

1주차

1. Phonology와 Phonetics의 구분

- Phonology(음운론) : 인지적인 단계, 발화되지 않은 소리, 비연속적
- Phonetics(음성학) : 현실적인 단계, 발화된 소리, 연속적

2. English Phonetics 내에서의 구분

- 2.1. Articulatory Phonetics : from mouth
- 2.2. Acoustics Phonetics : through air
- 2.3. Auditory Phonetics : to ear

3. Articulation

3.1. Upper Vocal Tract

종류 : lip, teeth, alveolar, hard palate, soft palate, uvula, pharynx wall, ridge
주의할 점 : 영어는 alveolar에서 조음되는 음들이 발달함.

3.2. Lower Vocal Tract

종류 : lip, blade, top, front, center, back, root, epiglottis

3.3. 5 speech organs (=constrictors)

velum, larynx, lips, tongue tip, tongue body

4. Oro-nasal Process

Velum lowered : nasal tract로 가는 길이 열림. 공기가 nasal tract로 통함. nasal sound (m,n,ŋ)

Velum raised : nasal tract로 가는 길이 막힘. nasal tract로 공기가 흐르지 않음.

5. Phonation Process

Voiced와 Voiceless의 생성과정

Larynx는 Voicebox와 같다. 성대의 떨림의 여부에 따라 Voiced와 Voiceless를 구분

Voiced : Larynx closed -> vibration

Voiceless : Larynx open -> no vibration

6. Articulatory Process

6.1. Constriction location (CL) : 혀의 전후 위치

- lips : Bilabial, Labiodental
- tongue body : dental, alveolar, Palato-Alveolar, Retroflex
- tongue tip : Palatal, Velar

6.2. Constriction degree (CD) : 혀의 고저 위치

- CD의 강도에 따라 4가지로 분류함.
- stops, fricatives, approximants, vowels. (폐쇄음, 마찰음, 접근음, 모음)
- approximants : l, r, j, w
- 비음들은 nasal tract가 열리는 것이기 때문에 영어에선 stops에 해당된다.

7. 정리

- 다음의 다섯가지 분류 기준에 따라 자음과 모음을 구분하면 된다.
- specify constrictors, CD, CL, nasal tract의 개폐 여부, 성대의 진동 여부

8. Acoustics

영어음성학 Summary

2014130031 국어국문학과 민종원

- Praat의 활용. 소리의 파형과 스펙트로그램을 통해 나타난 Duration, Pitch, Intensity를 주목하기
- F1, F2는 모음을 구별하는 기준점이 된다.

2주차

1. Vowel Acoustics

- Hz : the number of occurrences of a repeating event per second
- repeating event : vibration of vocal folds
- Repeating? This might remind you of a sine wave
→ 사인 웨이브에서 소리의 높이와 밀도 볼 수 있음.

2. Complex tone in spectrum

#####이 세상에 존재하는 모든 시그널은 사인 웨이브 형태의 여러 소리의 합으로 표현될 수 있다. #####

- 그래프의 x축은 time, y축은 value.
- magnitude(amplitude) : 소리의 강도.
- frequency : 음 높이와 관련.
- 1,2,3번 그림은 simplex tone, 4번 그림은 complex tone. 4번이 우리가 일반적으로 듣는 소리의 형태.
- 1,2,3에서 4를 만드는 것을 Synthesis, 4를 1,2,3으로 쪼개는 것을 analysis라고 함.
- complex tone의 frequency는 가장 frequency가 낮은 simplex tone의 frequency와 같다.

3. Human voice source

- 성대에서 나오는 소리를 source라고 함. filter를 어떻게 바꾸냐에 따라서 달라짐.
- 스펙트럼에서 f0 에서 frequency의 배수가 커질수록 amplitude가 점차 줄어든다. (gradually decreasing)
- 남자와 여자의 배음을 생각해보자. 여자의 pitch가 더 높기 때문에 임의의 Hz 구간에서 남자의 배음이 더 많을 것!
- 스펙트로그램에서 x축은 여전히 시간, 진한 부분은 amplitude가 큰 것.

4. Source & Filter

- filter : 입모양의 영향으로 source가 바뀜. mountains and valleys가 생긴다
- 사람의 목소리는 harmonics로 이루어져 있다 : sine wave의 배음으로 이루어져 있다
- complex tone : sum of pure tones
- Pure tone : F0, rate of vibration of the larynx
- the amplitude of pure tones gradually decreases
- Guitar plucking : a complex tone consisting of pure tone components. 사람말이랑 비슷함.
- waveform : time-value plot
- spectrum : frequency-amplitude plot
- Source Spectrum은 gradually decreasing

5. Synthesizing Source

- Hz는 100씩 증가, amplitude는 0.05씩 줄이면서 100 - 1000Hz까지!
- Combine to stereo + Convert to mono
- Convert to mono 했을 때 사운드는 100Hz와 비슷하게 들린다 (인지심리학)

6. Vowel Space

F1, F2를 기준으로 하는 그래프

F1 : Output spectrum의 1st peak. 혀의 높낮이를 결정

F2 : Output spectrum의 2nd peak. 혀의 front - back을 결정한다.

- Vowel Space를 봤을 때 한국어의 'ㅏ'와 영어의 'a'는 살짝 다름. 영어가 더 back, low
- 한국어는 구강을 좀더 좁게 이용한다.

3주차

코딩에 앞서서

- 목적 : 자동화
 - 컴퓨터 언어들의 공통점 : 단어와 문법에 영역에서
 - 컴퓨터 언어의 단어 (variable) : 정보를 담는 그릇
 - 컴퓨터 언어 문법의 핵심적인 내용
- 1) 변수에 정보를 할당하는 것 2) if문 3) for문 4) 함수에 대한 이해

Variable

- Variable은 정보를 저장한다.
- Variable에 저장될 수 있는 정보의 형태는 숫자, 문자 2개 뿐.
- 오른쪽에 있는 정보를 왼쪽에 variable에 저장하는 것.
- 함수는 원하는 대로 지정할 수 있음.

예시)

a = 1

print(a)

Jupyter notebook의 사용법

- 각 행을 클릭하고 하늘색 표시가 나올 때 단축키를 누르기
- b : below에 한 줄 추가
- a : above에 한 줄 추가
- x : 선택한 행 없애기
- 초록색 표시에서 shift + enter : run

Python의 함수

- type(*) : *의 type을 확인할 수 있다
- cf) 파이썬의 type : string, 숫자(integer, float), list, tuple, dictionary

4주차

1. Jupyter Notebook을 통해 Python 코딩 시 주의 사항

- Markdown을 체크하면 주석처럼 쓸 수 있다.
- indent 주의하기. 특히 for loop이나 if같은 조건문에서!

2. 조건문

영어음성학 Summary

2014130031 국어국문학과 민종원

(1) for loop

반복 용도

for (#A) in (#B) ####A는 변수 ####B는 범위
B는 range로도 범위를 지정할 수 있다.

예시) a=[1,2,3,4]

```
for i in a:
```

```
    print(i)
```

1

2

3

4

(2) for loop 중첩

예시)

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

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

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

3

4

6

8

(3) if 함수

조건에 맞으면 실행, 조건에 맞지 않으면 실행안됨. elif와 else로 여러 조건을 중첩시킬 수 있음.

예시)

```
a_again=0
```

```
if a_again == 0:
```

```
    print("yay!")
```

```
else:
```

```
    print("No way!")
```

3. Python 문법들

(1) enumerate : 번호를 매기는 함수. for문의 변수 중 i가 번호.

cf) <https://wikidocs.net/32#enumerate>

(2) zip : 동일한 개수로 이루어진 자료형을 묶어 주는 역할을 하는 함수이다.

(3) format : {} 안에 원하는 문자열을 넣어준다.

e.g. "{}: {}".format('energy', 100)

5주차

- numpy 사용법

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

1. numpy 기본 함수

```
X=np.array ([[0, 0, 0], [0, 0, 0]]) X.shape = (2,3)
```

영어음성학 Summary

2014130031 국어국문학과 민종원

```
np.empty([2,3], dtype='int') ## [2,3]은 2행 3열, 'int'는 정수로 넣으라는 것을 의미한다
np.zeros([2,3]) ## 2행 3열을 0으로 채우기
np.ones([2,3]) ## zeros 함수 대신 ones를 넣으면 1이 들어간 거 ### 뒤에 붙는 '.'을 보아하니
숫자의 디폴트가 float이다.
np.arange(0,10,2, dtype='float64') ## arange (시작, 끝, 간격)
np.arange(5) ## 숫자 하나 쓰면 0부터 시작, 간격 1
np.linspace(0,10,6) ## linspace (시작, 끝, 나누는 개수)
X = np.array([[1,2,3],[4,5,6]]) ## 벡터를 매트릭스로 만드는 식 ## 대괄호가 두 개인거 보고
2차원인 것을 알아야 함. ## 세 개면 3차원
A=np.array([[[1,2],[3,4],[5,6]],[[1,2],[3,4],[5,6]]]) ## 3차원 그래프
A.shape ## 모양에 대한 것.
A.dtype ## A의 타입을 말하는 것
A.astype(np.float64) ## 타입을 float64로 바꾸라는 함수
np.zeros_like(A) ## 다른 건 그대로 두고 A의 형식으로 해라.
Y = X1.reshape(-1,3,2) ## Y 는 모양을 바꾸는 함수. element들의 개수가 안맞으면 에러가 뜬다.
물리적으로 못 바꿈. 이 경우에선 24개가 맞아야 함. 숫자 모르겠으면 앞에 -1을 붙이면 된다
a = np.random.randint(0, 10, [2, 3]) ## randint 함수는 0부터 10까지 숫자에서 2 x 3 으로 임의로
픽해라
b = np.random.random([2, 3]) ## random 함수는 임의로 2 x 3로 만들어내라
np.savez("test", a, b) ## savez 함수는 실제 파일로 저장하는 함수
who ## 지금 사용 가능한 variables
del a, b ## 변수에 할당한 내용 지우는 거
```

2. 정규분포

```
data = np.random.normal(0,1, 100) ## 넘파이 속의 random 속의 normal이라는 함수. normal
distribution을 만드는 함수. (평균, 표준편차, 값의 개수)
plt.hist(data, bins=10) ## matplotlib 사용. bins가 10개라는 소리.
plt.show() ##### 이 값을 모두 합하면? 당연히 100개.
data.ndim ## 변수의 dimension을 알려주는 메서드
```

3. 벡터 계산

- 같은 형식이면 계산 가능하다.

```
a = np.arange(1, 10).reshape(3,3) ## 총 9개인데 3x3으로 재구성
b = np.arange(9, 0, -1).reshape(3,3) ## 총 9개인데 3x3으로 재구성
a==b ; a>b 해보기
a.sum(), np.sum(a) ## 형식의 차이. a.sum()은 그냥 숫자 45만 뱉어낸다.
a.sum(axis=0), np.sum(a, axis=0) ## 첫번째 차원의 관점에서 sum을 해라.
a.sum(axis=1), np.sum(a, axis=1) ## 두번째 차원의 관점
a+100
b = np.arange(6) 후에 a+b ## b가 각 열마다 더해짐.
```