영어 음성학 정리 4

2017190246 유서현

< 1 주차 9/5 >

- Praat: 음성 분석 프로그램
    - 소리 = 숫자. x 축은 시간, y 축은 분석값. 그래프는 그 숫자 값들이 연결된 것
    - 그 값들은 끊임없이, 연속적으로 이루어져 있음. 쪼개는 기준에 따라 나누어지는 거지 실제 소리는 continuous 하다. 즉, 목적에 따라 나누어 값을 뽑기 나름이다. (ex. 1 초를 얼마간격으로 나눌 것인가? Sampling frequency 44100Hz = 점 사이당 44100/1 초)
    - cut, pitch(높낮이), intensity(세기. 폭이 넓으면 intensity 가 강하고 좁으면 약하다), 분석본에서 위는 wave, 아래는 소리의 스펙트럼(빛을 프리즘으로 본 스펙트럼처럼)

(+ 프리즘에서 빨간색은 더 온도가 낮고 보라색은 더 높음. 빨간색이 더 slow 하고 저주파, 보라색이 더 빠르고 고주파. 즉, 위로 갈수록 고주파, 아래쪽으로 갈수록 저주파)

< 3 주차 9/17, 9/19 >

- Phonology(음운론)_cognitive, abstract
    - English consonants
    - English vowels: 단모음(monophthongs), 장모음(diphthongs)
- Phonetics(음성학)_physical
    - Study on sound system
        Articulation: 소리를 만들어내는 사람의 원리
        Acoustic: 공기에서 소리가 어떻게 나는가. 물리적인 것.
        Auditory: 어떻게 듣는가. (Ex. 귓바퀴는 소리를 증폭시켜 듣게 해줌)
    - The vocal tract: 크게 nose, ear, pharynx, larynx 로 구성
        Upper part: Palate, soft palate(velum), uvula(목젖), larynx(후두), alveolar.
        Lower part: lip, epiglottis(후두개), tongue → tip, blade, body.
- Articulation
    - Phonation process(with larynx)
        Larynx(voice box) open → voiceless (no vibration on vocal cord)
                closed → voiced
    - Oro-nasal process(in velum)
        Velum lowered → nasal (nasal tract open), 코로 숨쉴 때
            Raised → 비음 뺀 모든 자음 + 모음
    - Articulatory process(in lips, tongue tip, tongue body → 3 constrictors)
        CL(좌우) → Lip(bilabial, labiodental), tongue body(Palatal, velar), tongue tip(dental, alveolar, retroflex, palate-alveolar)
        CD(상하) → Stops > Fricatives > Approximants(r,l,w,j) > Vowels
        즉, By specifying velum, larynx, constrictors, CD, CL ➔ 소리가 결정된다.
        + 모든 모음은 constrictor 중 tongue body 만 사용한다.

< 4 주차 9/24, 9/26>

- Phoneme
  - individual sounds that form words. ex. 'psycho'=orthography, /s aɪ k oʊ/=phonemes
- Acoustics in Praat(DSP: Digital Signal Processing)
  - Pitch = 높낮이, 내가 1 초에 몇 번 larynx 를 움직였는가에 대한 계산. 즉 vocal fords 가 vibration 하는 것이 repeating event 이며, 이때 sine wave 가 repeating 하는 것이 성대의 떨림과 일치
  - Intensity = 크기 ➔ Pitch 와 intensity 는 서로 독립적.
  - 스펙트로그램에서 나타나는 4 개의 빨간띠를 formant 라 하며, 제일 아래부터 F1~F4 라고 부른다. F1, F2 가 무엇이냐에 따라 모음이 뭔지 결정하며, 즉 모음을 구분하는 수치로서 formant 가 쓰인다.
- Complex tone in spectrum
  - Sine wave = 가장 기본적인 시그널의 형태. X 축=시간, Y 축=value. Frequency 와 magnitude 에 따라서 웨이브의 형태 결정.
  - Time_value 그래프인 Sine wave 는 frequency_amplitude 그래프로 더 단순하게 표현할 수 있으며, 이때 frequency_amplitude 의 막대 그래프를 Spectrum 이라고 한다.
  - 스펙트로그램은 스펙트럼을 시간 축을 따라 길게 visualize 한 것이며, 스펙트로그램의 한 슬라이스가 스펙트럼이다.(Spectrum=Spectral slice) 즉, 스펙트럼에는 시간 개념이 없다. 스펙트로그램에서 X 축=시간, Y 축=frequency.

➔ 이 세상에 존재하는 모든 시그널은 서로 다른 사인 웨이브의 결합으로 표현된다.(Simplex tone→Complex tone) 즉, 모든 신호는 조금씩 다른 사인웨이브의 합으로 표현될 수 있다. 복잡한 형태의 어떤 것을 단순한 형태의 것들로 표현할 수 있다!

- Human voice source
  - 소리가 어떤 소리로 만들어 지는가는 성대가 아니라 입에서 어떻게 하느냐에 따라 결정됨

  즉, 성대에서 나는 소리를 source 라고 하고, 튜브(입)에서 어떻게 소리가 달라지는가를 filter 라고 한다. Filter 가 어떻게 달라지느냐에 따라 /아/ /이/가 달라지는 것.
  - Source 만 가지고 spectral analysis 를 하면 gradually decreasing 하는 형태를 보인다. 맨 처음 값이 F0(fundamental frequency)로서 pitch 값과 똑같고, 그 뒤의 값들은 F0 의 X2 값, X3 값으로 쭉 이루어지는데, 이를 harmonics 라고 한다.

  → Human voice source 는 harmonics 로 이루어져 있고, harmonic 는 sine wave 의 배음으로 이루어져있다. 이때 나타나는 amplitude of pure tone 의 패턴은 gradually decreasIng 한다.
- Filtered by vocal tract(filtered=carved)
  - Filtered 된 소리를 spectral analysis 하면 Jignagging with peaks and valleys 하는 형태를 보인다. X2, X3 의 배음의 구조는 깨지지 않았으나, amplitude 의 패턴이 깨져 더이상 gradually decreasing 하지 않는 것이다.
  - EGG(성대녹음) 소리는 gradually decreasing 한데, Audio(실제소리)는 peaks and valleys 의 형태

  → Vocal tracts 에 의해 Filtered 되었기 때문
  - 첫번째 산맥인 F1 은 formants=VT 가 좋아하는 주파수, Valley=VT 가 싫어하는 주파수라고 보면 됨.
  - 각 소리에 따라 어디에 산맥이 나타나느냐가 다르다.
- Source(from larynx)-filter(by vocal tract) theory
  - Voice source(source spectrum) –Vocal tract filter function→ Output from lips(output spectrum)
- Spectrogram = Airplane view of temporal concatenation of spectrum
  - 진하게 나타나는 부분 Dark band = mountains = formants
  - F1, F2 만 보면 충분하며, 그 둘이 달라지는 형태가 모음 도장이라고 생각하면 됨

    F1= 높낮이 결정, F2=Front & back 여부 결정

< 5 주차 10/4>

- 코딩 = 자동화라고 생각
  : 왜 자동화를 할까? 한번 쓰고 버릴 게 아니라 똑같은 걸 반복해서 하니까
- C, java 등 모든 Programming language 의 공통점 = '단어'와 '문법'으로 이루어져있다
  - 단어를 어떻게 combine 해야 커뮤니케이션이 될까?
  - 여기서 단어는 의미(정보)를 포함하고 있다. 단어 = 정보를 담는 그릇.
  그릇 하나가 있을 때 그릇에 사과를 담으면 그 단어는 사과가 되고 물을 담으면 물이라는 단어가 된다.
  한 문장을 얘기할 때 어떤 단어를 선택하고 combine 해야 전달하고 싶은 의미를 다 담을 수 있을까?
  - Computer language 에서 단어 = '변수'. 정보를 담는 그릇으로서 변수가 필요한 것. 거기에 숫자를 담을 수도 있고, 문자를 담을 수도 있고.
- 사람과 기계간의 커뮤니케이션을 위해서는 문법이 필요하다. 기계와의 문법은 그렇게 어렵지 않다.
  1. variable assignment. 변수라는 그릇에 정보를 assign 하는 것.
  2. if conditioning. ~할 때 ~해주세요 하는 조건, communication 에 대한 문법이 필요하다. If.
  3. for loop. 자동화의 가장 중요한 것 중 하나가 여러번 반복하는 것인데 거기에는 for 를 사용한다.
  4. 함수 = 어떤 입력을 넣었을 때 내가 원하는 출력이 나오는 것. (1,2,3 등으로 이용해서) 입력과 출력을 Packaging 하는 것이 바로 함수. 함수 속에 또 함수가 들어갈 수도 있는 등 재사용이 가능하다. 반복적 사용이 가능하다는데 코딩의 의미가 있는 것. (+인간 자체를 함수라고 한다면 음식을 먹고 배출 하는게 함수.)

<주피터 노트북 variables>

- a=1: 오른쪽에 있는 정보를 variable 로 assign. =은 equal sign 이 아님. 변수에는 숫자나 문자가 들어갈 수 있다.
- Print(a): print 는 누가 만들어 놓은 함수. print 의 출력 결과는 화면에 보이는 것 뿐 그 라인은 셀이 아니다. 그래서 out 이 없음. 비교) print 없이 변수 하나만 쳐도 값을 보여주는데, 그때는 out 값.
- a=[1,2,3,5]: 리스트. a 라는 변수 안에 들어있는 데이터의 유형이 뭔지 알려면 type 함수를 쓰면 됨.
- a=[1,2,3,5,'love']: 한 리스트 안에 문자와 숫자 같이 넣을 수 있음.
- a=[1, 'love', [1, 'bye']]: 한 리스트 안에 리스트가 아이템으로서 들어갈 수 있다.
- a=(1,'love', [1,'bye']): 튜플. 리스트에서 대괄호 대신 괄호 쓰는 경우. 튜플과 리스트는 거의 완전히 똑같음.
  → 근데 왜 존재할까? 그냥 튜플이 보안에 좀 더 강하다고 생각하면 됨.
- a = {'a':'apple', 'b':'banana'}: 딕셔너리 자료형. 표제어:설명의 구조, 컴마 보면 몇개 들어있는지 알 수 있음. 즉, 여기서는 컴마가 하나라 2 개가 들어간다고 생각하면 됨. Type(a)=dict.
- In [ ] 부분 누르고 a 누르면 위에 새로운 셀, b 를 누르면 아래에 새로운 셀, x 누르면 셀 사라짐.

< 9 주차 10/29, 10/31>

- 세상의 자료는 다 숫자화가 가능하다. 소리, 이미지, 숫자 데이터(주식, 날씨, 온도...) 숫자화 되어있지 않은 데이터가 어떻게 숫자화 되는가? 그러한 숫자의 열을 벡터라고함.

  ex) 흑백이미지를 흰색 0, 검정 10, 회색 0~10 으로 숫자화해서 나타낼 수 있다. 사진을 직사각형의 행(가로)과 열(세로)로 나타낼 수 있다. → 즉, 사진을 찍는 순간 행렬이 만들어지는 것이다.

- 이미지는 행렬. 행렬 값을 떼어와서 한줄로 나열해서 쓰면 벡터가 된다. (vectorize)

  모든 데이터는 벡터의 형태가 되어야함.

  행렬은 벡터가 아님. 길게 만들어야지 벡터가 됨. (벡터화)

  모든 데이터는 벡터로 만들어야 다루기 쉽다.

- 영상 = 4 차원 → 흑백은 2 차원, 칼라 이미지는 3 차원, 시간까지 있으면 4 차원이기 때문

- 소리) 어떻게 백터이미지가 될까?

  : wave form 확대해보면 결국 점들이 이어진 것이며 그걸 숫자값으로 바꾸어 나열할 수 있다. (벡터화)

- 텍스트) 어떻게 벡터화가 될까?

  : 어떤 사전이 있다고 할 때 1 번째 단어를 벡터화 시킬 때,

100....0000(5 만개) / 2 번째 단어는 010000...000 / 3 번째 단어는 00100....0000 / 5 만번째 단어는 0000....000001

➔ 이런 식으로 텍스트도 벡터화가 가능하다

- <numpy>

: 라이브러리 안에 패키지 A, 패키지 B, 함수가 들어있다. 패키지 안에 또 패키지가 들어있을 수도 있음. 그래서 Ex) numpy 안의 패키지 A 안의, 패키지 D 안의, 함수 f 를 쓰려면 numpy.A.D.f 하면 됨. 이게 복잡하니까 대신 from numpy import A.D 이렇게 할수도 있음.

- Sampling rate 에 대해 생각해보세요