

English Consonants and Vowels

모든 consonants and vowels는 유성음과 무성음으로 구분된다.

목이 떠는 음: 유성음(voice sound): 모음 전체, b, d, g, m, n, ŋ, v, ð, z, ʒ, l, w, r, j(y)

목이 떨지 않는 음: 무성음(voiceless sound): p, t, k, f, s, h, sh, ch, θ

모음에는 단모음인 monophthongs 와 복모음인 diphthongs 가 있다.

Phonetics

Phonology: 음운론 Phonetics: 음성학 Speech: 사람의 말

Articulatory phonetics: 사람이 만들어내는 원리, 성대가 펴리며 소리를 만들어 냄.

성대 펴림: 남자 100/s, 여자 200~250/s

Acoustic phonetics: 소리와 공기의 반응, 사람이 개입되지 않음.

Auditory: 어떻게 듣는가에 대한, 다시 사람이 수반된 매커니즘; 고막, 귓바퀴로 인한 소리의 증폭

Articulation

The vocal tract(관)

Upper vocal tract

구성 부위: Palate(구개), Alveolar(잇니 뒤쪽 턱처럼 나온 부분), Velum/Soft Palate(연구개)

Uvula(목젖), Upper Lip, Upper Teeth, etc.

Lower vocal tract

구성 부위: Epiglottis(성대 뚜껑), Glottis(성대), Lower Lip, Tongue, etc.

Nasal track: Open for nasal sounds: closed by Velum; Velum goes up: nasal track is closed

Velum goes down: nasal track is open.

Nasal track이 막히면 모든 모음과 비음을 제외한 자음은 발음 가능하다.

5 speech organs = constrictors = articulators: lips, tongue tip, tongue body, velum, larynx(voicebox)

Control of constrictors(articulators): Constriction location: where? 앞뒤

Constriction degree: how much? 상하

CL: Lips: 앞부터 bilabial, labiodental Tongue tip: 앞부터 dental, alveolar, palato-alveolar, 말리는 retroflex

Tongue body: 앞부터 palatal, velar

CD: Stops: p, t, k, b, d, g, m, n, ng Fricatives: z, s, f, h, v, th Approximants: r, l, w, j(y)

Vowels: vowels *모든 자음은 stops, fricatives, approximants 중 하나이다.

*모든 모음은 constrictor로 tongue body 만 사용한다.

Praat 음성 분석 구성: formant(can be multiple, Hz), pitch(Hz), duration(sec), intensity(dB)

Praat 사용 및 분석

Duration: 클릭 드래그로 특정 부분 선택, 윗부분에서 초를 읽기, 확대하여 자세히 보기

Praat 아래쪽: spectrogram, 이를 분석: spectral analysis Spectrogram의 x축은 time, y축은 frequency

색이 진한 것이 더 크게 나타나는 음. Low frequency: high energy High frequency: low energy

Sine Wave: frequency & magnitude Amplitude = magnitude

사인 그래프 x축: 시간, y축: 값 → x축: frequency(Hz), y축: amplitude → spectrum

모든 신호는 조금씩 다른 sine wave의 합으로 표현할 수 있다.

*Hz는 1초당 진동수를 의미

Simple (simplex tone) ↔ complex tone

하나의 sine wave 다 합친 모습

사인 그래프들의 합 그래프는 가장 낮은 주파수(Hz)의 패턴과 거의 일치한다 = pitch = 1초에 성대가 떠는 수

→speech의 source는 sine wave들의 합이다.

→가장 주가 되는 sign wave는 amplitude가 높다.

→배를 이루며 나타나는데, 이를 harmonics라고 한다.

F0: lowest pure tone, Fundamental frequency, 가장 저주파로부터 산맥이 있음 → 첫번째 산맥: formant 1(F1)....

$F1 = F0 \times 2$, $F2 = F0 \times 3$, $F3 = F0 \times 4$

여성: 첫 시작이 남자보다 높고, 더 듬성듬성하게 그려진다.

남성: 첫 시작이 여자보다 낮고, 더 촘촘하게 그려진다. →배음의 숫자는 남자가 더 크다.

Source: 성대: harmonics가 gradually decrease

Vocal tract로 소리가 filter: amplitude의 패턴이 깨진다.

Praat VowelEditor

Vowel space: F1과 F2가 모음을 결정

F1: 모음의 높낮이 F2: back & front

모든 언어에는 단어와 문법이 있음.

단어: 의미와 정보를 담는다 → 정보를 담는 그릇

컴퓨터 언어의 단어: 변수 → 숫자 or 글자

컴퓨터 언어의 문법 1: 변수에 정보를 assign하는 것

컴퓨터 언어의 문법 2: if로 정보를 conditioning 한다.

컴퓨터 언어의 문법 3: for를 이용, 여러 번 반복한다.

컴퓨터 언어의 문법 4: 함수로 packaging하는 것

왼쪽: variable, 오른쪽: 정보

function: 함수(입력) → ex) print(a)

셀을 선택 → b: 셀 아래에 만들기 a: 셀 위에 만들기 x:셀 삭제

문자를 변수에 지정할 때는 ‘ ‘ 혹은 “ ”를 사용한다.

shift & enter: 실행 단축키

마지막에 변수 명을 하나만 치면 변수 값을 print 해준다.

줄을 바꾸기 싫으면 ;으로 대신할 수 있다. ,는 안됨.

변수 안에 정보를 한꺼번에 넣는 list를 작성할 때는 []를 쓴다. *()도 가능 → list가 아닌 tuple이 됨, 보안 up

type(): 변수의 속성을 보여주는 함수

list: 정보의 나열 list에서는 숫자와 문자를 동시 표기 가능, list 안에 list도 가능

int: 정수

float: 실수

str: 문자

dict: 사전 {} 사용

string과 list는 유사하다.

내부적인 정보: 대가로 사용, 대가로 안은 index

리스트의 partial information을 가져올 때는 대가료를 사용한다.

```
Ex) a = (1, 2)
      b = (3, 4)
      c = a[0] + b[0]
```

dict의 정보를 access 할 때는 앞부분을 index로 활용한다.

```
index의 카운팅: a b c d e f
                0 1 2 3 4 5
                -6 -5 -4 -3 -2 -1
```

index 범위 지정: : 를 사용한다. → 이는 str, list 모두에서 적용된다.

a:b = a에서 b 직전까지 가져온다.

a: = a에서 끝까지 가져온다.

:a = 처음부터 a전까지 가져온다.

: = 전부 가져온다.

len 함수 = length의 줄임, 정보의 개수를 알려주는 함수이다. len(a)

.upper = str를 대문자로 바꾸어 놓는다. a.upper()

*NLP(Natural Language Processing)

.find = 띄어쓰기 포함 내용이 몇 번째에서 시작하는지 알려준다. a.find('b') = a 변수에서 b가 몇번째에 등장?

.rindex= 왼쪽부터 카운트하여 가장 마지막에 나타나는 곳을 알려준다. a.rindex('b') = a에서 b가 마지막 언제?

.strip: a copy of s without leading or trailing whitespace : 여백을 없애줌 a = a.strip

.split : 긴 str를 delimiter를 활용, list로 잘라주는 함수 c = a.split('delimiter')

.join : 나눈 리스트를 다시 str로 합치는 함수 a = 'delimiter'.join(c)

.replace: 내용 하나를 다른 내용으로 바꾼다. a.replace('b', 'c') = 모든 b를 c로 바꾼다.

노트를 남기기 위해서는 앞에 #을 붙인다. 혹은 줄 하나를 markdown으로 지정하면 더 편하다.

markdown일 때 앞에 #을 붙이면 글씨를 크게 할 수 있다.

for loop

list에 있는 것들을 하나씩 일괄 프린트할 때: **for i in a:** → in 뒤에 있는 것을 i로 받아서 아래를 하라.

print(i)

in 뒤에 range를 사용할 수 있다. range는 index를 만들어 준다.

0에서 시작한다. ex) range(4): 0, 1, 2, 3

a[i]: a의 i번째가 된다. → **for i in range(4)**

len(a)를 사용하면 편하게 a 길이 지정이 가능하다. → **for i in range(len(a))**

enumerate 함수 → 번호를 추가로 매겨주는 함수이다.

for i, s in enumerate(a):

i 에는 번호, s에는 그 값이 들어간다.

어떤 특정 format으로 두 리스트를 출력하고 싶을 때, 입력값을 변수에 받아온 후,

“ “ 안에 format을 지정, 변수 부분을 {}로 지정하고, 변수 각각에 무엇이 들어갈지 .format으로 지정해준다.

"{variable 1}: {variable 2}".format(variable 1, variable 2)

zip을 사용할 때, a와 b의 length가 같아야 한다.

if 조건문

if 뒤 조건이 맞을 때 : 아래의 명령을 실행하라.

if a == b: a가 b라면. ==: 변수 지정이 아닌 진짜 =의 의미가 된다 .

if a != b: a가 b가 아니라면

if a >= b: a가 b보다 크거나 같다면.

else 조건문 if 조건문의 조건 외의 것이 나왔을 때 어떤 값을 실행할지 설정한다. → else:

range 안 숫자 2개를 넣을 시 ex) range(1, 3) → 1, 2의 값이 된다. ex2) range(3,5)→ 3, 4

string과 list는 유사하다.

$n[:] = n$

harmonics는 f_0 부터 모두

vocal tract는 harmonics의 위치는 변화시키지 않고, 그저 amplitude만 변화시킨다.

모음은 constriction location을 따로 정의하지 않는다.

amplitude 단위는 알 필요가 없다.

amplitude는 pressure → 목소리 톤에서 결정된다.

행렬

소리, 영상, 텍스트, 숫자 data 등은 숫자의 열로 표현이 되어야 함

숫자의 열: 벡터

흑백의 그림을 명도에 맞추어 숫자열로 표현: 직사각형의 형태로 숫자가 배열됨 : 행렬

숫자를 펼쳐서 놓음(한줄 한줄 길게 늘어놓음). → 벡터라이즈 한다.

흑백: 직사각형 1장 : 2차원

컬러: 직사각형 3장(RGB) : 3차원

동영상: 직사각형 3장이 시간별로 다르게 구성됨 : 4차원(시간 추가)

소리: wave form의 요소 하나하나가 값을 가짐, 그 값들을 숫자 값으로 표현 가능.

텍스트: ex) 5만개의 단어가 담긴 사전의 첫번째 단어: 10000000....000(수가 총 5만개)

두번째 단어: 01000000....0000(수가 총 5만개)

NumPy

library를 import

import library as Variablename

library: 쓸 만한 함수들을 모아 놓은 창고

한 library 안에도 library가 있을 수 있다.

예를 들어, numpy 안에 A, B, C / A안에 D함수가 있을 때, D함수는 numpy.A.D로 나타낸다.

D 안에 함수 f가 있을 때, numpy.A.D.f가 된다.

다른 방법으로는, from numpy import A.D 라고 쓸 수 있다.

numpy: 리스트에 숫자가 들어갈 때 리스트들 간의 연산을 가능하게 해준다.

import numpy

numpy.array : 계산할 수 있는 행렬/벡터로 리스트를 변화시켜주는 것 →

`variable = numpy.array(listname)`

`import numpy as np` → np는 줄여 쓰는 이름

`Variable = np.array([[list1] , [list2]])` → 2차원으로 만들어 줌

`Variable.shape` = (행, 열) 혹은 차원에 따라 요소가 늘어남.

`matplotlib.pyplot` → plotting을 할 때 사용하는 library, pyplot은 matplotlib의 하위 library 이다.

`import matplotlib.pyplot as plt` = `from matplotlib import pyplot as plt`

`np.empty([행, 열], dtype='type')` → `ex([2,3])` → 2행 3열

`empty`: numpy 안의 함수, 아무 수로나 행열을 만듦.

`dtype` = 행열을 채울 값의 속성을 지정해준다.

`np.zeros([행, 열])`

`zeros`: numpy 안의 함수, 0으로 이루어진 행열을 만듦.

`np.ones([행, 열])`: 1로 이루어진 행열을 만듦.

`float64`: 숫자가 64비트가 될 때까지 표시됨, 정확도 up, 데이터 up, 계산 시간 down

`np.arange(a)` 0~a-1까지 나열

`np.arange(a, b)` → a ~ b-1까지 나열

`np.arange(a, b, c)` → a~ b-1까지를 c간격으로. 나열

`np.linspace(a, b, c)` → a~b 까지를 c개, 같은 간격(space)으로 나눔.

`linspace` = linear space의 준말.

Variable.ndim → variable의 차원을 알려줌.

Variable.dtype → variable의 타입을 알려줌.

variable.astype(np.typeName) → variable을 typeName 타입으로 바꿔준다.

np.zeros_like(variable) → variable을 0으로 바꿔준다.

= variable * 0 으로도 가능.

Variable = np.random.normal(mean, standard deviation, data number)

np 안의 random 안의 normal

normal → normal distribution(정규분포)를 만들어준다.

plt.hist(Variable, bins = a)

plt = matplotlib.pyplot

.hist → 히스토그램을 만들어준다.

bins → 히스토그램 기동 수

y값은 무조건 정수 값이 나와야 한다.

값들을 다 합하면 100개가 나온다.

plt.show

Variable2 = Variable1.reshape(a, b, c)

shape(a, b, c)의 element의 수는 a*b*c이다.

reshape → shape을 바꾼다. 그러나 element의 개수를 바꾸어서는 안된다. ex) abc → bca

-1: 값을 알아서 채우고 싶을 때 -1을 적는다 ex) bca → -1ca

np.allclose(variable1, variable2) → 두개의 행렬이 똑같은지 확인해줌, 맞으면 true 출력

a = np.random.randint(a, b [행, 열]) → a~b-1사이의 랜덤값으로

b = np.random.random([행, 열]) → 아예 랜덤으로!

np.save("variable", a, b) → 파일 저장

del a, b → 없애기

npzfiles = np.load("variable.npz") → 로드하기

npzfiles.files[로드된 파일 중 하나] → 값 보여주기

sum을 쓰는 두 가지 방법 : variable.sum(), np.sum(variable)

variable.sum()을 쓸 수 있는 이유는 numpy안의 함수에 해당되기 때문.

1초간 무한대로 답을 수 없기 때문에 얼마나 뻑뻑하게 값을 답을 수 있는지

다음 시간 내용: sound