

## 2019-2 ENGLISH PHONETICS

### [English consonants & vowels]

- 자음
  - Larynx가 떨리는 음(유성음, voiced)과 아닌 음(voiceless)
  - 입을 다물고 소리가 나는지 아닌지, 아랫입술과 윗니, 두 입술, 혀와 윗니
- 모음
  - 모든 모음은 유성음(매커니즘은 larynx voicebox)에서이다.

### [Articulation]

- 음의 높낮이는 떠는 횟수로 판단 ex)아1-아2-아3  
Cf)입모양에 의해 소리의 모양이 변경되는 경우 ex)아-에-이
- 한국어는 음절이 반복, 영어는 강세를 기본으로

### [Vocal <sup>1</sup>tract(upper&lower)]

- Upper part(윗니, 윗입술, 입천장)는 움직이지 않음
- Epiglottis: 꿀꺽하는 순간 기도로 가는 문을 닫아줘서 식도로 가게끔 한다.

### [5 speech organs = constrictors = articulators / Oro-nasal process in velum]

- "음"할 때, Oral tract은 막혀있고 Nasal tract은 열려 있음.
- "아"할 때, 소리가 안 바뀌는 것은 nasal tract을 안쓴다는 소리, 막혀 있음.
- Velum이 raised되면 nasal은 닫혀 있음(모든 모음의 경우)  
<-> 코로 숨을 쉴 때 nasal은 열려 있으므로 velum은 lower됨.
- 성대에서 유성, 무성음 구분 / velum, nasal 매커니즘에서 비음과 아닌 것 구분.

### [Constrictor]

- Lips(아파), tongue tip(아타), tongue body(아카)
- Constriction Location(CL), Constriction Degree(CD)에 의해 각각 앞뒤, 상하로 more specific해지며 각각의 관점에서 다르게 보일 수 있음.
  - ➔ CL: lips, tongue tip, tongue body가 각각 2개(b,f), 4개(th,d,sh,r), 2개(y,g)로 조정된다.
  - ➔ CD: 모든 자음은 stops(폐쇄음:), fricatives(마찰음), approximants(접근음)에 속함.

### [How to produce English consonants & vowels?]

- By specifying constrictors, cd and cl / velum, larynx(총 5가지)
- 모든 vowels는 constrictors로써 tongue body만 쓴다.

### [Phonemes]

- 철자가 아니다. 소리 나는 대로 읽었을 때의. Ex) psycho -> s ai kou

### [praak]

- Formant(f1 f2)에 따라 모음을 결정
- Pitch 104.6hz: 1초에 성대가 ~번 떨린다. -> 크기가 클수록 상대적으로 고음
- 신호처리 digital signal processing: 내 목소리의 hz를 pure tone에서의 hz로 변환

---

<sup>1</sup> 관.

### [lips, tongue tips, tong body / velum / larynx]

Ex) p: lips, bl(cl), stop(cd) / raised / open

B: / closed

D: tongue tips, alv(cl), stop(cd)

Z: tongue tips, alv(cl), fric(cd)

N: stops(cd) / lower / closed (open이라면 숨쉬는 것과 비슷)

### [complex tone in spectrum]

- (중요) 사운드를 포함한 이 세상에 모든 신호들은 여러 다르게 생긴 사인 웨이브의 합으로 표현 될 수 있다.
- Frequency가 작다: 흐름이 빠르지 않다, 저음 / magnitude(=amplitude)
- 100hz는 1초에 100번 반복, 합쳐서 만들어진 complex tone은 100hz와 주기가 같기에 똑같이 1초에 100번 반복이다.
- X축은 시간, y축은 value 그래프(wave form)를 frequency, amplitude 그래프(중요, spectrum)이라고 함, 시간 개념이 없음)로 변환할 줄 알아야 함
- Analysis와 synthesis(합성)의 과정
- Spectro gram: 스펙트럼을 시간 축으로 늘어 놓은 것(X축은 시간, y축은 frequency)
- Sum된 그래프의 Pitch는 가장 낮은 주파수의 frequency(진동수)와 같다.

### [human voice source]

- 떨리면서 나는 소리를 캡처했을 때 -> source
- 배음(harmonics)들의 합으로 정해지고(항상 complex tone) Filter에 따라 음이 바뀐다.
- F0 = fundamental frequency = pitch(amplitude가 크고 점점 작아진다) = 목소리의 높이
- \*2 \*4 등등은 harmonies / 남자의 배음이 더 많다. (hz가 낮기 때문에)
- X축은 time, y축은 frequency (진한 게 energy, amplitude가 크다)
- Ampiltude는 작아지지만 frequency가 커지는 사인그래프의 배턴

### [filterd by vocal tract & peak]

- 배음은 깨지지 않고 amplitude만 깨짐
- 산맥같은 느낌으로 해석(볼록 튀어나온 부분, 물결 느낌의 그래프)을 만들어주는게 입모양(filter)의 역할
- 똑같은 소리를 내는 경우 불규칙적이지 않고 누가 하든 패턴은 똑같다, 입모양에 따라 패턴은 달라진다.
- 첫번째 산맥에 해당하는 부분이 formants0 (1,2,3 ~) f1, f2가 무엇이나 -> peak에 해당하는 frequency
- <-> pitch에서의 f0과 구분
- F1과 f2만 있으면 구분 가능 -> f1, f2를 그래프로 나타냈을 때 입의 위치와 같다 -> f1은 그 모음의 높낮이(height), f2는 앞뒤를 구분해준다.

### [코딩 = 반복되는 것을 자동화]

- 모든 language는 단어가 있고 combine을 통해 communication을 만들어낸다.
- 단어에는 정보가 들어가 있음. = 단어는 정보를 담는 그릇
- Computer languages에서 단어는 변수(variable)
- 프로그래밍에 중요한 3가지 개별적인 문법
  - ➔ 변수에다가 정보를 assign하는 것
  - ➔ Conditioning에 대한 문법(if)
  - ➔ 여러 번 반복하는 문법(for root)
- 문법에서 가장 중요한 것은 함수(입력과 출력으로 packaging하는 것)
  - Ex) 두 개의 자연수를 주면 그들을 포함한 사이의 합을 구해라
  - Ex) 자동차 관련
- 재사용이 가능하고 반복적으로 필요한 부분에 코딩을 사용

### [파이썬 variable(variable assign & data type)수업]

- = : 같다가 아니라, 오른쪽에 있는 정보를 왼쪽에 있는 variable에 넣어준다.
- 아나콘다: 유용하고 좋은 함수들을 모아 놔둠.
- 변수에 넣을 정보는 숫자와 문자 2가지
- 왼쪽 칸 Select하고 b: 아래로 새로운 셀을 만든다. / a: 위로 새로운 셀 / x: 셀 삭제 / shift+enter: 실행(run)
- 맨 마지막에 변수 하나만 있으면 print함수를 쓰지 않아도 출력시킨다.
- 한 줄에 이어 쓰고 싶으면 , 말고 ; 쓰기 / 한 번에 넣고 싶으면 [ ] 쓰기
  - Ex) 1; 2; 3 / [1, 2, 3]
- Type함수에 배열을 입력하면 list로 출력, 그 외에 숫자나 문자의 성격을 알려줌(실수, 정수, 문자), list는 반드시 숫자만 들어가야 하는 것은 아님
- List는 대괄호 대신 괄호를 쓰면 tuple이라 나오는데 이름만 다르지 성격은 똑같음. Tuple은 단지 보안에 더 강함.
- Dictionary: { }를 사용하고 안에는 ,를 이용하여 구분 / '표제어' : '설명'의 형식