

영어음성학이란 영어의 소리의 발현에 대한 학문이다. 영어 음성학은 크게 조음 음성학인 articulatory phonetics, 음향 음성학인 acoustic phonetics, 그리고 청각 음성학인 auditory phonetics로 분류된다. 각각은 소리가 입을 통해 어떻게 발현되는지, 그리고 공기에 의해 어떻게 전달되는지, 그리고 귀에 어떻게 들리는지에 대한 세부 학문으로 볼 수 있다.

1. 음성학에서 조음에 관여하는 기관은 총 5개이며, 이들을 constrictor이라고 부른다. 이들은 각각 lips, tongue tip, tongue body, velum, larynx이다. 이 각각의 기관은 어떻게 위치가 변화하고 운동하는지에 따라 다른 소리를 나타내며, 이를 통해 어떤 소리가 발현될 것인지 예상할 수 있다.

2. larynx에서의 발성은 성대의 진동 유무에 따라 달라진다.

성대가 진동하는 경우(voiced): v,z,l,m,a,i...

성대가 진동하지 않는 경우 (voiceless): f,s,k,p,h...

Velum은 lowered된 경우와 아닌 경우에 따라 달라진다.

3. 정확한 소리의 판별과 발성을 파악하기 위해서는 각 constrictor의 발성이 정확히 어디에서, 그리고 어느 정도로 이루어지는지 알아야 한다. 이를 파악하기 위한 지표로는 constriction level (CL), 그리고 constriction degree (CD)가 존재한다.

#### Constriction level

Lips-bilabial: 입술과 입술 (양순음)

Labiodental: 입술과 이빨 (순치음)

Tongue tip- dental: 혀끝과 이빨 (치음)

Alveolar: 혀끝과 앞니 뒤쪽 (치경음)

Retroflex: 혀끝과 경구개 뒤쪽 (반전음)

Palato-alveolar: 혀끝과 경구개 (경구개 치경음)

Tongue body-palatal: 구개음

Velar: 연구개음

#### Constriction degree

Stops/ fricatives/ approximants/ vowels

- 1) Constrictor 인 lips, tongue tip, tongue body, 그리고 velum, larynx 각각의 특징을 통해 소리를 specify 할 수 있다

Ex) /p/ lips/ CL: bilabial CD: stop

- 2) sine wave 는 가장 기본적인 형태로, frequency 와 magnitude 에 의해 형태가 결정됨

존재하는 모든 signal (sound 를 포함)은 다르게 생긴 여러 sine wave 의 결합으로 표현될 수 있다

Sine wave 에서 x 축은 시간, y 축은 value (또는 voltage)이다

- 3) 복잡한 신호는 다양한 sine wave 의 합으로 생각할 수 있음

각각의 sine wave 하나를 simplex tone 이라고 하며, 여러 simplex tone 들의 합을 complex tone 이라고 한다

이때, complex tone 의 그래프는 x 축이 frequency, y 축이 magnitude (또는 amplitude)인 그래프로 변환된다 (Spectrum)

합성 synthesis <-> 분석 analysis

- 4) Complex tone 는 단순한 sine wave 의 연속으로 이루어져 있으며, 등간격으로 이루어짐

사람의 목소리를 녹음한 경우, 가장 느린 simplex tone 의 frequency 는 우리 목소리의 pitch 의 음높이와도 동일함

또한 우리 성대가 1 초에 몇 번 진동하는지와도 일치함

- 5) Spectrum 의 x 축은 frequency, y 축은 amplitude

- 6) Wave form 의 x 축은 시간, y 축은 value

- 7) 어떤 시점에서 어떤 주파수 성분이 많은지 분석하는 것을 spectrum 분석이라고 한다

- 8) spectrogram 에서 x 축은 시간, y 축은 frequency

- 9) 사람의 목소리는 harmonics 의 배합으로 이루어짐

이 때 lowest tone F0 는 larynx 성대가 1 초에 몇 번 떨리는지와 일치함

그러나 모든 소리가 harmonics 가 되는 것은 아님 (성대의 소리, 기타소리..)

#### <문법>

1. Variable assignment 변수에 정보를 넣는 것, 정보 assign
2. If Conditioning
3. For loop 여러 번 반복 하는 것
4. 함수

재사용. 반복적으로 사용하는 것 가능.

Computer language 에서 =는 같다는 의미 X

➔ 오른쪽에 있는 정보를 왼쪽에 있는 variable 에 assign 한다는 의미

Ex) a=1: a 랑 1 이 같다는 의미 x, 1 이라는 정보를 a 라는 변수에 assign

A=above, B=below, x=delete

반드시 run(실행)을 해야 변수에 정보가 입력됨

여러 개를 나열할 경우:

마지막 변수명은 print() 하지 않아도 run 하면 나옴

a=1

b=2

c=3

c

➔ run 하면 결과 값은 3

문자는 "따옴표 사용해야 함

-따옴표 사용하지 않으면 숫자와 같은 정보 취급

한 번에 여러 함수를 실행하고 싶으면 ;를 사용하여 나열

List 대괄호

종류를 알려주는 함수 type()

Ex) int. 정수

Str. 문자

Dict 사전 함수

Tuple ()

List[]

-> 똑같은 것

float 함수는 어떤 variable이 들어오면 float type으로 바꿔줌

ex) a는 원래 int. float 함수 사용하면 float로 바뀜. 소수점 표시

dict함수

pair set에서 앞 부분을 index로 쓴다

dict의 정보를 access 할 때는 a,b,c... 을 index 의 수단으로 사용

ex) a에 해당하는 정보를 가져오라

print(a["a"])

list함수와는 달리 'a'번째로 생각하면 됨. 숫자 x

표제어를 숫자로 해도 print(a[1]) 단순히 표제어를 바꾼 것이기 때문에 똑같은 결과값 나온다

\*string과 list의 공통점

index에서 정보를 접근하는 방식이 똑같다

s='abcdef'

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

01234

-5-4-3-2-1

제일 첫 번째 것은 항상 0

제일 마지막 것은 -1

여러 개의 정보 담고 싶을 때 :

ex) 1:3 첫 번째에서 세 번째 직전까지 → 첫번째와 두번째 (미만을 생각하면 됨)

1: 처음부터 끝까지

\*len 함수 =length, 변수에 담긴 정보의 길이를 알려준다

\*.upper() =대문자로 변경

ex) s='abcdef'

s.upper()=ABCDEF

어떤 variable을 만들고 .함수를 쓰면 실행됨

\*.find()

ex) 11 : 11번째부터 찾는 단어가 시작됨

s='this is a house built this year'

this는 문장에서 2번 나오지만 find함수를 쓰면 첫 번째의 위치만 나옴

\*.strip()

순수한 text만 남겨주는 함수

(a copy of s without leading or trailing whitespace)

\*.split()

: split s into a list wherever a t is found

긴 string을 () 안에 있는 내용에 따라 잘라라

ex) s.split(' ')

\*tokens는 그냥 variable에 이름 붙인 것

\*s는 행렬로, 모두 더해서 만들어짐

pulse train의 형태로 만들어진다

이때 x축은 시간 time, y축은 에너지 energy

y축이 에너지인 wave form으로 보거나 , y축이 frequency인 spectrogram을 기준으로 볼 수 있음

wave form의 단면을 잘라서

- x축을 주파수, y축을 진폭으로 정하면 spectrum이 만들어진다

이때, f0= 100hz~f49=5000hz

-이때 formant가 어느 frequency가 되는지에 따라 f1, f2가 달라짐

(소리가 하나의 톤로 압축되면서 peak가 달라지기 때문)

-f0부터 시작해서 f49까지 점점 감소하는 모양을 만든다

-peak를 하나 만든 뒤 하나씩 만들어서 적용한다

\* 사인과 코사인의 소리의 차이를 인간이 인식할 수는 없음

\* 코사인 그래프는 사인 그래프를 이동시킨 것

→ 사인 그래프를 파이/2만큼 이동하면 코사인 그래프가 된다

→ 그러나 얼마나 이동하던 간에 소리의 차이에는 변화가 없음

(인간의 귀는 phase shift를 인식하지 못하기 때문에 이 차이를 인식할 수 없다)

\* 사인과 코사인의 소리를 구별할 수 없지만 frequency에 대해서는 소리의 차이를 인식할 수 있다

Linear algebra

\* 인공지능= 행렬의 곱

입력 벡터를 조작→ 출력 벡터 산출

축적된 데이터를 통해 학습한 기계화된 과정이다

행렬의 차원은  $m \times n$  행→ $m \times n$ 행렬

\*벡터Vector

a sequence of numbers

숫자 데이터의 연속으로 숫자 데이터는 기하로 변환 가능하다

\*Scalar Vector

Scalar도 1-by-1 matrix라고 생각하면 됨

\*Vector Space 벡터 공간

무수하게 많은 벡터들의 linear combination이 만들어내는 공간이다

$n$ 차원의 모든 공간을 포함하며, 일부분만은  $x$

→ Vector space이 조건

-linear combination이 공간 내에 존재한다

$R^n$ :  $n$ 개의 구성요소를 가진 모든 벡터가 채워져 있는 공간

$R^1$ 은 line,  $R^2$ 는 평면

\*Column Space

Column space  $C(A)$ : 메트릭스  $A$ 의 column들이 spanning해서 만드는 공간

column들이 linear combination을 통해 만들어내는 모든 공간

column vector 2개, 이 2개의 column들을 통해 linear combination을 무한대로 하면

2차원 공간을 채울 수 있다→ column space

이 때, 2개의 column이 같은 선상에 있다면→ dependent

같은 선상에  $x$ → independent

\* whole space- column vector의 관점에서 row vector의 수

row vector의 관점에서는  $m \times n$  행렬은 두 가지 whole space 가짐 :  $R^m, R^n$

\*Linear Transformation

$Ax=b$

이 때  $A$ 는  $x$ 의 차원과 숫자 모두 바꿈→ linear transformation

행렬=대문자

벡터=소문자

$x$ =입력 벡터

$b$ =출력 벡터

$A$ = transformation matrix 입력벡터를 출력 벡터로 바꿔준다

방향에 따라 출력에 영향을 미칠수도 아닐수도 있다.

이를 구분하는 것이 null space와 그렇지 않은 space의 차이를 아는 것

방향에 따라 출력에 영향을 미치느냐 vs 아니느냐

→ null space와 그렇지 않은 space의 차이

\* null space

- 입력이 변하더라도 출력에 영향을 미치지 않음

1) 수학적 해석- 하나의 행렬이 주어졌을 때, 어떤 수를 곱해도 반드시 0이 될 가능성이 있는 모든 점

2) 기하학적 해석- whole space에서 column space를 제외한 공간

3) 실용적 해석- 아기가 드럼을 배우는 경우 등

\* 충분한 null space를 확보해야 하는 이유

- 하나의 업무에 대해서 우리가 최종적으로 하려는 일은 출력.

- 가는 길에 장애물이 있으면 피해야 함

- 그러나 특별한 영향 없이 업무를 마무리하는 것, 즉 출력을 내는 것 가능

→ 인류는 진화하면서 null space를 확장해왔음

- null space를 통해 인공지능과 선형대수와의 관계를 설명할 수 있다

\* correlation 상관관계

- 상관계수  $r$

$-1 \leq r \leq 1$ ,  $r=0$ 일 때 상관이 가장 적음

$\cos \theta = r$ ,

$\cos 90^\circ = 0$ ,  $\cos 0^\circ = 1$

\* inner product

- 기하학적 의미: 90도에서 projection 시킨 이후에 곱해도 값은 0이 된다

\* 행렬의 형태가 다를 경우 transpose 시키는데 두 가지 방법 존재

1) 행렬의 크기가 작을 경우- inner product

2) 행렬의 크기가 클 경우- outer product

\* wave의 헤르츠 별로 inner product를 구하면 해당 hz의 분포량 알 수 있음

- slice된 한 타임이 계속되면 spectrogram이 형성됨