

영어음성학

일어일문학과

2017131326 주형준

이 수업의 최종 목표라고도 할 수 있는 음성의 분석을 하기 위해서는 영어를 통해 분석을 해야 하기 때문에 영어의 자음과 모음을 구별할 필요가 있다. 이 때 여기서 모음과 자음의 소리는 우리가 Alphabetical 하게 아는 철자적인 것이 아닌 실제로 우리가 말할 때 나올 수 있는 소리들을 나타낸 것이다 보니 생소한 표기의 자음 소리들도 몇 있었다. 실제로도 모든 소리를 우리가 쓰는 것이 아니다 보니까 한국어에는 없는 소리까지 다 표시되어 있다. 모음의 경우 단모음(monophthongs)과 장모음(diphthongs)로 구별할 수 있다.

우리가 이러한 자음들과 모음들의 발음과 같은 다양한 소리를 낼 수 있는 것은 결국 우리의 입안에서 일어나는 혀의 위치, 높낮이, 성대의 열림 유무 등 다양한 객관적이고 과학적인 형태로 분석할 수 있다. 그것이 음성학, 즉 Phonetics인 것이다.

음성학은 사람의 말에 대한 연구와 어떻게 그 말이 묘사되는지 음성학적인 연구 등을 포함한다. 조음(Articulation)은 성도(聲道)의 영향을 받는다. 성도는 코, 귀, 인두와 성대를 포함하는데. 코로 나오는 것이 nasal tract이고 입으로 나오는 것이 oral tract이라고 할 수 있다. 그 안에도 위치별로 정말 세부적으로 다양한 기관 명이 있다. Oral tract의 경우 혀, 입술, 치아, 성대, 목젖 등 다양한 요소에 의해 영향을 받아 다양한 소리가 나오게 되는 것이다. 그 중에서도 위 아래로 구성요소를 나누어 볼 수 있다. 성도의 윗부분은 upper lip, velum, alveolar, teeth 등을 말하며 아랫부분은 tongue의 tip, front, back 등으로 나눌 수 있으며 아랫입술을 포함한다.

성대에서 발성이 시작되면 velum을 통해 oro-nasal과정으로 입 안에서 조음과정이 나타나는 것이다. 성대는 매우 짧은 시간동안 엄청난 횟수로 진동을 하고 그것이 발성을 만든다. 그 안에서는 진동을 느낄 수 있는지의 유무로 유성음과 무성음으로 나눌 수 있다. 유성음(voiced)의 경우는 진동을 느낄 수 있는 음으로 v, z, l, m, a, I 등이 있으며 무성음(voiceless)의 경우는 진동을 느낄 수 없는 음으로 f, s, k, p, h 등이 있다. 자음은 유성음일 수도 무성음일 수도 있지만 모음의 경우 모두 유성음이다.

연막(velum)에서는 oral-nasal 과정이 일어난다. 한국어로 구비(口鼻)과정이라고 할 수 있다. 연막이 낮아진 상태에서는 콧소리를 낼 수 있다. M n ng과 같은 소리이다. 입을 닫고 있어도 음은 응과 같은 소리를 낼 수 있는 것이다.

조음 과정은 위아래입술과 혀끝 그리고 혀의 몸통부분에서 일어난다. 그러한 조음을 할 수 있게 하는 것을 constrictors라고 한다. 그것들의 제어를 통해서 더 많은 소리들을 낼 수 있게 하는 것이다. CL은 Constriction location의 약자로 제

약이 어디에서 일어나는 지를 따지는 것이고, CD는 Constriction degree의 약자로 얼마나 정확한지를 따지는 것이다. CL의 경우 입술이 어디를 치는지, 혀의 몸체가 어디를 치는지, 혀 끝이 어디를 치는지에 따라 소리의 구별이 생긴다. CD의 경우는 4가지 정도 (stops, fricatives, approximants, vowels)로 구분이 가능하다.

Phoneme는 소리의 발음적인 것을 나타내는 것이다. psycho의서 p,s,y,c,h,o는 각각의 철자 하나하나를 의미하는 것이지만 psycho의 발음인 s ai k ou 각각의 발음을 phoneme라고 할 수 있다. Phoneme는 발음기관에 따라서 소리가 나뉘어진다.

2주차

음성은 조음 위치에 따라서 구별이 있다. 5가지 조음 위치가 있는데 Lips, Tongue Tip, Tongue body, Velum, Larynx가 있다. 그 중에서 Lips와 Tongue Tip의 경우 CL(Constriction Location)과 CD(Constriction Degree)의 영향을 받는다.

Sine Wave는 우리의 음성을 Praat으로 확인했을 때 그 모양이 Sine Wave여서 그러한 이름을 붙였다. Frequency는 단위 Hz(헤르츠)로 나타내고 이는 성대의 떨림의 반복의 수를 나타낸다. Wave size는 소리의 음질을 다르게 만드는 요인이다. Source는 필터가 적용되지 않은 상태의 소리로 우리의 성대에 마이크를 대었을 때 나오는 소리와 같다.

Spectrum의 경우 모든 Sine Wave는 다른 Sine Wave들의 합으로 표현 가능하다는 것에 착안하였다. 각각 다른 Frequency의 소리는 다른 Sine Wave를 갖는데, 이 소리들을 합치면 Complex tone이 된다고 할 수 있다. 이 소리들을 Spectral Slice를 했을 때 각 헤르츠에 다른 값들의 막대 그래프를 볼 수 있다.

Complex tone에서 가장 큰 주기는 100Hz의 모양과 유사하다. 가장 긴 주기를 가지는 Sine Wave를 중심으로 다른 Wave들이 중첩되는 것이다. 이는 사람이 듣기에 가장 낮은 주파수가 들린다. 이러한 Sign Wave의 X축은 시간을 나타내고 Y축은 Value 값을 가진다. 반면 Spectrum 그래프의 X축은 Frequency이고 Y축은 Amplitude를 나타낸다.

Source는 Harmonics로 구성된 인간의 음성이다. 일단 Complex Tone이라는 것은 Pure Tone의 가장 낮은 값들이 중첩된 값에 따라 배가 되는 것이라고 할 수 있다. 가장 낮은 Pure Tone은 Fundamental Frequency라고 하는데 이는 Spectrum 그래프의 F0과 같은 것이다. 또 이는 Larynx의 진동물이기도 하다. 이러한 Pure tone들의 진폭은 점차적으로 감소한다.

Filter는 입이 있는 상태로, 조음 기관을 거치고 나는 소리이다. 어떠한 특정

소리를 낼 때 어떠한 사람이 소리를 내어도 유사한 Formant를 형성한다.

코딩 = 자동화로 반복적인 행위를 자동화하게 해준다.

컴퓨터 언어에서 단어에 해당하는 부분은 변수, variable이라고 한다.

우리가 알아야 하는 것은 이 컴퓨터 언어의 문법에 해당하는 부분이다.

1. Variable assignment: 변수라고 하는 그릇에 정보를 넣는 것이다.
2. IF 조건문: IF~ 행동을 할 때 조건을 제시한다.
3. For loop: 여러 번 반복하는 행위이다.

우리는 print를 통해 이 함수를 표현한다. 숫자 (int(정수), float(실수)), 문자, list를 표현할 수 있다. List의 경우는 대괄호를 쓴다. 또 중괄호를 쓰면 dictionary라고 인식한다.

Code가 아니라 Markdown을 쓰면 실행 안되고 주석처럼 쓸 수 있음. for o in a: in 뒤에 있는 것을 하나씩 돌려서 i가 그 하나하나씩을 받아서 무언가를 하라는 뜻. in 뒤에 콜론(:)붙이는 것 잊지말기

```
In [2]: a = [1,2,3,4]
        for i in a:
            print(i)
```

```
1
2
3
4
```

```
In [4]: a = [1,2,3,4]
        for i in range(4):
            print(i)
        #range 뒤에 숫자가 나오면 list를 만들어줌. 예를 들어 4가 들어가면 0부터 3까지 리스트를 만들어줌. 4는 4개의 index를 만들어주라는 의미.
        #index는 0부터 시작하므로 0, 1, 2, 3 만들. a의 0번째 것을 print out해라, a의 1번째 ,, 2번째 ,, 3번째 것을 printout해라
```

```
0
1
2
3
```

```
In [5]: a = [1,2,3,4,5,6,7]
        for i in range(len(a)): #len은 문자의 갯수를 세주는 함수. index는 0,1,2,3,4,5,6 a(0)~a(6)까지 printout
            print(a[i])
```

```
1
2
3
4
5
6
7
```

```
In [6]: a = ['red', 'green', 'blue', 'purple']
        print(a[0])
        print(a[1])
        print(a[2])
        print(a[3])
```

```
red
green
blue
purple
```

```
In [7]: #for loop를 썼을때
a = ['red', 'green', 'blue', 'purple']
for s in a: #for loop에는 반드시 indent(간격)가 있어야됨.
    print(s)
```

```
red
green
blue
purple
```

```
In [10]: a = ['red', 'green', 'blue', 'purple']
for s in range(len(a)):
    print(a[s])
```

```
red
green
blue
purple
```

```
In [12]: a = ['red', 'green', 'blue', 'purple']
b = [0.2, 0.3, 0.1, 0.4]

for i, s in enumerate(a): #enumerate는 번호를매기다는 뜻...i에는 0 s에는 red가 들어간다고 볼 수있다. 앞에있는 게 번호, 뒤에 있는게 자기자신
    print(a[i])
```

```
red
green
blue
purple
```

```
In [14]: 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))
```

```
red: 20.0%
green: 30.0%
blue: 10.0%
purple: 40.0%
```

```
In [16]: a = ['red', 'green', 'blue', 'purple']
b = [0.2, 0.3, 0.1, 0.4]

for s, i in zip(a, b):
    print("{}: {}".format(s, i*100))
```

```
red: 20.0%
green: 30.0%
blue: 10.0%
purple: 40.0%
```

```
In [22]: a = 0
if a >= 0:
    print("yay!")
    print("let's go")
```

```
yay!
let's go
```

```
In [28]: for i in range(1,3): #두번 조건인거 시험나올듯 1부터 3까지 가는게 아니라 3직전까지간다.
    print(i)
for j in range(3,5):
    print(i*j)
```

```
1
2
6
```

tree?token=783060da657dba57f37856ff491c526975e85f61de8713a

```
In [26]: for i in range(1,3): #두번 조건인거 시험나올듯 1부터 3까지 가는게 아니라 3직전까지간다.
    for j in range(3,5):
        if j >=4:
            print(i*j)
```

```
4
8
```

```
In [32]: for i in range(1,3): #두번 조건인거 시험나올듯 1부터 3까지 가는게 아니라 3직전까지간다.
    if i >=3:
        for j in range(3,5):
            print(i*j)
```