Week 1 2018130841 손윤지

There are a majority of English sounds. They can be classified as consonants and vowels, as large. In terms of sounds, we can have a study on sounds, and this one is named as phonetics. Phonetics is a study that figures out how speech is described. To be specific, there are three types of phonetics: articulatory phonetics, acoustic phonetics, and auditory phonetics. First of all, articulatory phonetics is the study about the production of speech. This is the most primitive step of speech. The very first step of articulation is to release air from lung to vocal tract. Vocal tract is composed of larynx, pharynx, nasal tract and vocal tract. To take a closer look, vocal tract is organized by upper side and lower side. Upper vocal tract includes upper lip, upper teeth, alveolar ridge, hard palate, soft palate(velum), uvula and pharynx wall, from the front. In addition, lower vocal tract constitutes lower lip, tongue, and epiglottis, also from the front.

Now again about the speech process, there are 5 speech organs, which are constrictors, also articulators, that make the actual sound during the speech. They are lips, tongue tip, tongue body, velum, and larynx. Phonation process occurs in larynx. Every sound appears through the larynx. Also, the sound from larynx can be divided by voiced sound and voiceless sound. Voiced sound is the sound that occurs by the vibration of vocal cords, and the examples are b, m, v and so on.

Oral-nasal process happens by the position of velum. When nasal sounds are made, velum goes downward so as to make the air go through the nose. Otherwise, velum goes downward, when it is oral sounds. Articulatory process is created in lips, tongue tip and tongue body.

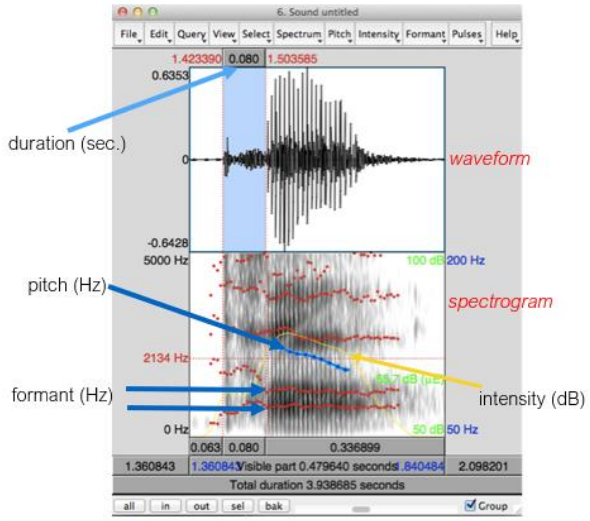
Moreover, each constrictor can be specified by constriction location and constriction degree. By the touch of lips, tongue body and tongue tip, the air flow is constricted at such location, and then it appears as different sounds. Constriction degree is about how much constrictions are made. The very much constriction made, we call it stops. The order goes as fricatives, approximants, vowels, from the most. Therefore, phonemes are produced by specifying constrictors, constriction location, and constriction degree.

Phonemes are individual sounds that form words. Also, phonemes can be defined as a combination of speech organs' actions. At lips, p, b, m, f, v, and w sounds are made. By tongue tip, ɵ, ð, ʒ, t, d, s, z, ʃ, l, and r sounds are made. By tongue body, k, g, j, ŋ and vowels are made. By velum, m, and ŋ sounds are made. By larynx, ɵ, ʃ, p, f, t, s, k and h sounds are made.

We can figure out acoustic process in Praat. We can check duration, pitch, formant and intensity in Praat. To verify voewl acoustics, we can measure pitch using Praat. By the number of occurrences of a reapeating event per second, we can find out the Hz. Then the repeating event shows the vibration of vocal folds, and the repeating parts show us a sine wave.

Week 2 2018130841 손윤지

Acoustic process in English. We can find out acoustic process by using Praat program.



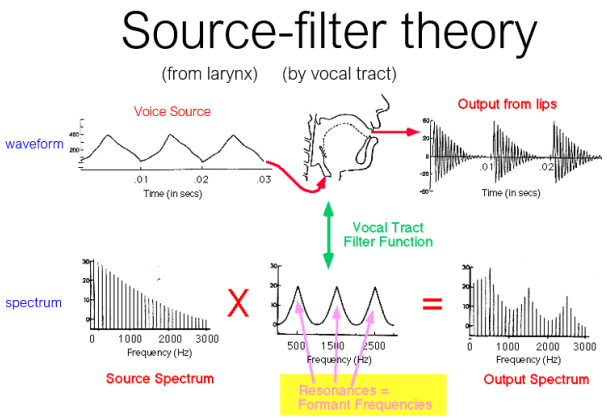
By Praat, we can figure out the waveform, spectrogram and other more information. To be specific, intensity shows the decibel, pitch shows the hertz. Formant is the darkest line of spectrogram. Also, spectrogram is the one that turns spectrum in to time line. Now, having a closer look at the vowel acoustics, all of us have different pitch, so the sine wave(pure tone) of our sound looks different.

If we listen to a sentence, we can notice that it is not a pure or simplex tone. Thus, the sound we perceive through the mouth is complex tone. In addition, how sound like differs depending on our vocal tract. Listening to the source of the sound, there are just sound and pitch. According to sign wave, x-axis describes time line and y-axis describes the value. Moreover, you can turn the sine wave into the spectrum. Spectrum shows frequency by x-axis, and amplitude by y-axis. Simple tone of 100hz is a fundamental frequency, which we will call it as F0.. By synthesizing all simple tones, harmonics are made, and we call it complex tone. In terms of complex tone, the frequency of complex tone is same as that of fundamental frequency.

Human voice source consists of harmonics. Harmonics are composed of harmonic overtone. As usual, female has less harmonic overtone than male. After that, if filtered by vocal tract, the organization of harmonic overtone is same but magnitude differs from that of simplex tone. In other words, peaks and valleys appear in the complex tone.

In conclusion, the lowest pure tone is fundamental frequency, the rate of vibration of the larynx and the number of opening-closing cycles of the larynx per second. Amplitude of pure tones gradually decreases. On the other hand, if the sound is filtered by the vocal tract, spectrogram shows formants which is the peaks that frequencies vocal tract likes.

By means of Praat, we can synthesize source ourselves. As we create one lowest pure tone and make its harmonic overtone, we can make harmonics. In spectrum, there is a mountain from the beginning and we call it F1 and F2 in order. In spectrogram it appears as formants. Each vowel has distinguishable formants and that why we can classify each vowel. Also, if we put F1 as y-axis and F2 as x-axis, we can watch vowel space.



Week 3 2018130841 손윤지

Computer program also has a language. We call it coding as usual. There are four important aspects of coding. First, there are variables which are same as the words of language. We need a function that makes system to work on, which is called 'if conditioning'. Next, for loops are needed for the repetitive act of program. Lastly, only number and characters could be the information in computer program.

By using 'anaconda prompt' we can figure out the way to test the function and programming. First of all, turn on the 'anaconda prompt', type 'jupyter notebook' on the black monitor. Then the program leads us to the jupyter notebook program. Then, now we are ready to practice to use the program.

The thing located on right is the information, and the one on the left is the variable. In the case of making a=1, 1 belongs to information and a to the variable. Then, put the function of 'print ()', which make the variable print out its information, the result would be 1. If we make new command such as 'a=2', the result of 'print' would be '2'. Thus, what we have to focus on is the number between the square brackets. After clicking 'in' at the left side, we can make new line below with tapping 'b' on the keyboard, make new line up above with tapping 'a' on the keyboard, and delete the line with tapping 'x'. Also, you can make the result by clicking 'run' box, or by tapping shift+enter.

Every character just used in the program are perceived as a variable, so to make it as a literal word of English, we need to put quotation marks such as 'love'. In addition, if you put any variable twice, especially for the last variable, when just typing variable without function, it makes same function as print function.

In the case of matching more than one information to one variable, you can put those information inside '[]'. After having type function, the result is 'list'. When there is just one information which belongs to one variable, the result of type function is 'int'. On the other hand, information of prime number and the word make the result of 'float'. When there are the word that

meets the variable, the result is the ‘dictionary’.

There could be both number and letters in the 'list'. Dictionary need more than two pairs of variable and information. Also, it is allowed only to use {}, while using dictionary. Int appears when it's natural number, and float appears when decimal. When the letter information is included, it is called as a string. Tuple is the one similar to list, but the difference is that tuple uses [] can be used when access to the information inside, such as for order. So, when it is written as 'a[2]', it means the third order of the letter. It's because the thing that comes first is regulated as '0'.

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

```
<class 'list'>
['1', '2', '3']
3
```

If it becomes as a formation of dictionary, the index becomes the variable itself, instead of using the order. Function of string is similar to that of list. In the case of order written in minus, you can think of it as a vertical line. Thus, [-1] always means the letter of the last index, and [0] always means the first. Moreover ':' shows the range of the information.

```
s = 'abcdef'
print(s[0], s[5], s[-1], s[-6]); print(s[1:3], s[1:], s[:3], s[:])
```

```
a f f a
bc bcdef abc abcdef
```

Function 'len()' shows the length of the information. You can compound the results by using plus or *. Using period after the variable and writing upper makes the information becomes upper case.

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

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

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

```
11
```

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

```
23
```

*rindex means to find the information at the right side.

s = s.strip () s
'this is a house built this year.'

*strip: wipe out all the inaccurate factors of the sentence.

Tokens = s. split (' ') tokens
'this', 'is', 'a', 'house', 'built', 'this', 'year'

*split: split the sentence by following order in the '()'

```
s = ' '.join(tokens)
s
```

'this is a house built this year.'

*join: put up the words into a sentence using the factor between '()'

s = s. replace('this', 'that') s
'that is a house built that year.'

One of the important processes of programming is 'for loop' process. The way to make for loop process is writing down 'for_ in _ :' in 뒤에 있는 것을 하나씩 돌려서 in 앞에 있는 것이 받아서 계속 동작을 반복하라는 것이다.

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

1
2
3
4

Range 함수는 range 뒤에 어떤 숫자가 나오면 몇 개의 index를 만들라는 것.

```
a = [1, 2, 3, 4] #index 를 4개를 i 에 넣어라.
 for i in range(4):
       print(a[i]) #print (i) 의 결과는 0 1 2 3임.
```

1
2
3
4

Len () 함수는 길이를 의미. Len (a) = 4

```
a = ["red", "green", "blue", "purple"]
b = [0.2, 0.3, 0.1, 0.4]
```

```
for i, s in enumerate(a): #i가 번호고(0, 1, 2, 3) s는 자기 자신.
    Print (a[i])
```

Red

Green

Blue

Purple

Enumerate: 번호를 추가로 매겨라.

```
a = ["red", "green", "blue", "purple"]
b = [0.2, 0.3, 0.1, 0.4]
for i, s in enumerate(a): #zip 은 페어로 4 개가 됨.
   print("{}: {}%".format(s, b[i]*100)) #왼쪽과 같은 형태로 적고 싶을 때.
Format 뒤에 두개가 왼쪽과 같이 적힌다는 것.
```

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

```
a = 0
if a == 0:
   print(a)#두개 써야 진짜 equal 사인
else:
   print(a+1) #if a != 0 > !가 0 이 아니면 이라는 뜻.
```

0

```
for i in range(1, 3):
     for j in range(3, 5):
          print(i*j)
```

3

4

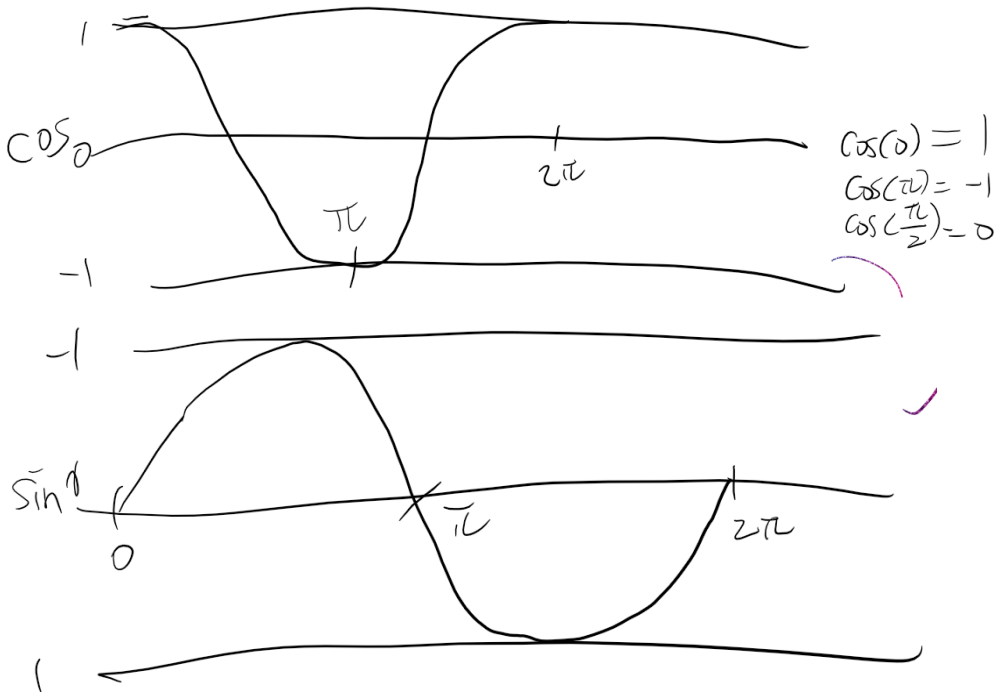
6

8

모든 데이터는 행을 늘어놓은 벡터의 상태로 이루어진다. 모든 이미지는 행렬로 이루어져 있다. 흑백 이미지가 3층이 되면 색을 나타낸다. 동영상은 그 이미지들이 시간 순으로 놓인 것이다. 이를 차원의 개념으로 생각해 볼 수 있다. 흑백 이미지는 2차원이고 색 이미지는 3차원, 동영상은 4차원이다. Wave form 과 텍스트 모두 벡터로 데이터화 할 수 있다.

Numpy는 list와 비슷한데 list와 달리 수학적인 데이터로 쓸 수 있게 하는 것이다. Import는 누가 만들어 놓은 library를 부르는 작업이다. 이를 직접 사용할 때에는 import numpy 로 사용하면 된다. 그리고 '.'은 그 안에 있는 subpackage를 빼낼 때 사용한다. 예를 들어, import numpy. A. D. function() 이라면, numpy라는 library 안에 있는 subpackage A를 사용한다는 것이다. 이는 From numpy import A. D 와 같은 것이다.

Python을 이용한 코딩으로도 소리를 분석하고 만들 수 있다. 이를 시행하기 위해서 미리 알아야하는 것이 싸인, 코싸인 함수이다. Radians 는 각도 값으로 이를 pi 값으로 표현할 수 있다. 따라서 0도는 0으로 표현이 가능하고 2*pi는 360도가 된다. 이를 그대로 sin, cos 값으로 함수를 그릴 수 있다.



함수를 그리면 위와 같은 형태로 나타난다. 또한 이 두 함수를 phasor로 볼 수 있다. Cos(θ), sin(θ)의 값으로 표현하여 숫자 값으로 나타낼 수 있기 때문이다. 데이터는 숫자로 볼 수 있으며 이를 벡터라고 할 수 있다. 따라서 오일러 공식이 필요하다. 오일러 공식은 다음과 같이 나타낼 수 있다.

$$e^{ix} = \cos x + i \sin x$$

θ값 중 주요 값들을 이 공식에 넣어서 복소평면(complex plain)으로 보면 원이 나타난다. 이 원에서 보면 project하면, cos 값이나 sin 값만 볼 수 있다. 이렇게 만드는 것은 단순한 벡터 값이고, 소리를 만들고자 하면 시간 값을 넣어줘야 한다.

Sampling rate는 1초에 몇 번으로 나누는 것이고 이는 곧 음질의 정도를 이야기한다. 만약 sampling rate가 10000이라고 하면, 1초에 10000개의 지표를 만드는 것이다. 소리에는 이에 더해 frequency도 필요하다. 그렇다면 sampling rate이 100이고 frequency가 1hz라고 해보자. 이 경우, sine wave를 한 번 그려주면 된다. 같은 sampling rate에 frequency가 100hz라고 해보자. 이 경우는 불가능하다. 표현할 숫자가 너무 적기 때문이다. 그렇다면 최대 몇 개까지 표현할 수 있을까? Sampling rate의 반까지만 가능하다. 이를 Nyquist frequency라고 한다. 즉, Nyquist frequency는 최대로 표현할 수 있는 frequency이고 이는 'sr/2'이다. 만들어 놓은 sound 값에 amplitude를 곱하면 사람의 소리와 가까워지는 소리가 나타난다.

또한, sin과 cos값은 90도, 즉 2*pi만 이동하면 똑같은 함수를 나타낸다. 그렇기에 cos과 sin 함수가 바뀐다고 해서 소리가 바뀌지는 않는다. 이는 우리의 귀가 각도(phase)의 변화를 인식하지 못한다는 것을 의미하기도 한다. 그러나 우리의 귀는 frequency의 차이는 인식한다.

pulse train 역시도 programming을 통해 만들 수 있다. 음성의 기본 원리와 같이 최소의 harmonics와 frequency를 잡는다. F0를 정하고 sine wave 를 만들고 배음을 만드는 것이다. F0을 정하고 최대가 Nyquist frequency인 sr/2이므로 Fend 값을 sr/2 로 만들어 준다. 그리고 s를 계속해서 더하고 더해나가는 것으로 pulse train을 만들어간다.

Spectrum 역시 만들 수 있는데, spectrum은 타임을 나열하는 것이 아닌 한 타임 안에 있는 주파수를 보여주는 것이다. Def 로 만들고 싶은 function을 정의할 수 있다. 또한, return은 출력을 의미하는 것이다. 이후 기본 source에 vocal tract가 지나면 사람이 발음하는 소리처럼 나타난다. 이는 RG와 BWG를 통해서 만들 수 있다.