SPEECH SIGNAL PROCESSING PROJECT 2 REPORT

Due date : 5/27

Professor : 장길진 교수님

제출자: 2022226464 전인구

Voiced, Unvoiced and Silence Detection on TIMIT dataset

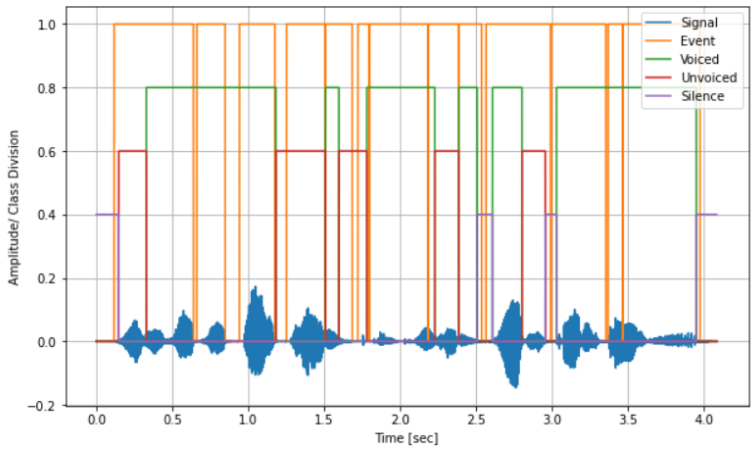
* 설명의 연속성을 위해 제출한 코드와 순서가 조금 다를 수 있습니다.

ARPAbet의 발음 기호와 TIMIT에서 사용되는 발음 기호들을 기반으로 Voiced, Unvoiced, Silence로 구분합니다. 제출한 코드상에서는 아래의 기준으로 구분했습니다.

```
ARPAbet set:

{'uh', 'epi', 'v', 'p', 'z', 'sh', 'w', 'hv', 'ay', 'q', 'tcl', 'aa', 's', 'n', 'ax', 'ao',
'ah', 'h#', 'ax-h', 'ux', 'axr', 'gcl', 'dh', 'ch', 'uw', 'eng', 'aw', 'nx', 'er', 'kcl', 'm',
'eh', 'ey', 't', 'el', 'zh', 'ow', 'iy', 'em', 'ng', 'wh', 'jh', 'th', 'b', 'dcl', 'pau', 'k',
'ae', 'oy', 'd', 'bcl', 'en', 'g', 'r', 'dx', 'l', 'pcl', 'f', 'y', 'hh', 'ih', 'ix'}

voiced set:

{'uh', 'v', 'z', 'w', 'ay', 'q', 'aa', 'n', 'ax', 'ao', 'ah', 'gcl', 'ux', 'axr', 'dh', 'uw',
'eng', 'aw', 'nx', 'er', 'm', 'eh', 'ey', 'el', 'zh', 'ow', 'iy', 'em', 'ng', 'wh', 'jh', 'b',
'dcl', 'ae', 'oy', 'ih', 'd', 'bcl', 'y', 'en', 'g', 'r', 'dx', 'l', 'hv', 'ix'}

unvoiced set:

{'ch', 'p', 'th', 'sh', 'k', 't', 'pcl', 'tcl', 'f', 's', 'hh', 'kcl'}

silence set:

{'epi', 'pau', 'h#', 'ax-h'}
```

그 후, EPD와 ZCR를 이용해서, Silence인 부분은 0을, 그 외의 부분(Unvoiced, Voiced)은 1으로 구분했습니다. 바로 아래에 구분 결과와 알고리즘의 설명이 있습니다.

```
Class Division:

Event       : 1/0 (VAD sense ON / OFF)
Voiced      : 0.8/0 (VAD sense ON / OFF)
Unvoiced    : 0.6/0 (VAD sense ON / OFF)
Silence     : 0.4/0 (VAD sense ON / OFF)
```



x : 입력 신호, Nf : 프레임의 길이

EPD

$Dec= (x^2 > 10^{-5})+0$

$Dec=Mean\_filtering(Dec, Nf//4)$ # 음성의 유무 판단

ZCR

$Zcr= (|x| > 5 \times 10^{-6})+0$ # 진폭이 너무 작은 부분

Combination

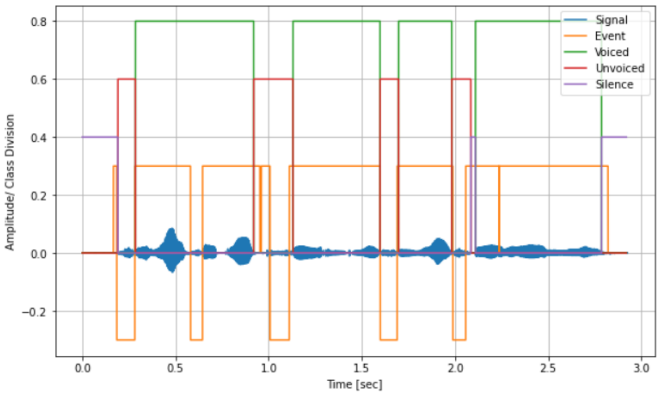
$COMB= (Dec \times (1 - Zcr) > 0)+0$

진폭이 어느 정도 크면서, 에너지가 큰 부분을 silence가 아니라고 판단한다. 음성이 있다면 1, 없다면 0으로 표시.

위의 알고리즘을 적용한 뒤에, 너무 짧은 시간 (3Nf보다 작은 경우)동안 일어나는 사건에 대해서는 앞뒤로 1.5Nf의 margin을 줘서 각각의 이벤트의 길이가 3Nf 이상이 되도록 만들었습니다.

위의 그림상에서 주황색 선이 음성의 유무를 판단한 것들입니다. 오차가 있긴 하지만, Silence를 찾아낼 수 있음을 확인할 수 있습니다. 그림 상의 voiced, unvoiced, silence는 .phn파일에서 얻어진 label이며, 0보다 큰 값들로 표시된 부분들이 해당 label임을 의미합니다.

Voiced, Unvoiced의 구분은 silence가 아닌 부분에 auto correlation을 사용해서 구현했습니다.
구분한 결과와 알고리즘에 대한 설명이 아래에 있습니다.



알고리즘이 구분한 결과가 주황색 선으로 위의 그림에 표현되어 있습니다. 주황색 선이 0보다 큰 부분이 voiced, 0인 부분이 silence, 0보다 작은 부분이 silence를 의미합니다. Unvoiced와 voiced가 적은 오차로 잘 구분되고 있음을 확인할 수 있습니다.

```
Ns : skip length (=Nf//2), mean : 평균 연산

For idx in range(len(x)//Ns):
    x_i=x[idx*Ns: (idx+1)*Ns]
    r0=mean(x_i*x_i) # autocorrelation (딜레이 : 0)
    r1=mean(x_i[1:]*x_i[:-1]) # autocorrelation (딜레이 : 1)
    result[idx*Ns: (idx+1)*Ns]=( (r1/r0)>0.5 )+0

x_i=x[idx*Ns:] # 끝 부분에도 같은 연산을 한다.
r0=mean(x_i*x_i) # autocorrelation (딜레이 : 0)
r1=mean(x_i[1:]*x_i[:-1]) # autocorrelation (딜레이 : 1)
result[idx*Ns:]=( (r1/r0)>0.5 )+0

return (result >= 0.5)+0 - (result < 0.5)+0
# voiced인 부분을 1, unvoiced인 부분을 -1로 구분한다.
```

Unvoiced는 파열음이나 마찰음으로 구성되어 있고, voiced는 성대의 울림으로 만들어진 tonal signal 입니다. tonal signal의 자기 상관성이 파열음이나 마찰음의 자기 상관성보다 크기 때문에 autocorrelation으로 voiced, unvoiced를 구분할 수 있는 것 같습니다.

앞서 설명한 알고리즘의 구분성능을 정확도(accuracy)로 확인해봤습니다. 우측의 그림처럼 voiced, unvoiced, silence를 학습 데이터에서는 85%이상의 정확도로, 검증 데이터에서는 90%이상의 정확도로 구분하는 것을 알 수 있습니다.

```
Class Division:

Event           : -0.3/0/0.3 (Unvoiced / Silence / Voiced)
Voiced          : 0.8/0 (VAD sense ON / OFF)
Unvoiced        : 0.6/0 (VAD sense ON / OFF)
Silence         : 0.4/0 (VAD sense ON / OFF)

==========================================

TRAINING DATA PERFORMANCE [accuracy]
Voiced : 87.38%,        Unvoiced : 88.74%,      Silence : 91.17%

==========================================

VALIDATION DATA PERFORMANCE [accuracy]
Voiced : 90.34%,        Unvoiced : 91.08%,      Silence : 96.42%
```

```
sa data ==========================================

TEST DATA PERFORMANCE [accuracy]
Voiced : 90.57%,        Unvoiced : 92.87%,      Silence : 94.53%

sx data ==========================================

TEST DATA PERFORMANCE [accuracy]
Voiced : 87.45%,        Unvoiced : 86.23%,      Silence : 95.55%
```

시험 데이터에 적용한 결과는 좌측의 그림으로 나타나 있습니다. sa데이터에서는 90% 이상의 정확도를 보여줬고, sx 데이터에서는 약 85% 이상의 정확도를 확인할 수 있었습니다.