the auditory periphery. *Journal of the Acoustical Society of America*, tom 89:2866–2882.

Meddis R. i O'Mard L.P., 2006. Virtual pitch in a computational physiological model. *Journal of the Acoustical Society of America*, tom 120(6):3861–3869.

Mehnert D. i Hoffmann R., 2006. Measuring pitch with historic phonetic devices. W PROSODY'06.

Mermelstein P., 1976. Distance measures for speech recognition — psychological and instrumental. W *Joint Workshop on Pattern Recognition and Artificial Intelligence*, pod redakcją C.H. Chen. Hyannis, Mass.

Mertens P., 2004. The Prosogram : Semi-automatic transcription of prosody based on a tonal perception model. W PROSODY'04.

Mertens P., 2009. The Prosogram, v2.4f: Transcription of prosody using pitch contour stylization based on a tonal perception model and automatic segmentation. [2009-01-13].
URL {http://bach.arts.kuleuven.be/pmertens/prosogram/}

Mishra T., van Santen J. i Klabbers E., 2006. Decomposition of pitch curves in the general superpositional intonation model. W PROSODY'06.

Mixdorff H., 2000. A novel approach to the fully automatic extraction of Fujisaki model parameters. W *Proceedings of International Conference on Acoustics Speech and Signal Processing*. IEEE, Istambul, Turkey.

Möhler G. i Conkie A., 1998. Parametric modeling of intonation using vector quantization. W WSS'98.

Nakatani C.H., Grosz B. i Hirschberg J., 1995. Discourse structure in spoken language; studies on speech corpora. W *Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*. American Association for Artificial Intelligence, Palo Alto.

Nolan F. i Grabe E., 1997. *Can 'ToBI' transcribe intonational variation in British English*. W Botinis i inni (1997).

Nooteboom S., 1997. The prosody of speech: Melody and rhythm. W *Handbook of Phonetic Science*, pod redakcją W. Hardcastle i J. Laver. Blackwell Publishers, Oxford.

Nowak I., 1991. *Automatyczna transkrypcja polszczyzny nieregionalnej (odmiana północno-wschodnia i południowo-zachodnia)*, tom 21. Prace IPPT PAN, Warszawa.

O'Connor J. i Arnold G., 1973. *Intonation of Colloquial English*. Longman Group Ltd., London.

of Linguistics T.O.S.U.D., 2007. ToBI. [2009-10-11].
URL {http://www.ling.ohio-state.edu/~tobi/}

Oliver D., 2008. Modelling Polish intonation for speech synthesis.

Oliver D. i Clark R.A.J., 2005. Modelling pitch accent types for Polish speech synthesis. W INTERSPEECH'05.

O'Mard L.P., Sumner C., Holmes S., Meddis R. i Patterson R.D., 2007. Development system for auditory modelling (DSAM). [2007-09-12].
URL {http://www.pdn.cam.ac.uk/groups/dsam/index.html}

OMG 2010, 2010. Unified Modelling Language. Superstructure specification 2.3.

Oppenheim A.V., Schafer R.W. i Stockham T.G., 1968. Nonlinear filtering of multiplied and convolved signals. W *Proceedings of the IEEE*, str. 1264–1291.

Osowski S., red., 1996. *Sieci neuronowe w ujęciu algorytmicznym*. Wydawnictwa Naukowo-Techniczne.

Ostendorf M., Price P. i Shattuck-Hufnagel S., 1996. Boston University Radio Speech Corpus.

Ostendorf M. i Ross K., 1997. *A Multi-level Model for Recognition of Intonation Labels*. W Sagisaka i inni (1997).

Ostendorf M., Shafran I., Shattuck-Hufnagel S., Carmichael L. i Byrne W., 2001. A prosodically labeled database of spontaneous speech. W *ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*. Red Banks, New York, USA.

Ottaviani L. i Rocchesso D., 2001. Separation of speech signal from complex auditory scenes. W *Proceedings of the COST G-6 Conference on Digital Audio Effects*. Limerick, Ireland.

Pages B., 2010. Bouml. [2010-06-10].
URL {http://bouml.free.fr/download.html}

Paliwal K. i Rao P., 1983. A synthesis-based method for pitch extraction. *Speech Communication*, tom 2.

Paliwal K.K. i Alsteris L., 2003. Usefulness of phase spectrum in human speech perception. W EUROSPEECH'03.

Palmer H.E., 1922. *English Intonation with Systematic Exercises.* W. Heffer and Sons Ltd., Cambridge.

Papazachariou D., 1994. Semantic-intonation units on one word yes/no questions. W *Themes in Greek Linguistics*, pod redakcją I. Philippaki-Warburton, K. Nicolaidis i M. Sifianou. John Benjamins, Amsterdam.

Pellom B., 2005. *SONIC: Digit Recognition Tutorial.* University of Colorado, Boulder.

Pellom B. i Hacioglu K., 2005. *SONIC: The University of Colorado Continuous Speech Recognizer. Technical Report TR-CSLR-2001-01, Revised.* University of Colorado, Boulder.

Petrillo M., 2003. *APA: an object oriented system for automatic prosodic analysis.* Rozprawa doktorska, Universita'degli Studi di Napoli Federico II.

Pickett J., 1999. *The Acoustics of Speech Communication.* Allyn and Bacon.

Pierrehumbert J., 1980. *The phonology and phonetics of English intonation.* Rozprawa doktorska, MIT.

Pike K.L., 1943. Phonetics: a critical analysis of phonetic theory and a technic for the practical description of sounds. *Language and Literature*, tom 21:182.

Pitrelli J., Beckman M. i Hirschberg J., 1994. Evaluation of prosodic transcription labeling reliability in the ToBI framework. W ICSLP'94.

Plante F., Meyer G.F. i Ainsworth W.A., 1995. A pitch extraction reference database. W EUROSPEECH'95.

POSIX, 2004. The Open Group base specifications, Issue 6. basic regular expressions. [2010-05-01].
URL {http://www.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap09.html}

PROSODY'00, 2000. *ISCA Workshop: Prosody 2000 Speech Recognition and Synthesis.* AGH, Krakow.

PROSODY'02, 2002. *Proceedings of Speech Prosody 2002.* Laboratoire Parole et Langage, Aix en Provence.

PROSODY'04, 2004. *International Conference on Speech Prosody 2004.* ISCA Special Interest Group on Speech Prosody, Nara.

PROSODY'06, 2006. *Speech Prosody 3rd International Conference.* TUDpress, Dresden.

PROSODY'08, 2008. *Proceedings of Speech Prosody 2008.* Campinas, Brazil.

R Development Core Team, 2010. *R: A Language end Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.
URL http://www.R-project.org

Rabiner L., 1977. On the use of autocorrelation analysis for pitch detection. *IEEE Transactions on Acoustics, Speech and Signal Processing*, tom 25:24–33.

Rapp S., 1998a. Automatic labelling of German prosody. W Mannell i Robert-Ribes (1998). [2008-01-05].
URL {http://andosl.anu.edu.au/icslp98/main.html}

Rapp S., 1998b. *Automatisierte Erstellung von Korpora für die Prosodieforschung*. Rozprawa doktorska, IMS, University of Stuttgart.

Ren Y., Kim S.S., Hasegawa-Johnson M. i Cole J., 2004. Speaker-independent automatic detection of pitch accent. W PROSODY'04.

Roach P., 1994. Conversion between prosodic transcription systems: Standard British and ToBI. *Speech Communication*, tom 15:91–99.

Roach P., Knowles G., Varadi T. i Arnfield S., 1993. MARSEC: A Machine-Readable Spoken English Corpus. *JIPA*, tom 23:47–54.

Rojc M., Agüero P.D., Bonafonte A. i Kacic Z., 2005. Training the Tilt intonation model using the JEMA methodology. W INTERSPEECH'05.

Rolland G., 2000. Automatic stylisation of the fundamental frequency F0 using MOMEL. [2010-01-19].
URL {http://www.icp.inpg.fr/~loeven/Praat/momel_english.html}

Rosenberg A., 2009. *Automatic Detection and Classification of Prosodic Events*. Rozprawa doktorska, Columbia University.

Rosenberg A., 2010. Autobi homepage. [2010-10-03].
URL {http://eniac.cs.qc.cuny.edu/andrew/autobi}

Rosenberg A. i Hirschberg J., 2009. Detecting pitch accents at the word, syllable and vowel level. W *Proceedings of HLT-NAACL*. Boulder, Colorado.

Sagisaka Y., Campbell N. i Higuchi N., red., 1997. *Computing Prosody. Computational Models for Processing Spontaneous Speech*. Springer-Verlag, New York.

Scheffers M., 1988. Automatic styilzation of F0-contours. W *Proceedings of the 7th FASE Symposium*, pod redakcją W.A. Ainsworth i J.N. Holmes. Edinburgh.

Schroeder M.R., 1968. Period histogram and product spectrum: New methods for fundamental frequency measurement. *Journal of the Acoustical Society of America*, tom 43:829–834.

Schroeder M.R., 2004. *Computer Speech. Recognition, compression, synthesis*. Springer-Verlag, Berlin Heidelberg.

Schwarz G., 1978. Estimating the dimension of a model. *The Annals of Statistics*, tom 6(2):461–464.

Schweitzer A. i Möbius B., 2009. Experiments on automatic prosodic labeling. W INTERSPEECH'09.

Secrest B. i Doddington G., 1982. Postprocessing techniques for voice pitch trackers. W ICASSP'82.

Seneff S., Hurley E., Lau R., Pao C., Schmid P. i Zue V., 1998. Galaxy-ii: A reference architecture for conversational system development. W Mannell i Robert-Ribes (1998). [2008-01-05].
URL {http://andosl.anu.edu.au/icslp98/main.html}

Shamma S. i Klein D., 2000. The case of the missing pitch templates: how harmonic templates emerge in the early auditory system. *Journal of the Acoustical Society of America*, tom 107:2631–2644.

Shannon C., 1948. A mathematical theory of communication. *The Bell System Technical Journal*, tom 27:379–423.

Shimamura T. i Kobayashi H., 2001. Weighted autocorrelation for pitch extraction of noisy speech. *IEEE Transactions on Speech and Audio Processing*, tom 9:727–730.

Shriberg E. i Stolcke A., 2004. Prosody modeling for automatic speech recognition and understanding. W *Mathematical Foundations of Speech and Language Processing*, str. 489–508. Springer-Verlag.

Silverman K., Beckman M., Pitrelli J., Ostendorf M., Wightman C., Price P., Pierrehumbert J. i Hirschberg J., 1992. ToBI: A standard for labeling English prosody. W *Proceedings of International Conference of Spoken Language Processing*. Banff, Alberta.

Sjölander K. i Beskow J., 2000. Wavesurfer – an Open Source speech tool. W ICSLP'00.

Slaney M., 1988. *Lyon's Cochlear Model. Apple Technical Report 13*. Apple Computer, Inc.

Slaney M. i Lyon R., 1993. On the importance of a time: A temporal representation of sound. W *Visual Representation of Speech Signal*, pod redakcją M. Cooke, S. Beet i M. Crawford. Wiley, New York.

Sobol E., 2002. *Słownik wyrazów obcych*. Wydawnictwo Naukowe PWN, Warszawa.

Sobolev V. i Baronin S., 1968. Investigation of the shift method for pitch determination. *Elektrosvyaz*, tom 12:30–36. [po rosyjsku].

Spaai G., Derksen E., Hermes D. i Kaufholz P., 1996. Teaching intonation to young deaf children with the intonation meter. *Folia Phoniatrica et Logopaedica*, tom 48:22–34.

Sreenivas T., 1982. *Pitch estimation of aperiodic and noisy speech signals*. Rozprawa doktorska, Department of Electrical Engineering, Indian Institute of Technology.

Sridhar V., Bangalore S. i Narayanan S., 2008. Exploiting acoustic and syntactic features for automatic prosody labeling in a maximum entropy framework. *IEEE Transactions on Audio, Speech and Language Processing*, str. 797–811.

Steele J., 1775. *An essay towards establishing the melody and measure of speech to be expressed and perpetuated by peculiar symbols*. The Scolar Press Limited, Menston.

Steffen-Batogowa M., 1975. *Automatyzacja transkrypcji fonematycznej tekstów polskich*. PWN, Warszawa.

Steffen-Batogowa M., 1996. *Struktura przebiegu melodii polskiego języka ogólnego*. Wydawnictwo Sorus, Poznań.

Steffen-Batóg M. i Nowakowski P., 1992. An algorithm for phonetic transcription of orthographic texts in polish. *Studia Phonetica Posnaniensia*, tom 3:135–184.

Stevens K.N., 1998. *Acoustic Phonetics*. MIT Press, Cambridge, MA.

Strom V., Elsner A., Hess W., Kasper W., Klein A., Krieger H., Spilker J., Weber H. i Görz G., 1997. On the use of prosody in a speech-to-speech translator. W EUROSPEECH'97.

Sudhoff S., Lenertova D., Meyer R., Pappert S., Augurzky P., Mleinek I., Richter N. i Schliesser J., red., 2006. *Methods in Empirical Prosody Research*. Walter de Gruyter, Berlin.

suk Kim S., Hasegawa-johnson M. i Chen K., 2003. Automatic recognition of pitch movements using time-delay recursive neural network.

Sun X., 2002a. Pitch accent prediction using ensemble machine learning. W *Proceedings of Interspeech 2002*. Denver, Colorado.

Sun X., 2002b. Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio. W ICASSP'02.

Sutton C. i McCallum A., 2006. *An Introduction to Conditional Random Fields for Relational Learning*. MIT Press.

Syrdal A.K. i McGory J., 2000. Inter-transcriber reliability of ToBI prosodic labeling. W ICSLP'00.

Szabatin J., 2000. *Podstawy teorii sygnałów*. Wydawnictwa Komunikacji i Łączności, Warszawa.

Szczyszek M. i Wypych M., 2007. Reguły akcentuacji leksykalnej dla języka polskiego.

't Hart J., 1976. Psychoacoustic background of pitch countour stylisation. *IPO Annual Progress Report*, tom 11:11–19.

't Hart J., 1979. Explorations in automatic stylization of fo curves. *IPO Annual Progress Report*, tom 14:61–65.

't Hart J. i Collier R., 1975. Integrating different levels of intonation analysis. *Journal of Phonetics*, tom 3:235–255.

't Hart J., Collier R. i Cohen A., 1990. *A Perceptual Study of Intonation: An Experimental-Phonetic Approach to Speech Melody*. Cambridge University Press, Cambridge.

Tadeusiewicz R., 2000. *Wstęp do sieci neuronowych*, rozdz. 1. W Duch i inni (2000).

Tadeusiewicz R. i Lula P., 2000. *Nuronowe metody analizy szeregów czasowych i możliwości ich zastosowań w zagadnieniach biomedycznych*, rozdz. 16. W Duch i inni (2000).

Talkin D., 1995. A Robust Algorithm for Pitch Tracking (RAPT). W *Speech coding and Synthesis*, pod redakcją W. Klejin i K.K.Paliwal. Elseiver Science.

Tamburini F., 2003. Automatic prosodic prominence detection in speech using acoustic features: an unsupervised system. W EUROSPEECH'03.

Taylor P., 1995a. The Rise/Fall/Connection model of intonation.

Taylor P., 1995b. Using Neural Networks to locate pitch accents. W EUROSPEECH'95.

Taylor P., 2000. Analysis and synthesis of intonation using the Tilt model. *Journal of the Acoustical Society of America*, tom 107(3):1697–1714.

Taylor P., 2009. *Text-to-Speech Synthesis*. Camebridge University Press, Camebridge.

Taylor P., Black A. i Caley R., 1998a. The architecture of the Festival speech synthesis system. W WSS'98, str. 147–151.

Taylor P., King S., Isard S. i Wright H., 1998b. Intonation and dialog context as constraints for speech recognition. *Language and Speech*, tom 41:489–508.

Tench P., 1996. *The Intonation Systems of English*. Cassell, London.

Terhardt E., Stroll G. i Seewann M., 1982. Algorithm for extraction of pitch and pitch salience from complex tonal signals. *Journal of the Acoustical Society of America*, tom 71:679–688.

Unicode, 2008. The Unicode Standard, Version 5.1. [2008-09-29].
URL {http://www.unicode.org/standard/standard.html}

van Noord G., 2009. FSA utilities (version 276). [2009-05-01].
URL {http://odur.let.rug.nl/~vannoord/Fsa/fsa.html}

van Santen J., 2002. Quantitative models of pitch alignment. W PROSODY'02.

van Santen J., Shih C. i Möbius B., 1998. Intonation. W *Multilingual Text-to-Speech Synthesis: The Bell Labs Approach*, pod redakcją E. Sproat, str. 141–189. Kluwer, Dordrecht.

Vapnik V., 1995. *The Nature of Statistical Learning Theory*. Springer Verlag, New York.

Venditti J.J., 1997. Japanese ToBI labelling guidelines. *Ohio State University Working Papers in Linguistics*, tom 50:127–162.

Veronis J. i Campione E., 1998. Towards a reversible symbolic coding of intonation. W Mannell i Robert-Ribes (1998). [2007-03-12].
URL {http://citeseer.ist.psu.edu/veronis98towards.html}

Vishwanathan S., Schraudolph N., Schmidt M. i Murphy K., 2006. Accelerated training of conditional random fields with stochastic gradient methods. W *The Twenty-Third International Conference on Machine Learning*. Pittsburgh.

von Heusinger K., 1999. *Intonation and Information Structure*. University of Konstanz.

Wagner A., 2008. *A comprehensive model of intonation for application in speech synthesis*. Rozprawa doktorska, University of Adam Mickiewicz, Institute of Linguistics.

Wagner A., 2009. Analysis and recognition of accentual patterns. W INTERSPEECH'09.

Waibel A., Hanazawa T., Hinton G., Shikano K. i Lang K.J., 1989. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, tom 37(3).

Walach H.M., 2004. Conditional random fields: An introduction. technical report ms-cis-04-21.

Wang Y.R., Wong I.J. i Tsao T.C., 2002. A statistical pitch detection algorithm. W ICASSP'02.

Wells J., 1997. Sampa computer readable phonetic alphabet. W *Handbook of Standards and Resources for Spoken Language Systems*, pod redakcją D. Gibbon, R. Moore i R. Winski. Mouton de Gruyter, Berlin and New York.

Wells J., Barry W., Grice M., Fourcin A. i Gibbon D., 1992. Document no. SAM-UCL-037: Standard computer-compatible transcription. ESPRIT project 2589.

Wells J.C., 2006. *English Intonation*. Cambridge University Press, New York.

Wightman C., 2002. ToBI or not ToBI? W PROSODY'02.

Wightman C.W. i Ostendorf M., 1992. Automatic recognition of intonational features. W *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*.

Willems N., Collier R. i 't Hart J., 1988. A synthesis scheme for British English intonation. *Journal of the Acoustical Society of America*, tom 84:1250–1261.

Witten I.H. i Bell T.C., 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, tom 37(4):1085–1094.

Woodland P.C. i Young S.J., 1993. The HTK tied-state continuous speech recognizer. W EUROSPEECH'93.

Wright H. i Taylor P., 1997. *Modelling Intonational Structure Using Hidden Markov Models*. W Botinis i inni (1997).

WSS'98, 1998. *3rd ESCA Workshop on Speech Synthesis*. Jenolan Caves, Australia.

Wypych M., 1999. *Implementacja algorytmu transkrypcji fonematycznej*. Tom 3 z serii Jassem i inni (1999).

Wypych M., 2001. *Synteza mowy z tekstu. Środowisko SLOPE*. Praca magisterska, Uniwersytet im. Adama Mickiewicza, Poznań.

Wypych M., 2005. An automatic intonation recognizer for the Polish language based on machine learning and expert knowledge. W INTERSPEECH'05.

Wypych M., 2006. Automatic pitch stylization enhanced by top-down processing. W PROSODY'06.

Wypych M., Demenko G. i Baranowska E., 2003. A grapheme-to-phoneme transcription algorithm based on the SAMPA alphabet extension for the Polish language. W ICPhS'03.

Yapanel U. i Hansen J., 2008. A new perceptually motivated mvdr–based acoustic front–end (pmvdr) for robust automatic speech recognition. *Speech Communication*, tom 50.

Ying G.S., Jamieson L.H. i Michell C.D., 1996. A probabilistic approach to AMDF pitch detection. W ICSLP'96.

Yoon T.J., Chavarria S., Cole J. i Hasegawa-Johnson M., 2004. Intertranscriber reliability of prosodic labeling on telephone conversation using ToBI. W *Proceedings of ICSA International Conference on Spoken Language Processing*, str. 2729–2732. Jeju, Korea.

Yoshimura T., Tokuda K., Masuko T., Kobayashi T. i Kitamura T., 1999. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. W EUROSPEECH'99.