

09.17.(화)

(basic phonology-기본 음운론)

*English consonant

철자와 소리는 다르다.

ex) gap에서 g와 '그-'는 다르다.

*자음의 구분 - 유성음(voiced)과 무성음(voiceless)

*English vowels

monophthong: 단모음 diphthong: 이중모음

Phonetics(음성학)

phonology: 추상적인 개념, 인지적인 절차- a study on sound system

phonetics: 물리적인 것, 매번 달라지는 말소리(speech) 그 자체- a study on speech

*Articulatory phonetics (from mouth)

말소리가 만들어지는 요소(바람, 성문의 개폐 및 진동)를 바탕으로 조음과정에 대한 연구
the most primitive

cf) 턱의 높낮이는 소리를 좌우하는 주요 요소가 아님

*Acoustic phonetics (through air)

소리가 사람의 입을 떠나 공기를 타고 갈 때를 의미. 즉, 사람의 힘과는 별개의 물리적 메커니즘. 그것에 대한 음성학.

how to transmit speech

*Auditory phonetics(to ear)

귀가 어떻게 소리를 받아들이는지에 관한 음성학.(고막, 청각세포...)

how to hear speech

Articulation

*the vocal tract

tract: 관

ear(이) nose(비) pharynx(인) larynx(후)

*the vocal tract(upper)

hard palate(경구개) soft palate=velum(연구개) alveolar(치경음. 앞 윗니 뒤쪽. 영어에서 중요!) uvula(목젖)

*the vocal tract(lower)

epiglottis(후두덮개, 기도로 가는 길을 막는 역할) lip blade tongue

*5 speech organs

*oro-nasal process: 비음과 비음이 아닌 걸 나눔.

nasal tract(velum 뒤에, 코와 입이 연결되는 관)

velum이 올라가면 nasal tract이 막히고, velum이 내려가면 nasal tract이 뚫린다.(중요!)

코로 숨을 쉬면 nasal tract이 열리고 velum은 lower!(중요!, 시험문제)

*phonation process in larynx: 유성음과 무성음을 나눔

9/19(목)

*phonation process: larynx=voicebox(후두)의 개폐

voiced: can feel vibration voiceless: can't feel vibration

*articulatory process - **constrictors: in lips/ tongue tip/ tongue body**

*control of constrictors(articulators)- constrictors의 미세조정

constriction location(CL): 앞뒤

-> bilabial, labiodental,(lips), palatal, velar(tongue body).....

th-: 혀가 윗니 hit

d-, t-: 혀가 alveolar 등등

constriction degree(CD): 상하

->ex) d-: tongue tip, upper part stops

->ex) s, z, f: fricatives(소리가 끊기지 않고 지속될 수 있다)

->ex) r, l, w, y: approximants

모음은 따로 구분되므로, 모든 자음은 3가지(stops, fricatives, approximants안에 포함)

결론: How to produce English consonants vowels: by specifying constrictors, CD, CL

시험문제 예시

velum raised, larynx(glottis) open, tongue tip alveolar stop: t

모든 모음은 constrictor로서 오직 tongue body만 쓴다.

모음과 같은 constrictor를 쓰는 자음의 예를 드시오.: k

k상태에서 velum이 lower되고 glottis closed: ng

*phoneme: 음소: 소리의 최소 단위: 'a sound'- a consonant or vowel

Acoustics

formant

***Vowel acoustics: 한 학기동안 배울 기본 theme**

praat실습: record vowel /a/

반복되는 구간을 선택해서 1로 나온 수치를 나눈다.

100미만: 낮은 남자 목소리

200이상: 여자 목소리

-> 목소리의 높낮이가 나옴

new->sound->create sound as pure tone-> tone frequency에 해당 수치 입력

->같은 높낮이의 소리가 만들어짐.

숙제: email로 github계정과 학적사항 보내기

배운 내용 한 페이지로 요약해서 pdf파일로 github에 올리기

9/24(화)

복습: CL, CD(lips, tongue body, tongue tip-bilabial, labiodental, alveolar, palatal, velar), (stop, fricative, approximant)

velum(비음과 비음이 아님-비음은 3개밖에 없음)

larynx(glottis의 개폐- 무성음 개, 유성음 폐)

***vowel acoustics**

Hz: the number of occurrences of a repeating event per second=frequency

Hz=주파수 ex) sine wave(초당 주기 수)

human voice source(성대 녹음): 소리가 거의 구분되지 않음. 성대에서 나오는 소리는 기본적으로 다들 비슷하고, 입모양에 따라 소리가 달라짐.

***complex tone in spectrum**

sine wave: 가장 기본적인 소리의 형태(pitch, magnitude로 결정됨)

->sound를 포함한 세상의 모든 signal(신호) sine wave의 합으로 표현될 수 있다.

ex) wifi, 주식 시장의 흐름, 소리 등등

***38p 예시**

frequency $1 < 2 < 3$ (1, 2, 3배) magnitude $1 > 2 > 3$

4: 1, 2, 3의 합 - 간단한 신호는 복잡한 신호로 합쳐질 수 있다.

4의 주기는 가장 낮은 주파수의 1과 같다.

complex tone이란 4를 말하고 sine wave는 1, 2, 3을 말한다.

x축: 시간 y축: value or voltage

*sine wave/complex tone을 amplitude, frequency 그래프로 표현할 수 있다.

오른쪽 그림 = spectrum(=amplitude, frequency graph)

ex) 노래방 기기의 equalizer는 complex tone의 spectrum이다.

sine wave → complex tone = synthesis, complex tone → sine wave: analysis

spectrogram: spectrum을 시간 축을 기준으로

실습: '아'를 녹음해서 spectrum slice(spectrum analysis)를 봐보자. 제일 작은 pure tone의 frequency = complex tone의 pitch

*human voice source(성대 녹음): 소리가 거의 구분되지 않음. 성대에서 나오는 소리는 기본적으로 다들 비슷하고, 입모양에 따라 소리가 달라짐.

human voice source로 녹음한 소리: source라고 부름.(sound from larynx)

43/44p human voice source와 filtered by vocal tract(입을 거쳐 나온 소리)의 비교.

cf) f_0 = fundamental frequency = pitch

43 source의 spectrum: 배를 이루는 sine wave들의 합.

ex) 첫 sine wave의 hz가 115라면, 두 번째 sine wave의 hz는 약 230정도 된다. amplitude가 점점 작아진다. hz는 harmonics(배를 이룬다)

frequency축 주기는 f_0 (첫 sine wave의 주기, 가장 작은 주파수)과 일치한다.

시험문제: 10000hz까지 남자가 갖는 배음의 수가 많을까 여자가 많을까? 답: 남자

filtered by vocal tract: amplitude의 패턴이 깨짐. 배음은 그대로.

spectrogram(아래의 이미지, spectrum을 90도 회전하여 3차원화한 것.)

-> 까만색: (amplitude가 큰 값): 에너지가 높다.

시험문제: spectrogram에서 x축은 시간 y축은 frequency

9/26(목)

세상의 모든 소리가 harmonics 되는 건 아니다. 목소리의 특징이다. 다른 소리 중에서 harmonics되는 대표적인 소리는 guitar plucking이다.(voice source와 굉장히 유사하다.)

*54 f_0 : 첫 sine wave의 주파수. f_1, f_2, f_3 → 각 산맥의 정상들의 주파수

모든 모음을 만드는 데에는 f_1, f_2 만 있으면 된다.

58: 각 모음을 만들려면 f_1, f_2 만 있으면 된다. f_1 은 혀의 높낮이를 결정하고 f_2 는 혀의 앞뒤를 결정한다.

실습: new → sound → vowel editor

임의의 지점을 클릭하면 모든 언어의 모음을 다 들을 수 있다.

10/01(화) - python coding

*코딩: 자동화

자동화의 이유: 똑같은 과정을 쉽게 반복할 수 있게 하기 위해. = 코딩의 이유

모든 프로그래밍 언어의 공통점: 단어, 문법. 단어를 combine하여 정보 전달

단어란: 정보(의미)를 담는 그릇 → 컴퓨터의 변수 (variable)

이 정보를 바탕으로 기계와 소통

- 문법: 1. 변수에 어떤 정보를 assign 하는지. 숫자, 문자(variable assignment)
2. conditioning --> "if / then"
3. 여러 번 반복 --> "for / loop"
4. 함수: 여러 명령을 package화해서 간단하게, 입력하면 출력되게 만든 것.
ex) 두 숫자를 주면 그 사이의 모든 자연수를 더한 값을 출력하는 함수.
반복적으로 사용.

디렉토리: 영역

*jupyter notebook: 웹 페이지에서 프로그래밍. 자동 저장

오른쪽의 정보를 왼쪽의 변수에 넣는다.

*jupyter notebook 실습

제목칸 select: a(above) b(below) x(delete) 셀의 추가 및 삭제

a = 1

print(a) -> 어떤 변수를 입력하면 그 값을 출력해줌

a = 2를 입력하면 2가 a에 overwrite된다.

실행: shift enter 문자의 입력에는 "필수

; -> 줄 바꿈(두 줄에 적을 걸 한 줄에 적을 수 있다)

a = [1, 2, 3, 5] ->list

type(a): 출력: list -> type 함수는 변수의 형태를 출력(int, float, str, list, ...)

list 안에는 숫자도 문자도 다른 list도 들어갈 수 있다.

[1, 2, 3, 'love', [1, 'bye']]

() : tuple -> list와 사실상 동일함. 보안에 조금 더 강하긴 함.

dictionary(dict): a = {'a' : 'apple', 'b' : 'banana'} 형식: 중괄호, : (문자 말고 다른 것도 가능)

*10/8 variables

a = [1,2] b = [3,4] c = a[0] + b[0] c = 4(a의 첫 항과 둘째 항의 합)

c = a[1] c = 2(둘째 항)

어떤 변수 a의 내부정보를 가져오기 위해서는 대괄호를 쓴다.

a = '123'; print(type(a)); print(a[1])

2(문자 2)

a = 123 ; print(type(a)); print(a[1])

error(숫자는 하나의 데이터기 때문에 두 번째 항이 없다.)

a = '123' ; a = list(a)

['1', '2', '3']

dict의 경우 :로 묶인 pair에서 앞부분을 index data로 가져온다.

a = {"a": "apple", "b" : "orange", "c" : 2014}

```
print(a["a"])
```

```
apple
```

*정보접근에 있어서 string과 list의 유사성

```
s='abcdef'
```

```
print(s[0],s[5],s[-1],s[-6])
```

```
print(s[1:3]), s[1:], s[:3], s[:])
```

```
<class 'str'>
```

```
a f f a
```

```
bc bcdef abc abcdef (즉 : 앞은 +1 : 뒤는 그대로)
```

```
n = [100,200,300]
```

```
print(n[0], n[2], n[-1], n[-3])
```

```
print(n[1:2], n[1:], n[:2], n[:])
```

*len함수: len() -> 정보의 길이를 알려준다.

```
a='abcdef'
```

```
len(a)
```

```
6
```

upper() -> 대문자로 바꾼다.

형식: a.upper()

```
s[1]+s[3]+s[4:]*10
```

```
'bdefefefefefefefefef'
```

cf)자연언어처리 궁금하면 찾아봐라.(어떻게 텍스트를 처리하는지)

```
s = ' this is a house built this year.'
```

```
result = s.find('house')
```

```
result
```

```
11
```

```
result = s.find('this')
```

```
result
```

1 -> this가 두 번인데 가장 앞의 this를 찾았기 때문

```
result = s.rindex('this')
```

```
result
```

23 -> 가장 오른쪽의 index

```
s = s.strip()
```

s -> 쓸데 없는 공백 제거

tokens = s.split(' ') -> space 마다 쪼개라.(tokens는 그냥 맘대로 정한 variable이름)

```
['this', 'is', .....]
```

tokens -> 긴 소설 등의 텍스트를 관리하는데 유용.

```

s = ' '.join(tokens) -> '안의 내용을 사이에 넣어 list의 각 요소 합쳐 str화
s
'this is a house built this year'
s = ''.join(tokens)
s
'thisisahousebuiltthisyear'
s=s.replace('this', 'that')
s
'that is a house built that year.'

```

10/10

*comment 메모 다는 법

1. 데이터 앞에 #
2. 위의 스크롤 상자에서 code -> markdown으로 바꾸면 됨.

*for loop

```
a=[1,2,3,4]
```

1. for i in a -> a의 각 데이터 차례대로 하나씩 추출
2. for i in range(len(a)) -> a의 각 항 번호 차례대로 추출 이때 print(a[i])하면 1과 동일
3. for I s in enumerate(a) -> I에 번호, s에 데이터 추출
4. for I s in zip(a,b) -> I는 a 데이터, s는 b 데이터 추출

#시험문제! 이중, 삼중.... for loop

```

for i in range(1,3):                #range(1,3) -> 1에서 시작해서 3직전까지 즉, 1, 2
    for j in range(3,5):
        #과정: 3*1, 4*1, 3*2, 4*2 (즉, 하위 loop먼저 돌고, 상위의 loop가 돈다.)
        print(i*j)

```

3,4,6,8

```

for i in range(1,3):
    for j in range(3,5):
        if j >=4:
            print(i*j)

```

4,8

```

for i in range(1,3):
    if i >=3:
        for j in range(3,5):
            print(i*j)

```

값없음

데이터: 반드시 숫자화 되어야 한다.

데이터 종류: 소리, 영상, 텍스트, 숫자 데이터

벡터: 숫자의 열을 벡터라고 한다. (행렬과 달리 한 줄에 길게 쭉 나열해놓은 것.)

사진을 확대해서 보이는 하나의 픽셀은 각각 하나의 숫자다.

행렬: 숫자를 직사각형 안에 넣어놓은 것. 사진은 행렬의 형태다.

모든 데이터는 행렬으로 벡터로 전환할 수 있다. 사각형 안에 넣을지, 한 줄로 할지 선택하면 되고, 데이터는 일반적으로 벡터일 때 더 다루기 쉽다.

영상: 기본적으로 2차원. 여기에 컬러 이미지면 3차원, 여기에 시간까지 도입하면 4차원.

소리: 벡터화 할 수 있다. waveform의 소리를 한 줄로 나열하면 그것이 벡터화다.

텍스트: 50000개의 단어가 있다고 가정하면, 첫 단어를 10000...000, 두 번째 단어를 0100.....000, 마지막 단어를 0000...00001의 방식으로 벡터화할 수 있다.

*라이브러리

파이썬에서 중요한 함수들을 정리해놓은 집합. 하나의 라이브러리 아래에 하위의 라이브러리가 존재한다.(import로 불러와야한다.)

형식: import numpy(필수!!!) as(선택)

import numpy from a를 통해서 하위의 함수로 바로 접근 가능 시험!! 문법!

ex. numpy안의, a안의, d 라이브러리의 함수: numpy.a.d.함수

*numpy란

a = [1, 3, 5]

b = [2, 4, 6]

c = a+b

c [1, 3, 5, 2, 4, 6] 즉, 수학적 처리(사칙연산)가 되지 않고 단순 나열 형태로만 나옴
이때, 사칙연산이 가능하게 하는 것이 numpy임.

형식: import numpy (필수!!!)

A = numpy.array(a); B = numpy.array(b); A+B

import numpy as np (numpy를 np로 줄일 수 있음)

type(A) numpy.ndarray

X=np.array([[1,2,3],[4,5,6]])

x