11-737 Multilingual NLP

Speech ASR
Automatic Speech Recognition

- Speech to text
- Audio signal to words
- You will need (standardly)
  - Transcribed audio
  - Pronunciation lexicon
  - Language model
- But we will see how to continue with these
Voice Dialing System

• Library
  – Mom
  – Dad
  – Bob
  – Mario’s Pizza
  – Let’s Go Bus Information System
Matching in Frequency Domain

Mom

Bob
Dynamic Time Warping

Template

Sample Speech
For each square

- \( \text{Dist}(\text{template}[i], \text{sample}[j]) + \) 
  - \( \text{smallest}_\text{of} \ (\text{Dist}(\text{template}[i-1], \text{sample}[j])) \) 
  - \( \text{Dist}(\text{template}[i], \text{sample}[j-1]) \) 
  - \( \text{Dist}(\text{template}[i-1], \text{sample}[j-1]) \)

Remember which choice your took (count path)
Matching Templates

For Word in Templates
    Score = dtw(Template[Word], Sample);
    if (Score < BestScore)
        BestWord = Word;
    DoAction(Action[BestWord])
DTW issues

- **What happens with no-matches**
  - Need to deal with none of the above

- **What happens with more templates**
  - Harder to choose between
  - Once variance greater than differences

- Choose templates that are very different
More reliable matching

- Distance metric
  \[ \sqrt{\sum_{i=0}^{N} (T_i - S_i)^2} \]

- But some distances are bigger than others
  - Silence is pretty similar
  - Fricatives are quite larger
    - A longer fricative might give large score
    - A longer vowel might give smaller score
More reliable matching

- Having multiple template examples
  - Individual matches or
  - Average them together
- DTW align all of the examples
- Collect statistics as a Gaussian
  - Mean and standard deviation for each coeff

\[ \{ \mu_0, \sigma_0, \mu_1, \sigma_1, \mu_2, \sigma_2, \ldots \} \]
More reliable distances

• Instead of Euclidean distance
  – Doesn’t care about the standard deviation
  \[ \sqrt{\sum_{i=0}^{N} (T_i - S_i)^2} \]

• Use Mahalanobis distance
  – Care about means and standard deviation
  \[ \sqrt{\sum_{i=0}^{N} \left( \frac{\mu_i - S_i}{\sigma_i} \right)^2} \]
Extending template model

- String phoneme templates together
  - A template model for each phoneme

Sample

Phoneme Templates

k  ae  t
But Phones aren’t isolated

- Phones are dependent on their context so
  - We need an inventory of contextualized phones
  - (pre) P (post)
  - Triphones (contextualized phones, not a trigram)
- But we need examples to train from
  - 45 phones in all (91K triphones)
  - At least 10ish examples of each … 200hrs
  - But phones are not equi-probable
  - Need to deal with approximate triphones
Training an acoustic model

- Transcribed speech
  - Text (with pronunciations) of what is actually said
  - The audio (in a standard format)
- How much
  - At least the phones in context
  - 1000s of hours is good
  - 100s of hours will work
  - 20 hours
  - 1 hour
  - zero
- We’re trying to build “dtw” like models for each contextualized phone
  - HMMs (Baum-Welch)
  - Neural
Not just acoustics

• But not all phones are equi-probable
• Find word sequences that maximizes

\[ P(W \mid O) \]

• Using Bayes’ Law

\[ \frac{P(W)P(O \mid W)}{P(O)} \]

• Combine models
  – Use acoustic model to provide \( P(O \mid W) \)
  – Use language model to provide \( P(W) \)
Language Model

- Estimate cost of sequence of words in the language
  - Need **appropriate** training data
  - At word, token (BPE), character/phone level
  - Exploit pretrained models
    - Multiple models and combine them
- LM is traditionally
  - Smoothed trigram word model with backoff
    - Because that captures most of requirements
    - LM is not a grammaticality test, its choosing between similar phone strings
- Separate LM
  - Allows good use of other data (boosting performance)
  - There may be a mismatch between LM data and AM data
  - That different may be a good thing (or not)
Pronunciation Model

- ASR doesn’t work unless it knows the words
  - We need pronunciation for every word in vocabulary
  - Pencil → p eh n s ih l
- CMUDICT 150K English words with pronunciation
  - Good for names, good for actual speech
  - But not always consistent
- But I don’t have a pronunciation lexicon
  - Use the graphemes (letters)
  - Use epitran (David Mortenson)
Measuring ASR Success

- Word Error Rate (WER)
  - Substitutions: word is replaces
  - Deletions: word is missed out
  - Insertions: word is added

\[
\text{Subs} + \text{Dels} + \text{Ins} \\
\text{WER} = 100 \times \frac{\text{Subs} + \text{Dels} + \text{Ins}}{\text{word in correct sentence}}
\]

Wer can be >100%
How good is good?

● What is a good WER?
  • Read speech (dictation) from “good” performance <5% WER
  • Spoken Dialog (task oriented) < 30%
  • Drunk friends in outside busy café <80%
● Collect data in your application and retrain
  • Application specific data is very valuable
ASR Summary

- ASR components
  - Acoustic model from Transcribed Audio
  - Pronunciation model from Lexicon
  - Language model from text data
- Want training to contain all phones in all contexts (from all speakers)
- WER standard measure (lower is better)
  - Applications may interpret ASR output even at 30% WER
- Next time
  - How to do ASR without the components
ASR Discussion Point

• ASR requires (or at least wants) acoustic model, pronunciation model and language model
• For languages you know find examples where these three components would have different relative importance in an ASR system.
  • e.g. Spanish: pronunciation model can just be characters (mostly)
  • Arabic (spoken form is dialect, so LM from (MSA) text wont really help