BIL735 Konuşma Tanıma

Emre Kağan Akkaya

N14128491

İçerik

- Otomatik son bulma
- Öznitelikler in çıkarılması (MFCC)
- İzole kelime tanıma (DTW)
- Dil modelinin eğitilmesi (Fonemler)

Otomatik son bulma

- Konuşmanın olduğu bölgenin olmadığı bölgeden ayırt edilebilmesi işlemidir.
  - Konuşmanın olduğu bölgelerin enerjisi, olmadığı bölgelerden daha fazla
- Konuşma tanımada, sözlü ifadenin son noktalarının bulunması anahtar faktörlerden biri
  - Konuşmanın olmadığı bölgelerde konuşma tanımadan kaçınmak gerekli
    - Var olmadığı halde kelimelerin saptanması
    - Gereksiz hesaplama yükü
  - Çevresel faktörler, arka plan gürültüsü

Otomatik son bulma

- Diğer birçok yöntemin (threshold, entropy, zero-crossing rate, two-threshold gibi) yanında **adaptif son bulma** değişken arka plan gürültüsüne uyum sağlayarak çalışan yaklaşımlardan biridir.
  - Örneklem başına enerji seviyesi hesaplanır.
  - Enerji seviyesi arka plan seviyesi ile karşılaştırılır.
  - Karşılaştırma sonucuna göre arka plan seviyesi (uyum sağlar) güncellenir.
    - Başlangıçta ilk 10 çerçevenin enerji ortalaması
    - Adjustment: ne kadar hızla uyum sağlayacağını belirleyen parametre
  - Enerji seviyesi pürüzsüzleştirilerek (smoothing) arka plan seviyesiyle karşılaştırılır.
    - Elde edilen seviye, arka plandan belirli bir eşik değer kadar daha yüksekse konuşma tanımlanır.

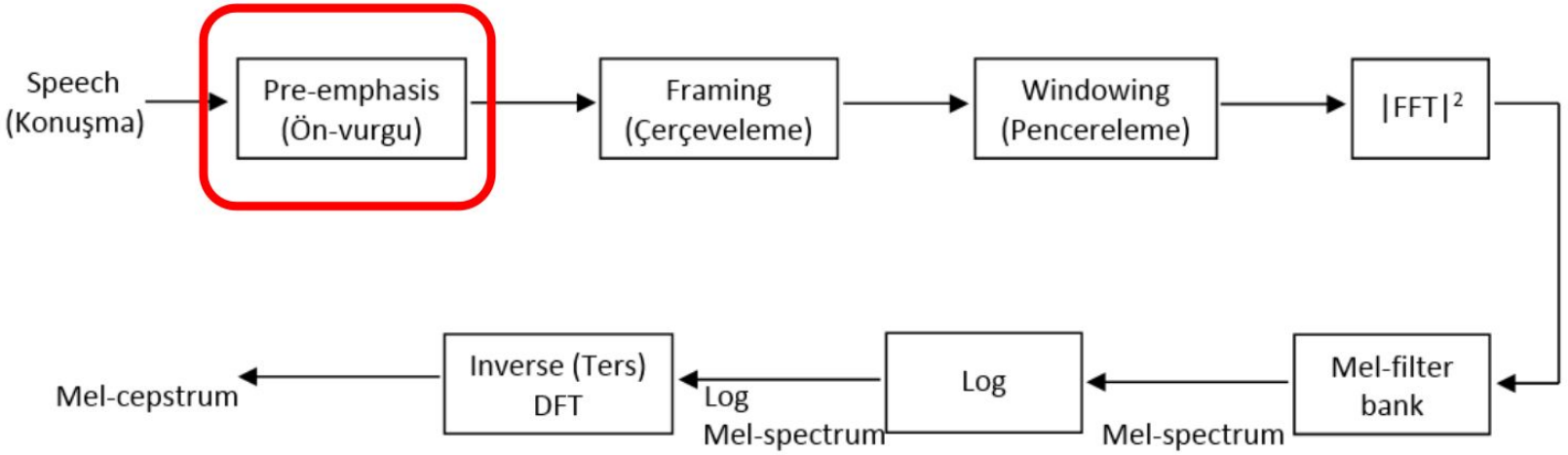
Otomatik son bulma

- Kullanılan yöntem
  - ✓ Python 2.7
  - ✓ PyAudio ile non-blocking veri kaydı
  - ✓ 44.100Hz örneklem
  - ✓ 16-bit PCM
  - ✓ Adaptif son bulma gerçekleştirimi

```
Function classifyFrame(audioframe):
    current = EnergyPerSampleInDecibel(audioframe)
    isSpeech = False
    level = ((level * forgetfactor) + current) / (forgetfactor+ 1)
    if (current < background):
        background = current
    else:
        background += (current - background) * adjustment
    if (level < background):   level = background
    if (level - background > threshold): isSpeech = True
    return isSpeech
```

MFCC - özniteliklerin çıkarılması

- Konuşma tanımadaki ikinci adım, sesin karakteristiğini özetleyen özniteliklerin çıkarılması
  - En sık kullanılan öznitelik **Mel-frekans cepstrum**

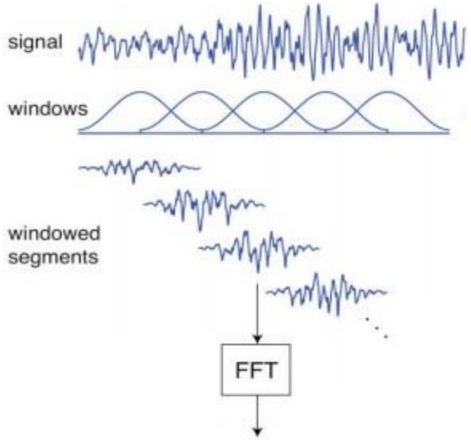


MFCC - özniteliklerin çıkarılması

1. **Ön-vurgu**
    a. Yüksek frekanslara kıyasla düşük frekanslarda sese ait spectrumun daha fazla enerjisi vardır (*spectral tilt*)
    b. Yüksek frekanslardaki enerji vurgulanarak akustik modele daha fazla katkı yapılır. (tanımadaki başarımı arttıran iyileştirmelerden bir tanesi)
2. **Çerçeveleme**
    a. Birbiriyle örtüşen ve N örnek nokta içeren çerçeveler oluşturulur.
        i. Genellikle 20ms'lik örtüşen kısımlardan oluşan 30ms'lik çerçeveler kullanılır. Eğer çerçeve daha küçük olursa güvenilir bir tahmin yürütmek için yeterince örnek alınamaz, daha büyük olursa da sinyal çerçeve boyunca çok fazla değişir.

MFCC - özniteliklerin çıkarılması

3. **Pencereleme**
   a. DFT hesaplaması öncesi spektral etkinin azaltılması, sinyalin pürüzsüzleştirilmesi (*smoothing*)
   b. Sinyalin çerçevedeki başlangıç ve bitiş kesintileri sıfıra yaklaştırılarak kesintilerin önüne geçilmiş olur.
   c. Genellikle Hamming yöntemi kullanılmakta (Hanning, Blackman, Gauss, rectangular, triangular...)



MFCC - özniteliklerin çıkarılması

4. **Fast Fourier Dönüşümü (FFT)**
   a. Periyodik sinyalin Fourier spectrumu olarak nitelendirilebilir
   b. Her bir çerçevedeki N örneğin zaman alanından frekans alanına çevrimi için kullanılır.
   c. Bu sayede sinyal işlenebilir alana çekilmiş olur.
   d. FFT büyüklüğü genellikle 512, 1024 veya 2048

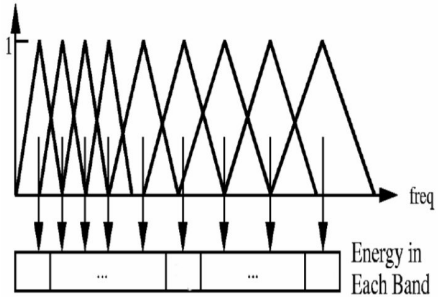
$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\frac{\pi}{N}kn}$$

MFCC - özniteliklerin çıkarılması

5. Mel-filtre dizisi

a. İnsan kulağı tüm frekans bantlarına eşit şekilde duyarlı değil
   i. Yüksek frekanslara daha az duyarlı. **Kabaca > 1000 Hz**
   ii. Kulak sadece belli frekans bileşenlerine yoğunlaşacak şekilde filtreleme işlevi görür. (***Band-pass filters***)
   iii. Bu filtreler frekans ölçeğinde düzensiz şekilde dağılmıştır. (düşük frekanslarda daha fazla filtre, yüksek frekanslarda daha az filtre)

b. Benzer şekilde Mel ölçeğine göre spektruma bir dizi filtre uygulanır.



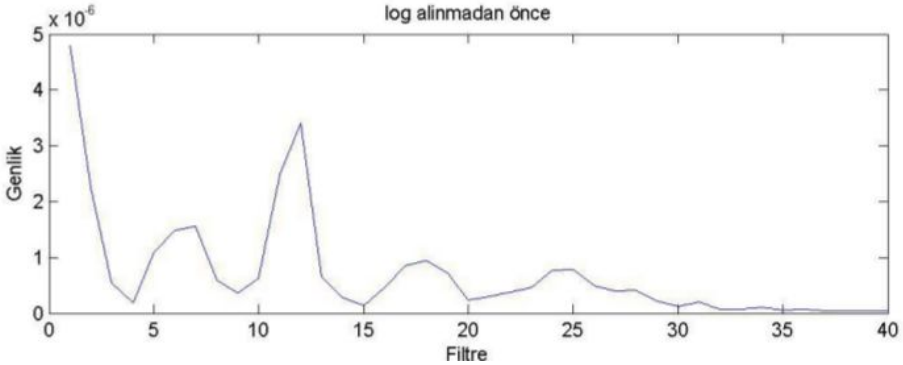
MFCC - özniteliklerin çıkarılması

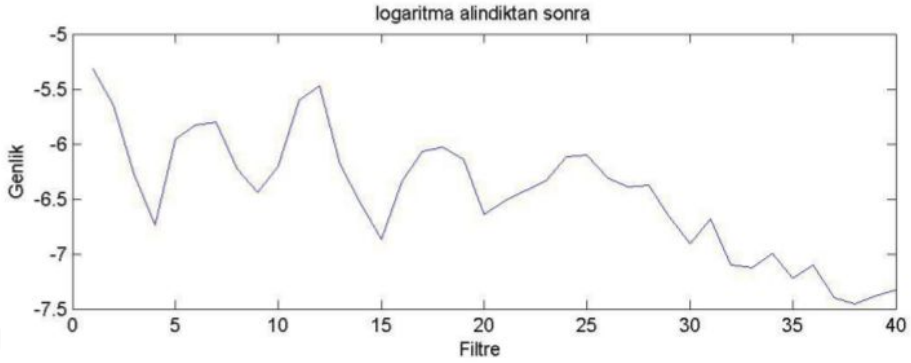
6. Log

a. Mel spektrum büyüklüğünün karesinin logaritması alınarak hesaplanır.

b. Logaritma sinyal genliğinin değişken aralığını sıkıştırarak insan kulağının davranışını taklit eder diyebiliriz.

   i. İnsanlar yüksek genlikteki küçük değişikliklere düşük genliktekilere kıyasla daha duyarsızdır.



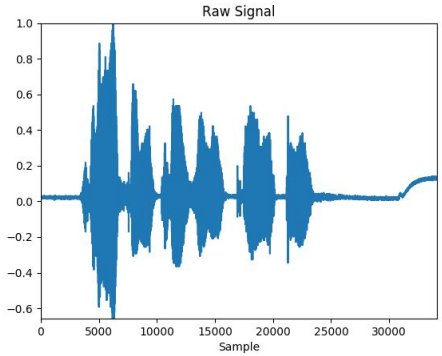


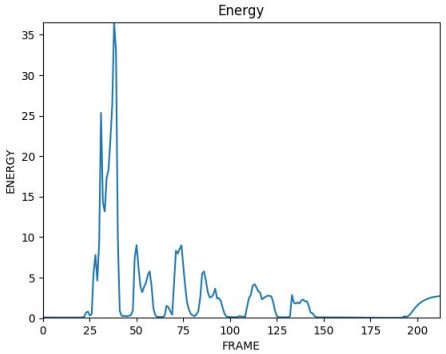
MFCC - özniteliklerin çıkarılması

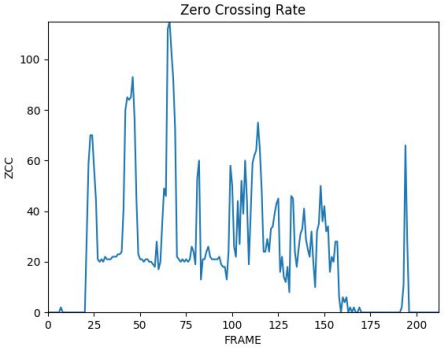
7. **Ters DFT dönüşümü**
   a. Bir önceki adımda elde edilen log Mel-spektrumun zaman alanına geri çevrilmesi olarak nitelendirilebilir. (***Discrete Cosine Transform/DCT***)
   b. Bu sayede her bir sesli ifade bir dizi akustik vektöre çevrilmiş oldu.
   c. MFCC:
      i. 12 MFCC (mel frequency cepstral coefficients)
      ii. 1 enerji özniteliği
      iii. 12 delta MFCC özniteliği
      iv. 12 çift-delta MFCC özniteliği
      v. 1 delta enerji özniteliği
      vi. 1 çift-delta enerji özniteliği

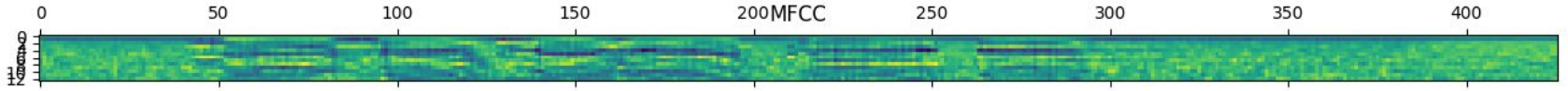
MFCC - özniteliklerin çıkarılması

- Kullanılan yöntem
  - ✓ Python 2.7
  - ✓ MFCC
- Sonuçlar



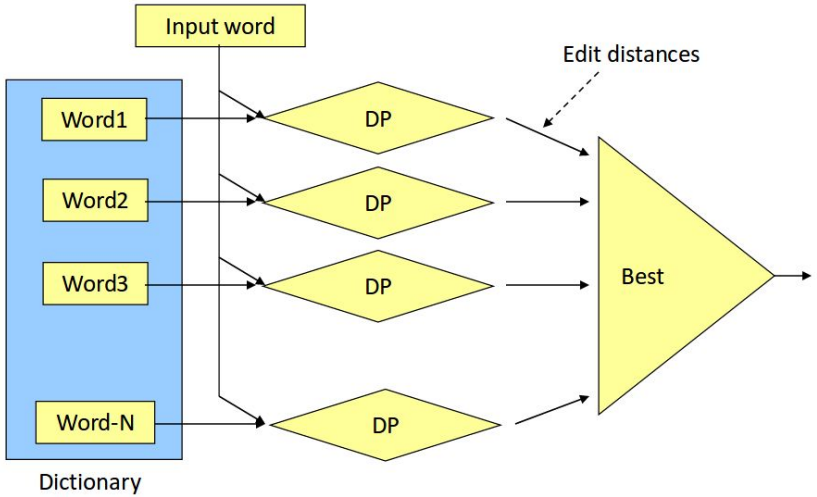






DTW - izole kelime tanıma

- Kelimenin tanınması için izlenebilecek en temel yol:
  - Kelime şablonlarından oluşan bir sözlük oluşturulur
    - şablon = karşılaştırma amacıyla kullanılan kayıt
  - Girdi sözcük sözlükteki her bir şablonla karşılaştırılır
  - Girdi sözcüğe en çok benzeyen şablon seçilerek sözcük tanımlanmış olur.

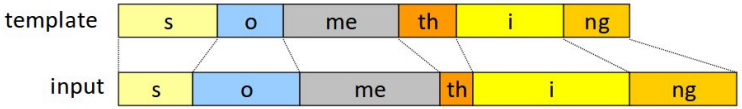


DTW - izole kelime tanıma

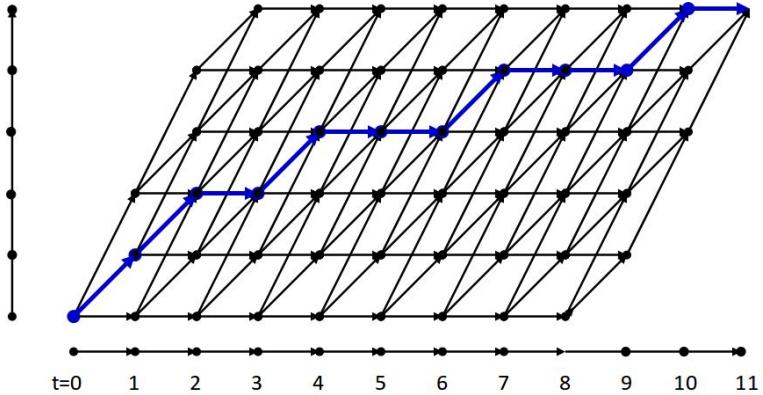
- Karşılaştırma temelde bir kelimenin diğerine dönüştürülmesi için yapılması gereken minimum sayıdaki değişikliğin belirlenmesi
  - 2 boyutlu diyagrama oturtulup değişiklik sayısıyla ölçülebilir
  - Temelde yol bulma problemi!
- **Dinamik programlama** (DP)
  - 2 boyutlu diyagram (***trellis***) üzerinde en uygun alt-yollar kullanılarak optimum (minimum maliyetli/maksimum skorlu) yol bulunabilir.
    - Arttırımlı olarak birimler (string ise harf, ses ise öznitelik vektörleri) karşılaştırılır
    - Her bir karşılaştırma bir öncekinin sonuçlarına dayanır
  - Tüm konuşma tanıma sistemlerinde temel alınan yaklaşımlardan!

DTW - izole kelime tanıma

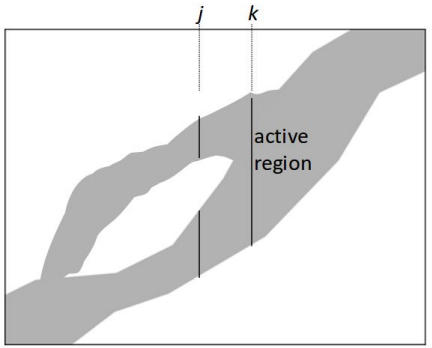
- Sorun: Girdi sözcük ile şablon farklı uzunluklarda olabilir.
  - Uzunluktaki farklılıklar düzensiz de olabilir



- Çözüm: Dinamik Zaman Saptırması (DTW)
  - İki dizilim arasında optimal bir eşleşme bulmak için kullanılır.
  - Temelde vektör hizalama! Trellis üzerinde her bir düğümde girdi ve şablona ait vektörler karşılaştırılır (**Öklid uzaklığı, Manhattan, Minkowski**).
  - En iyi yol maliyeti (DP skor) karşılaştırma maliyetinden hesaplanır.



DTW - izole kelime tanıma



- Doğru tanıma oranını arttırma
  - Birden fazla şablon kullan (tercihen birden fazla konuşmacıdan)
  - Arama maliyeti artar
- Arama maliyetinin düşürülmesi
  - **Time-sync DTW?**
  - **Budama *(pruning)***:
    - Karşılaştırmada en iyi maliyet X ise, göreli T eşik değerini kullanarak >X+T değerinden büyük maliyetli düğümleri ele (**beam search**) Sadece geriye kalan düğümler bizi ilgilendirir.
    - Çok küçük eşik değeri en iyi yolun da budanmasına neden olabilir.
      - Eşiği belirlemek için deneme-yanılmadan başka yaklaşım yok...

DTW - izole kelime tanıma

- Kullanılan yöntem
  - ✓ Python 2.7
  - ✓ DTW
- Sonuçlar
  - ✓ 1 şablon kullanıldığında doğruluk oranı: 0.5333
  - ✓ 2 şablon kullanıldığında doğruluk oranı: 0.5125
  - ✓ 3 şablon kullanıldığında doğruluk oranı: 0.7285
  - ✓ 4 şablon kullanıldığında doğruluk oranı: 0.6666
  - ✓ 5 şablon kullanıldığında doğruluk oranı: 0.64
  - ✓ 5 şablon (budama ile birlikte) kullanıldığında doğruluk oranı: 0.74

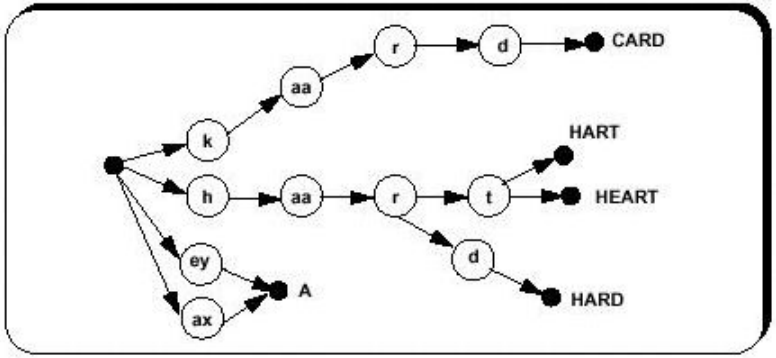
Fonemler - modelin eğitilmesi

- Kelime-tabanlı model ile tanımada her kelime için model eğitilmelidir
  - Her cümle için şablonların oluşturulması... Çok fazla veri gerekli
  - Bilinmeyen kelimelerin eğitilmeden tanınamaması...
  - Sözcükten daha temel bir yapıya ihtiyaç var
- Herhangi bir dildeki kelimeler bir takım seslerin sıralı söylenişiyle ortaya çıkar
  - Sözcük-altı ses birimlerine **fonem** adı verilir. Fonemlerin bir araya gelmesiyle kelimeler oluşur.
  - İngilizce'de yaklaşık 40 fonem mevcut. Bu da eğitilen yaklaşık 40 model ile herhangi bir konuşmanın tanınabilmesi anlamına gelir...
  - Fonemler ile sözcükler arasındaki çevrimin de tanımlanması gerekir (**Sözlük yada lexicon**)

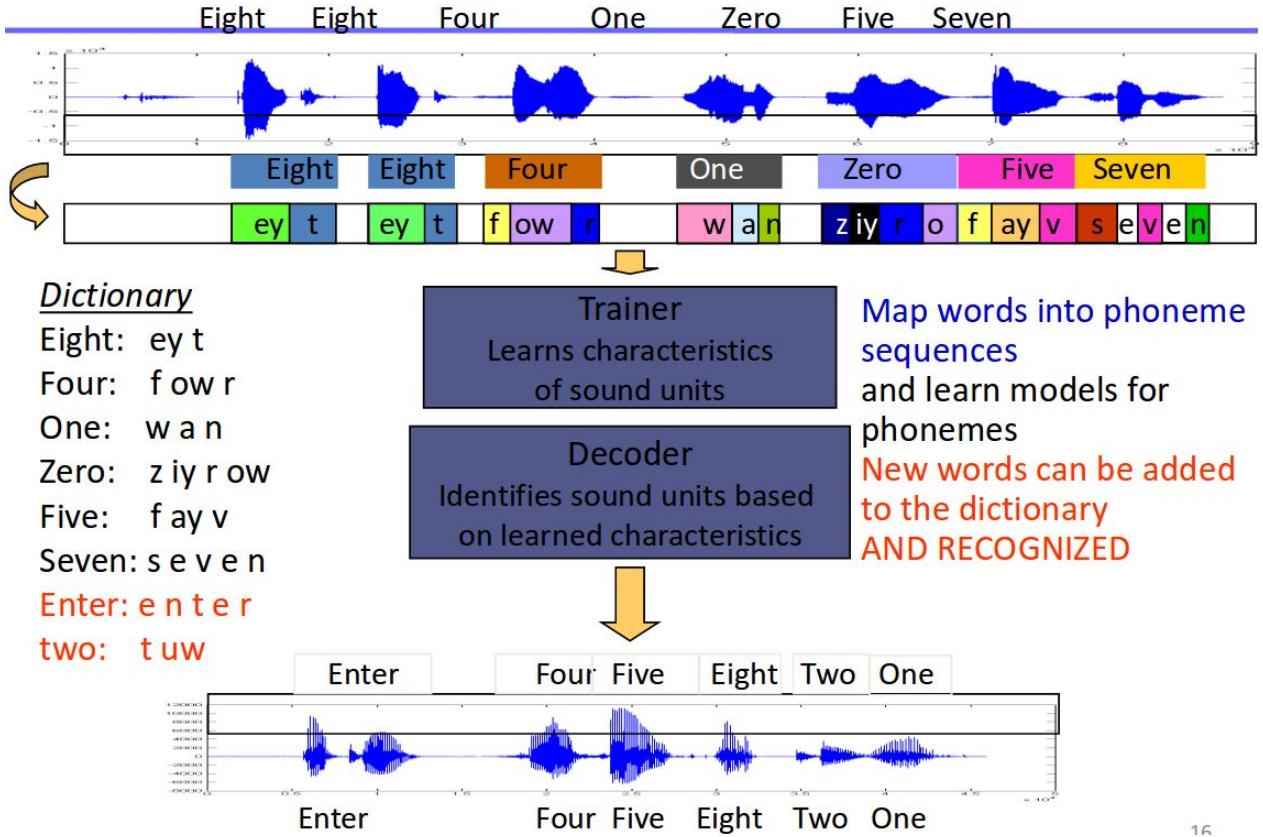
```
ONE:    W AX N
TWO:    T UW
THREE:  TH R IY
FOUR:   F OW R
FIVE:   F AY V
SIX:    S IH K S
SEVEN:  S EH V EH N
EIGHT:  EY T
NINE:   N AY N
ZERO:   Z IY R OW
```

Fonemler - modelin eğitilmesi

- Veri kümesindeki her bir kelime bir dizi foneme çevrilir
- Her bir fonem için HMM eğitilir.
  - **HMM** eğitildiği sınıfa karşılık verilen bir girdi dizisinin olasılığını tahmin etmek için kullanılan istatistiksel bir modeldir.
  - Fonem düzeyinde eğitilen HMM'ler konuşmanın tanınmasında kullanılmakta (**Akustik model**)
  - Tanıma hala kelime düzeyinde, sadece fonem eğitmek için HMM kullanılıyor.
  - Daha sonra HMM'ler bir araya getirilerek kelime düzeyine çıkılmış oluyor.

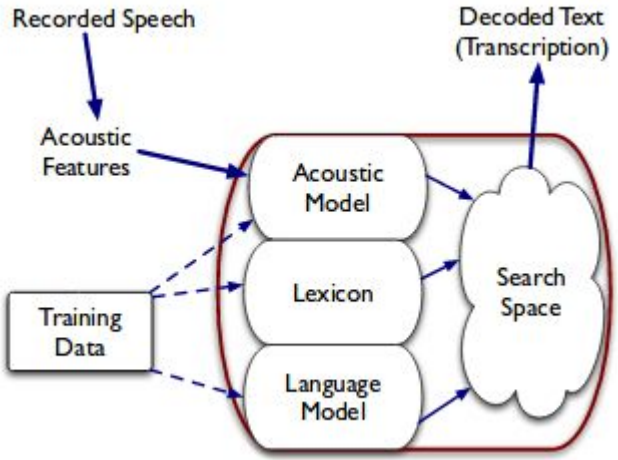


Fonemler - modelin eğitilmesi



Fonemler - modelin eğitilmesi

- Konuşmanın tanınması için kelime tanımaktan daha fazlası gerekli
  - **Dil modeli** olmadan bir kelimeden sonra herhangi bir kelimenin elde edilmesi olası.
  - Dil modeli herhangi bir arama anında bir kelimeden sonra hangisinin gelebileceği konusunda kısıtlama sağlayarak daha hızlı çalışma zamanı ve daha doğru sonuçlar sağlar (**n-gram model**)
  - Genellikle tri-gram kullanılarak hesaplanır.



Dinlediğiniz için teşekkürler!