

영어음성학 정리 (1주 차)

2018130848 영어영문학과 민성하

1. English consonants and vowels (Phonation process in larynx)

유성음(voiced sound) vocal cords vibration O / 모든 모음 포함 ex) v, z, l, m, a, i

무성음(Voiceless sound)- vocal cords vibration X / 일부 자음 포함 ex) f, s, k, p, h

2. Phonetics

-A study on speech (음성학) (cf. Phonology-study of sound system/음운론)

-1) Articulatory phonetics (음성기관들에서 자음/모음 어떻게 만들어지는가)

-2) Acoustic phonetics (소리가 공기 중에 전파되는 것을 연구)

-3) Auditory phonetics (타인이 소리를 어떻게 인식하는지)

3. Articulation

-발성기관 이용해 소리 형성

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

4. Oro-nasal process in velum

-nasal sound... velum(입천장 격) gets lowered ex) m, n, ng (Velum raised- Oral sound)

-when breathing(코로)? 이때도 velum gets lowered (그래야 nasal track이 열리지)

5. Articulatory process in lips/ tongue tip/ tongue body

-ex) lips – p sound / tongue tip – t sound / tongue body – k sound

6. Control of Constrictors (Articulators)

-Construction location (CL): where (앞 뒤)

-Construction degree (CD): How much (상 하) [상-stops-fricatives-approximants-vowels-하]

7. Acoustics in Praat

-맨 아래쪽 띠 (점이 모여 띠처럼 보이는 부분) – Formant(Hz)1 (=F1), 올라갈수록 F2, F3...

-Pitch: 1초에 성대가 올리는 정도. 남: 약 65-200 Hz, 여: 약 145-275 Hz

-praat에서 녹음 후 'Create sound as a pure tone' 누르고 Modify에서 Multiply를 100으로- 하나의 음이 올리는데 그게 녹음된 목소리의 Pitch를 듣기 좋은 소리로 보여준다고 할 수 있다

1. Vowel acoustics

-Complex tone: 단순한 sine wave들의 합

-sine wave는 1. 얼마나 빠른가 2. 진폭이 얼마나 큰가 이두 개가 중요

-너의 voice 를 그냥 record 한 거랑 pure tone 으로 record 한거랑 Sound의 높이는 똑 같지만, Quality는 다름 -왜냐면 Source & Filter 이라는 차이가 있기 때문

2. sine wave- 가장 기본적인 시그널의 형태이다. 'Frequency' 란 '소리가 얼마나 큰가'에 의해 결정됨. 이 세상 모든 시그널(소리 포함)은 여러 개의 다르게 생긴 sine wave의 합으로 표현된다. PPT page 38의 1,2,3번이 sine wave인데

1번: 크기는 큰데 frequency는 적음(그래프 위아래로 크게 진동/꼭지점 간 거리넓음) (저음)

3번: 상대적으로 frequency가 많음. 1번보다 3배 빠름 / 2번: 1번보다 2배 빠름

4번째 거가 우리가 말하는 소리라고 생각하면 됨. 여러 sine wave의 합은 sine wave가 아닌 복잡한 신호로 만들어진다. 이 것의 반복주기는 1번이랑 똑같음 (1초에 백번 반복)

하나하나를 simplex tone(sine wave) 마지막 꺼를 complex tone

하나하나의 sine wave를 완전 간단히 표현하면 오른쪽의 막대 바들(Amplitude-magnitude)

왼쪽 그래프- X축 시간/Y축 그냥 value(숫자값) (이걸 오른쪽 그래프로 바꿀 수 있어야)

오른쪽 그래프- X축 Frequency/ Y축 amplitude 이런걸 spectrum이라고 부름

뚜-하는 pure tone을 spectrum으로 만들면 1번처럼 하나의 바만 나올 것

3. Pure tone = simplex tone. / Wave form- X축 시간 Y축 value / Spectrum X축 frequency Y축 amplitude / Spectrogram X축 시간 Y축 frequency (스펙트럼을 시간축으로 늘어놓은 것)

Human voice- 여러 개 sine wave 합쳐진것 - 100Hz, 100Hz X2, 100Hz X3... 등 그걸 다 합한게 source (larynx)에서 나는 소리다

4. Tone frequency 440-A음 (라)에 해당 무슨뜻? 1초에 440개의 표준 길이가 들어감.

스펙트럼은 시간개념이 없음. 어떤 frequency 가 많은지 보는 것일 뿐.

녹음하고 spectrum analysis하면 스펙트럼은 일정한 간격(등간격)으로 변화. 젤 처음 나온

가장 꼭짓점의 값은 Pitch와 일치. (Pitch-한 세트가 1초에 몇번 반복되는가, 단위: Hz)

사람의 모음은 어떤 frequency와 일치? 가장 작은(slow) simplex tone의 frequency가 우리의 pitch (성대가 1초에 얼마씩 떨리는가?)와 일치

5. 만일 같은 Pitch로 말한다고 하면 우리의 입모양에 따라 '아', '이' 소리 달라지는 것

-모음 말할 때 성대에서 떨리면서 소리가 나는데 그걸 그대로 캡처해서 나는 소리는 Source라고 부름. 말그대로 성대 (Larynx)에서 나는 소리

-그로부터 입 통해 나온 소리 Filter (입모양만 바꾸면 아,이,오 등 소리남)

6. Spectrum은 frequency로 되어있음. Ppt 43 page 보면 까만쪽이 amplitude 더 높은 거임 (low frequency로 갈수록 값이 커짐-에너지가 커짐)

7. Source- Human voice source consists of harmonics

8. Sine wave를 다 합친 것의 주기는 F0의 주기와 일치. 무한대로 합치면 부드러웠던 sine wave가 점점 피크 하나-0-0-0-0 이렇게 변함. Pulse train이라고 부름

9. Source만 가지고 Spectrum analysis 할 수 있음

43p 슬라이드의 두번째 사진보면 115-230-345... 약 2배 3배 4배 됨

Smooth하게 막대가 점점 줄어듦. 이중 음의 높낮이를 결정하는 것은 제일 첫번째 나오는 막대 F0 (Fundamental frequency=pitch)

10. Harmonix-배음 (Frequency 가 F0의 배를 이루는 것)

여자가 더 간격 듬성듬성, 같은 Frequency 에는 남자 음성이 배음의 수가 더 많음

11. F1이랑 F2만 있으면 모든모음의 특징 알수 O (F1 and F2 are enough to disambiguate vowels)

12. Source-filter theory (slide 54)

waveform: Voice Source -> by vocal tract -> Output from lips

Spectrum: Source spectrum -> Formant Frequencies -> Output spectrum
Output spectrum에서 가장 볼록한 피크가 순서대로 F1, F2 등...

13. Vowel space (slide 58)

F1: Y축 F2: X축 (...입의 위아래/앞뒤와 일치)

F1은 그 모음의 높낮이(혀의 높낮이) Height 결정/ F2는 그모음의 Front,Back 결정

1. 코딩은 자동화

자동화의 필요성: 똑같은 것이 반복되니까. 초기 에너지를 투자하면 반복하는데에 필요한 불필요한 에너지를 줄일 수 있다. 그래서 코딩도 하는 것임

2. 모든 언어의 공통점: 단어와 문법이 있음.

단어=정보를 담은 그릇(의미를 담은 것) / 문법=Combine시켜주는 것

Computer language도 마찬가지. 단어: '변수'(variable) 그리고 문법이 필요

3. 기계와의 문법

-여러번 반복하는 것 For (For route)

-변수에 정보를 assign하는 것 (variable assignment)

- '~할때는 이렇게, ~할때는 이렇게'를 이야기하는 If 문법 (If conditioning)

-함수 (어떤 입력 넣으면 출력이 나오는 것)

4. 컴퓨터에 명령을 할때 주는 단어 같은 것인 variable은 두 종류 1. 숫자 2. 글자

5. 컴퓨터 명령에서 equal sign은 절대 같다는 뜻 아님! 오른쪽에 있는 정보를 왼쪽에 있는 variable로 assign을 한다는 의미 (오른쪽이 정보 왼쪽이 variable)

6. 'print'도 누가 만든 함수임. 함수는 누군가가 꼭 만들어 놔야함.

print함수의 입력은 a, 그건 ()안에 표현됨

7. 만일 a = 1, 그 이후 다음 셀에 a = 2를치면 자동으로 그 값은 overlap 되어 현재 a에는 2가 들어가 있는 셈!!!! 항상 정보는 Overlapped!

8. 문자는 항상 ' ' 안에 써주어야함 (b = love 안됨, b = 'love' 이렇게!)

9. 실행 단축키 shift + enter

10. <List>

A=[1,2,3,4]

대 괄호 안에 넣어주면 됨

A안에 들어있는 게 그냥 숫자인지 list인지 알려주는 함수 'type'

List 안에 list 가능. 대괄호 대신 괄호 사용 가능

list[]랑 tuple()은 완전 완전 똑같은 tuple이 보안에 더 강하다

```
A=1.2
```

```
Type(a)
```

```
Float
```

```
A='love'
```

```
Type(a)
```

```
str
```

```
a=[1,2,3,5,'love']
```

```
type(a)
```

```
list
```

11. dictionary { } - 표제어와 설명 그원리

ex. a = {'a' : 'apple' , 'b' ; 'banana' } a가 표제어 apple이 설명

1008 영어음성학

Dictionary – pair을 맺어줌

a=1 b=1 assign 했음

a= [1,2]

b= [3,4]

c= a[0]+ b[0]

c =4나옴

왜냐면 a리스트에서 0번째, b리스트에서 0번째 값을 가져온 것이기 때문

대괄호해서 assess

```
a = 1; a = float(a); print(type(a))
```

a=1 이거의 type는 int 이걸 float로 바꿔줘(소수점있는 숫자로) 그리고 다시 a에 넣어줌. 현재 a는 float -> 그니까 a는 float

```
a = 1.2; a = int(a); print(type(a))
```

float로 만든걸 int로 바꿔줬으니까 현재 a type는 int

대괄호 반드시 써서 index적이면 세부적 정보를 선택적으로 가져온다

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

'' 안에있는건 문자이니까, 2라는 문자를 가져온 것. 숫자를 가져온 게 아님

String을 빼고('')빼고 a[1]가져오라하면 error나옴. 왜냐면 a에 달랑 숫자하나 123이 들어있는 거니까

```
a = '123'; a = list(a); print(type(a)); print(a); print(a[2])
```

```
<class 'list'>
```

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

3

List함수가 쪼개주는 역할 하는구나 1,2,3은 문자화된거

```
a = [1, '2', [3, '4']]; print(type(a)); print(a[0]); print(a[1]); print(a[2])
```

```
<class 'list'>
```

```
1
```

```
2
```

```
[3, '4']
```

List 속에 list . 1은 숫자, '2'는 문자로서의 '2'

튜플이랑 list 같음

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

```
print(type(a))
```

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

```
<class 'dict'>
```

```
apple
```

list와 같지만 정보가 pair로 들어감. Pair set에서 앞부분을 index로 쓴다 (0,1,2,3, 이런 숫자 안쓰다구) a["a"] -> a의 "a"번째 정보 가져와라

전부다 string ... int도 표제어 될 수 있을까? "a" 대신 그냥 1써보기. 그래도 됨

```
a=[(1,2,3), (3,8,0)]
```

```
print(type(a))
```

```
type (a[0])
```

```
<class 'list'>
```

```
Tuple
```

```
s = 'abcdef'
```

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

```
a f f a
```

-는 뒤로부터 오는거임 (끝에서 몇번째)

Range 해서 여러 개 정보 가져오고 싶을 땐 : 사용

Ex) 1:3 -> 1~3번째 직전까지 가져와라 (1:n이면 n-1까지만 가져오는 거임)

1: -> 첫번째에서 끝까지 / :3 -> 처음에서 3-1까지 (이것도 n-1까지) / : 전부다

List와 string은 정보가져올 때 똑같은 방식으로 한다.

len함수: variable안에 담긴 정보의 길이를 준다

ex) s = 'abcdef'에서 len(s) = 6

s.upper()

'ABCDEF' 대문자로 바뀜

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

In [5]:

```
'bdefefefefefefefefefefef'
```

Out[5]:

왜 s. 하는데 대문자로 바꿨을까?

s.find -> string에서 찾아라 라는 뜻

```
In [20] result = s.find('house')
```

```
result
```

```
Out[20] 11
```

11번째 라는 뜻 (0번째에서 시작)

This는 이문장에 두개있는데 s.find 하면 첫번째거만 나옴.

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

```
result
```

```
23
```


R은 오른쪽에서부터 count 그니까 this처럼 여러 개있을때 오른쪽에 있는 거부터 세주는거

```
s = s.strip() # a copy of s without leading or trailing whitespace
s
```

Out[14]:

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

잡스러운 거 다 없애주고 text만 남기는거

```
tokens = s.split(' ') # split s into a list wherever a t is found (whitespace by default)
tokens
```

Out[15]:

```
['this', 'is', 'a', 'house', 'built', 'this', 'year.']
```

단어별로 잘라줌 ex어떤 단어가 쟈 많이나왔을까? 이런 거 찾을 때처럼 단어의 list가 필요할때
' ' 이 것을 이용해서 잘라라 여기서의 이것은 스페이스
만약 ',' 옆으면 comma를 이용해서 잘라라 라는 뜻이 되는 거임

```
s = ' '.join(tokens) # combine the words of the text into a string using s as the glue
s
```

Out[16]:

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

Join은 반대로 자른 걸 붙여줌 애도 마찬가지로 ' ' 이거 안ㅇㄹ 있는 거 이용해서 붙여줌

만일 ',' 쓰면 this,is,a,house, 뭐 이런 식으로 됨

```
s = s.replace('this', 'that') # replace instances of t with u inside s
s
```

Out[17]:

```
'that is a house built that year.'
```

아까랑 다르게 첫번째 this만 말하는 게 아니라 모오든 this

1010 영어음성학

For loop쓰는 것/ if conditioning/ function이란 무엇인가? 누가 미리 입력출력을 만들어놓은 것 (ex. Print, len, type 등)

Cell에 comment달고 싶으면 #쓰고 하고 싶은 말 해도됨

또는 code-> markdown 으로하면 뭘 입력하든 실행이 안되어서 주석처럼 쓸 수 있음

For loop

어떤 language라도 똑같이 for/if을 씀.

for i in a: -> in 뒤에 있는 것을 하나씩 하나씩 돌려서 i가 그것을 하나씩 하나씩 받아서 뭔가를 하라

range함수- range뒤에 어떤 숫자가 나오면 list를 만든다고 생각하면 됨

ex) range(4)-> 0,1,2,3 (4개의 인덱스를 만들라는 뜻)

i는 쟈처음 0받고 a의 i번째, 즉 0번째 것을 실행해라

```
a=[1,2,3,45,6,7]
```

```
for i in range(len(a)):
```

```
    print (a[i])
```

1

2

3

4

len(a) 는 7.

```
a=['red','green','blue','purple']
```

```
b=[0.2,0.3,0.1,0.4]
```

for i, s in enumerate(a):

enumerate는 번호를 추가로 매겨준다는 뜻

a에다가 번호도 추가로 매겨주는 거 (pair로)

i에는 번호가 들어가고 s에는 a가(자기자신이) 들어간다.

```
a=['red','green','blue','purple']
```

```
b=[0.2,0.3,0.1,0.4]
```

for i, s in enumerate(a):

```
    print("{}:{}".format(s,b[i]*100))
```

For loop는 4번 돌립니다.

첫 루프가 돌 때 i는 0 s는 red이 들어옴

두번째 루프 돌 때 i는 1 s는 green

s, b[i]를 뿌려주는 중임 이걸 각각 중괄호에 넣어주는 것

zip이랑 enumerate랑 비슷 어떤 차이있는지 알아보기

IF conditioning

Equal sign을 assignment로 씀. Equal sign을 두개를 써야 진짜 equal sign가 됨

'A가 0이 아니면'은 어떻게 해야하나?

If a != 0: 이렇게 느낌표써야됨

If a >= 0: 이런 거도 됨 a가 크거나 같으면, 머이런 느낌

항상 부등호를 먼저 써야함

range함수에 range(1,3)처럼 숫자를 두개를 넣으면 print(i)했을 때 3직전의 숫자까지인

1

2

가 나온다.

1,2 3,4

```
for i in range(1,3):
```

```
    for j in range(3,5):
```

```
        print(i*j)
```

제일 큰 roop는 두번 돌고 그 밑에 두번째 줄 루프도 두번 돈다

총네 번 도는 거임

i가 1일 때 j루프 도는데 첫번째 값은 3 -> 1곱하기 3은 3

...

총네숫자 나옴

```
for i in range(1,3):
```

```
    print(i)    -> 이 프린트는 총 2번 실행 됨
```

```
    for j in range(3,5):
```

```
        print(i*j)    -> 이 프린트는 총 4번 실행됨
```

이런거 시험에 기본문제로 나옴 ㅌㅌ

```
for i in range(1,3):
```

```
    if i>=3:
```

```
        for j in range(3,5):
```

```
            print(i*j)
```

이건 아무 결과도 안나옴