

Homework 9

# Recognizing vowels

---

R.A.J. WACANNO (11741163)

## 9.5

```

#
# createCepstrumTable.praat
# Paul Boersma, 2019-09-30
# modified by R. Wacanno 2019-10-01
#
form Create table
    choice Dataset 3
        option no
        option class
        option course
        option open
endform

folder$ = "vowels_" + dataset$
soundFiles = Create Strings as file list: "soundFiles", folder$ + "/*.wav"
numberOfSpeakers = Get number of strings
table = Create Table with column names: "table_" + dataset$, 0,
... "speaker vowel duration c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12"
numberOfRows = 0
for speaker to numberOfSpeakers
    selectObject: soundFiles
    soundFile$ = Get string: speaker
    sound = Read from file: folder$ + "/" + soundFile$
    # Don't guess gender:
    mfcc = To MFCC: 12, 0.015, 0.005, 100.0, 100.0, 0.0
    speaker$ = soundFile$ - ".wav"
    textgridFile$ = speaker$ + ".TextGrid"
    textgrid = Read from file: folder$ + "/" + textgridFile$
    isIntervalTier = Is interval tier: 1
    if isIntervalTier
        numberOfIntervals = Get number of intervals: 1
        assert numberOfIntervals = 25
        for interval to numberOfIntervals
            selectObject: textgrid
            vowel$ = Get label of interval: 1, interval
            if vowel$ <> ""
                @fixVowel
                t1 = Get start time of interval: 1, interval
                t2 = Get end time of interval: 1, interval
                @storeAnalysesInTable
            endif
        endfor
    else
        numberOfPoints = Get number of points: 1
        assert numberOfPoints = 24
        #
        # Is the annotation left-aligned or right-aligned?
        #
        firstPoint$ = Get label of point: 1, 1
    endfor
endfor

```

```

        isLeftAligned = ( firstPoint$ <> "" )
        annotationOffset = if isLeftAligned then -1 else 0 fi
        for point to numberOfPoints
            selectObject: textgrid
            vowel$ = Get label of point: 1, point
            if vowel$ <> ""
                @fixVowel
                t1 = Get time of point: 1, if isLeftAligned then point else
                t2 = Get time of point: 1, if isLeftAligned then point + 1
                @storeAnalysesInTable
            endif
        endfor
    endif
    removeObject: textgrid, sound, mfcc
endfor
selectObject: table
Text writing preferences: "UTF-8"
Save as tab-separated file: "vowels_" + dataset$ + ".Table"
procedure storeAnalysesInTable
    selectObject: mfcc

    duration = t2 - t1

    @getFrames
    nFrames = f2 - f1

    @initArray
    @sumCoefs

    selectObject: table
    Append row
    numberOfRows += 1
    Set string value: numberOfRows, "speaker", speaker$
    Set string value: numberOfRows, "vowel", vowel$
    Set string value: numberOfRows, "duration", fixed$ (duration, 6)
    Set string value: numberOfRows, "c1", fixed$ (sum# [1] / nFrames, 3)
    Set string value: numberOfRows, "c2", fixed$ (sum# [2] / nFrames, 3)
    Set string value: numberOfRows, "c3", fixed$ (sum# [3] / nFrames, 3)
    Set string value: numberOfRows, "c4", fixed$ (sum# [4] / nFrames, 3)
    Set string value: numberOfRows, "c5", fixed$ (sum# [5] / nFrames, 3)
    Set string value: numberOfRows, "c6", fixed$ (sum# [6] / nFrames, 3)
    Set string value: numberOfRows, "c7", fixed$ (sum# [7] / nFrames, 3)
    Set string value: numberOfRows, "c8", fixed$ (sum# [8] / nFrames, 3)
    Set string value: numberOfRows, "c9", fixed$ (sum# [9] / nFrames, 3)
    Set string value: numberOfRows, "c10", fixed$ (sum# [10] / nFrames, 3)
    Set string value: numberOfRows, "c11", fixed$ (sum# [11] / nFrames, 3)
    Set string value: numberOfRows, "c12", fixed$ (sum# [12] / nFrames, 3)
endproc
procedure fixVowel
    if length (vowel$) > 1
        assert length (vowel$) = 2
    
```

```
        if mid$ (vowel$, 2, 1) = ":"
            vowel$ = vowel$ - ":" + ""
        endif
        assert mid$ (vowel$, 2, 1) = "" ; 'speaker$' 'interval'
        firstSymbol$ = left$ (vowel$, 1)
        assert index (" a e o ", " " + firstSymbol$ + " ") <> 0
    else
        assert length (vowel$) = 1
        assert index (" i y u ", " " + vowel$ + " ") <> 0 ; 'speaker$' 'vowel$'
    endif
endproc
procedure getFrames
    f1 = Get frame number from time: t1
    f1 = round(f1)
    f2 = Get frame number from time: t2
    f2 = round(f2)
endproc
procedure initArray
    sum# = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
endproc
procedure sumCoefs
    for f from f1 to f2
        for c from 1 to 12
            val = Get value in frame: f, c
            sum# [c] = sum# [c] + val
        endfor
    endfor
endproc
```