Sample FSA homework

- Video game
- The Plan of Salvation
- My Morning Routine
- Fruits and Vegetables Parser
- Germanic words for “book”
- Weather forecast

- Traffic lights in Romania
- Beatles song lyrics
- Arabic perfective 3rd person verbs
- Plurals in Tolkein’s Sindarin
- How to make friends
Task #4 from last semester

- foma was the most popular.
- English: parse verbs, generate present participles, spell checker, negative allomorphs, plurals, velar softening
- Persian plurals, diminutives
- Greek verbs
- Hebrew verbs
- Portuguese plurals

- I wish I were better at understanding my own language.
- Any useful FSM in linguistics is probably going to be pretty massive.
- I can definitely see how to make a FSA with any real value and complexity you have to have some linguistic knowledge.
- You really have to have a good working knowledge of the morphological rules that have already been documented by linguists. Thank goodness for those who put in the work to figure these rules out already!
- A lot of the linguistic stuff is pretty interesting, trying to determine rules for things that I have done my entire life.
- This assignment brought to light the fact that writing all-encompassing linguistic rules can prove very challenging and requires extremely thorough research/planning.
COMPUTATIONAL PHONETICS, PHONOLOGY

(Computing language sounds)
Languages and sounds

• Over 7100 languages, many more dialects
• Sound segments (aka “phones”): individual sounds in isolation (e.g. p, b, g, h)
  • Relatively small set of possibilities: limited by human anatomy
• 3 basic classes of phones:
  • Vowels: no substantial constriction of airflow (3-64 depending on language)
  • Consonants: some kind of substantial constriction of airflow (8-98 depending on language)
  • Glides: half-way between V and C (y, w)
• Speech processing: phones, diphones, triphones
• Syllable: coherent structure of associated sounds
• Phoneme: combined contextual variants of a basic sound in the language
  • The basic phoneme /l/ in English has two variants (voiced and voiceless).
  • The basic phoneme /t/ in English has as many as eight (!) variants
  • Allophones: variant forms of a basic sound segment
Phonetics

• Definition: the science of describing and inventorying the features and the structure of the basic sounds of speech in different languages

• Two main branches
  • Articulatory: body parts (articulatory apparatus) and how they’re used in speech production
  • Acoustic: sound waves and their properties

• NLP deals with both
International Phonetic Alphabet (IPA)

- Systematic way to represent all sounds in all languages
- Based on articulatory features
- Great for linguistics, not very practical for many speech tools
- Often used for corpus annotation

### CONSONANTS (PULMONIC)

<table>
<thead>
<tr>
<th>Place of Articulation</th>
<th>Bilabial</th>
<th>Labiodental</th>
<th>Dental</th>
<th>Alveolar</th>
<th>Postalveolar</th>
<th>Retrolabial</th>
<th>Palatal</th>
<th>Velar</th>
<th>Uvular</th>
<th>Pharyngeal</th>
<th>Glottal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fricative</td>
<td>p, b</td>
<td>t, d</td>
<td>c, f</td>
<td>k, g</td>
<td>q, g</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nasal</td>
<td>m</td>
<td>n</td>
<td>n</td>
<td>ñ</td>
<td>n</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trill</td>
<td>B</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tap or Flap</td>
<td></td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plosive</td>
<td></td>
<td>t</td>
<td>d, n</td>
<td></td>
<td>ñ</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lateral approximant</td>
<td></td>
<td>ɾ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approximant</td>
<td></td>
<td>ɾ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CONSONANTS (NON-PULMONIC)

<table>
<thead>
<tr>
<th>Other</th>
<th>Voiced imclusives</th>
<th>Ejectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clichs</td>
<td>Bilabial</td>
<td>Bilabial</td>
</tr>
<tr>
<td>dental</td>
<td>Dental-alveolar</td>
<td>dental-alveolar</td>
</tr>
<tr>
<td>approximants</td>
<td>Palatal</td>
<td>Palatal</td>
</tr>
<tr>
<td>approximants</td>
<td>Palatal-alveolar</td>
<td>Palatal-alveolar</td>
</tr>
</tbody>
</table>

### VOICELESS VOWELS

- Long
- Short

### TONES & ACCENTS

- Level
- Rising
- Falling
- Contour

### DIACRITICS

- Voiced
- Voiceless
- Apical
- Labialized
- Nasalized
- Nasal release
- Advanced
- Retracted
- Pharyngealized
- No audible release
- Centralized
- Raised
- Lowered
- Syllabic
- Non-syllabic
- Advanced Tongue Root
- Retracted Tongue Root
Phonetic scripts for NLP

- IPA and variants (e.g. BIPA)
- NAPA
- ARPAbet
- SAMPA
- MBROLA
- eSpeak
- ASJP: 41 sound classes
- Nice reference here

![Table of IPA and ARPAbet symbols](Figure 7.1)

Figure 7.1: ARPAbet symbols for transcription of English consonants, with IPA equivalents. Note that some rarer symbols like the flap [dx], nasal flap [nx], glottal stop [q] and the syllabic consonants, are used mainly for narrow transcriptions.
Phonology

• The scientific study of what changes happen when sounds combine
• Often involves morphology, hence morphophonology
• Every language has many rules; not obvious to native speakers
• Sample English phonological rules:
  • CiV-Lengthening
    • Canada/Canadian, felon/felonious, comedy/comedian
  • Diphthongization (Caucasus/Caucasian)
  • Velar Softening
    • Public/publicity, prodigal/prodigy, medic/medicine
  • Palatization
    • Race/racial, diffuse/diffusion, enclose/enclosure
  • Spirantization
    • Explode/explosion, president/presidency

• ghoughphtheightteeau
Computational phonetics/phonology

- Phonological aspects of speech storage and manipulation by computer
- Modeling of speech-specific properties of language(s)
- Systems that assist in capturing, annotating, analyzing, exploiting speech data
- Systems that implement (aspects of) phonological theory

- Very employable skill in industry, government, academia
  - Especially for people with both programming and linguistics background
  - Specifying and developing systems
  - Testing/evaluating systems
  - Speech corpus collection, annotation, mining
Phonetic/phonological rules

• These rules interact with each other in (often) very complicated ways; dependencies are ordered, interdependent, even cyclic
• Accounting for these rule sequences is a large part of phonological investigation
• Various theories attempt to describe these interactions as succinctly as possible
• Native speakers have little/no explicit knowledge
Blackfoot phonological rules

(Frantz, 1991)

1. Gemination: \( C \rightarrow C_C / _-_ + C_A \)
2. s-Insertion: \( \phi \rightarrow s / _{-}l \)
3. s-Connection: \( \phi \rightarrow s / G + _-s \) and \( \phi \rightarrow i / V(i) + _-s \)
4. o-Replacement: \( o \rightarrow a / _- + a \)
5. Coalescence: \( w(i) \rightarrow o \)
6. Breaking: \( k \rightarrow ke / _{-}l \)
7. Neutralization: \( i \rightarrow i \)
8. Desyllabification: \( \{ i \rightarrow y, o \rightarrow w \} / V + _- V \) where \( i \) and \( o \) are unaccented
9. Semivowel drop: \( C \rightarrow \phi / _- # \)
10. Vowel shortening: \( V_j \rightarrow V; / _- + V \)
11. i-Loss: \( i \rightarrow \phi / V y _- \{a, o\} \)
12. i-Absorption: \( i \rightarrow \phi / s _- \{a, o\} \)
13. ih-Loss: \( ih \rightarrow \phi / s _- s \)
14. Presibilization: \( ths \rightarrow as, ths \rightarrow ias \)
15. Semivowel loss: \( C \rightarrow \phi / _- C \) where \( C \neq i. \)
16. y-Reduction: \( iy \rightarrow ii / C _- y \)
17. Postibilization: \( lh \rightarrow s / e _- s \)
18. t-Affrication: \( l \rightarrow ts / _- l \)
19. Glottal metathesis: \( 'V \rightarrow V' / V _- C \)
20. Glottal loss: \( ' \rightarrow \phi / VV: _- C \) where \( V \) is underlyingly long.
21. Glottal assimilation: \( V V \rightarrow V; / _- (s) C \)
22. Glottal reduction: \( ' \rightarrow \phi / _- ' \)
23. Vowel epanthesis: \( \phi \rightarrow V; / V: _- h \)
24. ssa-Shortening: \( ass \rightarrow as / _-_ C \)
25. Accent spread:
   \[ V \rightarrow [+\text{accent}] / \left[ \begin{array}{c}
   V \\
   [+\text{accent}]
   \end{array} \right] _- \]
Languages and writing

• Orthography: writing system(s) used by language(s)
  • Alphabetic: (roughly) one symbol / sound (e.g. English, Armenian)
  • Syllabic: (usually) one symbol / syllable (native Japanese, Cherokee)
  • Abugidic (alphasyllabic): CV* (Inuktutitut, Thai, Tibetan, etc.)
  • Logographic: (roughly) one symbol / word (Egyptian, Cuneiform, Chinese)
  • Abjadic: alphabetic, but omit (some/all) vowels (Arabic, Hebrew, Persian)

• Grapheme: individual symbol or character

• Wide variation in how closely the orthography reflects pronunciation
  • Deep orthography: wide difference between written and spoken language:
    tough / though / through / bough / cough / thorough / hiccough; ghoughpteightbeau
    (English=0.83, French=0.46)
  • Shallow orthography: little difference
    (Finnish=0.0, Hungarian=0.17)
NLP: often uses both phonetics and orthography

• We need to convert between the two levels of representation
  • Called phoneticization or grapheme-to-phoneme conversion (g2p)

• One solution: new resources
  • Standardize all sounds in all languages into one representation (e.g. phonetic alphabets for NLP)
  • Develop tools to automatically perform dictionary look-up to generate transcriptions
  • Problem: out-of-vocabulary (OOV) words

• Another solution
  • Develop tools that can automatically convert between representations and generalize
    • Knowledge-based
    • Machine-learned
CMUdict

- Used in a wide array of NLP tools for English
- Multiple forms for many words
  
  representative (R EH2 P R AH0 Z EH1 N T AH0 T IH0 V)
  representative(2) (R EH2 P R IH0 Z EH1 N T AH0 T IH0 V)
  representative(3) (R EH2 P R AH0 Z EH1 N AH0 T IH0 V)
  representative(4) (R EH2 P R IH0 Z EH1 N AH0 T IH0 V)

- Stress: 0=none, 1=primary, 2=secondary
- Open-source, downloadable, online lookup
- Bundled in NLTK
CELEX

• Tons of information on hundreds of thousands of words (English, German, Dutch)
  • Orthography, phonology (e.g. syllable structure), morphology (structure and frequency), syntax
• Subscription only, but we have it

• 5760\basketball\43\3253\1\P\ 'b#-skIt-b$1\ [CVV][CCVC][CVVC]\ [bA:\][skIt][bO:l]
Different approaches to g2p (grapheme-to-phoneme)

- Dictionary lookup accessing hand-curated resources
  - Look up “dog” in a database/wordlist:
    - CMUdict: ð ð ð
    - CELEX:
      - OED: Brit. /dɒɡ/, U.S. /dɔɡ/, /dɑɡ/
  - +: very fast
  - -: a difficult, time-consuming, never-ending task: OOV words, disfluencies, etc.

- Finite-state transducers
  - +: very fast, generalizable
  - -: need some hand coding, training resources

- Machine learning (including neural)
  - +: generalizable
  - -: needs vast training resources (especially neural), can be very slow
Two Japanese rules (simplified)

; Voicing, t:d
; LR: Sin+ta yom+ta yob+ta
; SR: Sin0da yon0da yon0da

RULE t:d <=> M:@ (+:0) _

Classes:
M: m,n
B: b,m

; Nasalization, B:n
; LR: yom+ta yob+ta
; SR: yon0da yon0da

; @:d due to t:d rule
RULE B:n <=> _ (+:0) @:d
Finite-state transducers

• The finite-state approach we discussed in the last lecture: two-level phonology
  • Starting point is underlying phonological form (UR)
  • Define rule application(s) that mediate between UR and SR
  • Compile the rules (and letters/words) into a FST that directs execution
  • Output is surface orthographical form (SR)

• Modes: generation/production and recognition/parsing

\[ /l \rightarrow [dx] / \hat{V} \_ V \]

Figure 11.1 Transducer for English Flapping: ARPA\textsuperscript{b}et “dx” indicates a flap, and the “other” symbol means “any feasible pair not used elsewhere in the transducer”. “@” means “any symbol not used elsewhere on any arc”. 
Finite-state transducer

- Special type of FSA
- Each state transition: input/output pair
- Allows for changing input sequence into output sequence
  - E.g. orthography ↔ IPA
Machine learning

• Obtain a large number of g2p correspondences
  • Corpora transcribed at the phoneme level
• Use them to train a computer to recognize the best alignments
  • Several possible algorithms
• Present new input to the system and have it analyze and generate results
• Many different systems, toolkits
  • **Sequitur-g2p**: based on neural machine translation as described in [this paper](#)
• Levenshtein edit distance frequently employed
State of the art

- FST works pretty well; room for improvement
  - Still, about 25% error (phoneme-level)
- Neural not quite as accurate (yet)
- Best: hybrid systems
  - Combine neural + FST
- Varies across languages
- Specialized hardware (GPU) improves performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Word Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen [7]</td>
<td>24.7</td>
</tr>
<tr>
<td>Bisani and Ney [2]</td>
<td>24.5</td>
</tr>
<tr>
<td>Novak et al. [6]</td>
<td>24.4</td>
</tr>
<tr>
<td>Wu et al. [12]</td>
<td>23.4</td>
</tr>
<tr>
<td>5-gram FST</td>
<td>27.2</td>
</tr>
<tr>
<td>8-gram FST</td>
<td>26.5</td>
</tr>
<tr>
<td>Unidirectional LSTM with Full-delay</td>
<td>30.1</td>
</tr>
<tr>
<td>DBLSTM-CTC 128 Units</td>
<td>27.9</td>
</tr>
<tr>
<td>DBLSTM-CTC 512 Units</td>
<td>25.8</td>
</tr>
<tr>
<td>DBLSTM-CTC 512 + 5-gram FST</td>
<td>21.3</td>
</tr>
</tbody>
</table>

Table 2. Comparison of various G2P technologies.
Transcription more generally

• Rewriting content in another writing system
• Many NLP applications treat proper nouns
  • (CL)IR of text, spoken documents
  • Information extraction
  • i18n, l10n
  • OCR/digitization
  • Semantic Web annotation
  • Homeland security and DoD
  and, of course,
  • Family history research
A few conversion strategies

- Lexical lookup: hashes, tables, tries, etc.
  - Unending, expensive, ambiguity problems, proper noun variants, etc.
- Transcoding: (mostly) rote character-by-character symbol conversion
  - Bush → 布什 vs. 布希 vs. 布殊; Osama bin Laden: 10 Hanzi variants
- Transcription: close phonetic transcription (g2p)
  - Bush → bʊʃ
- Transliteration: rewrite symbols of source language in target alphabet
  - Source/target sounds don’t always align
    - 32 English spellings for Muammar Gaddafi
    - 6 Arabic spellings for Clinton
  - Sensitive to properties of target language (Yuschenko vs. Iouchtchenko)
  - Romanization chaos: scores of schemes
- Transduction: finite-state transducer
  - Translation: names are rendered non-literally, non-phonemically to/from logograph (sequence)
  - Great Salt Lake → 大鹽湖
- Machine learning (including neural)
Romanization challenges

<table>
<thead>
<tr>
<th>Arabic Name</th>
<th>Romanization</th>
<th>Persian Name</th>
<th>Romanization</th>
</tr>
</thead>
<tbody>
<tr>
<td>شهید بیشته</td>
<td>Shaheed Baishtee</td>
<td>زهراء</td>
<td>Zahra</td>
</tr>
<tr>
<td>فاطمی</td>
<td>Fatimee</td>
<td>ولی</td>
<td>Wali A...</td>
</tr>
<tr>
<td>کوی علی</td>
<td>Koy Halawee</td>
<td>آستارکی</td>
<td>Aastarkee</td>
</tr>
<tr>
<td>شهر صنعتی</td>
<td>Sharai Sanati</td>
<td>علی محمد</td>
<td>Ali Mohmmad</td>
</tr>
<tr>
<td>دانشگاه</td>
<td>Daanishga</td>
<td>رضا</td>
<td>Reza</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Romanization</th>
<th>Persian Name</th>
<th>Romanization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hafizee</td>
<td>حافظ</td>
<td>450.000000 Hafizee</td>
</tr>
<tr>
<td>Hafeezee</td>
<td>حافظ</td>
<td>450.000000 Hafeezee</td>
</tr>
<tr>
<td>Jamsheed</td>
<td>جمشید</td>
<td>399.414000 Jamsheed</td>
</tr>
<tr>
<td>Jamshid</td>
<td>جمشید</td>
<td>396.716000 Jamshid</td>
</tr>
<tr>
<td>Jamshaid</td>
<td>جمشید</td>
<td>394.940000 Jamshaid</td>
</tr>
<tr>
<td>Jamasheed</td>
<td>جامشید</td>
<td>384.322000 Jamasheed</td>
</tr>
<tr>
<td>Shaahpur</td>
<td>شاهپور</td>
<td>450.164000 Shaahpur</td>
</tr>
<tr>
<td>Shaah+Pur</td>
<td>شاهپور</td>
<td>395.169000 Shaah+Pur</td>
</tr>
<tr>
<td>Behnaam</td>
<td>بهنام</td>
<td>436.044000 Behnaam</td>
</tr>
<tr>
<td>Behnaam</td>
<td>بهنام</td>
<td>402.424000 Behnaam</td>
</tr>
</tbody>
</table>
Tools

- foma
- Helsinki Finite-State Technology (HFST)
- Xerox Finite-State Morphology tool (XFST)
- Kleene
- OpenFST
- OpenGrm Thrax
- Morfessor
- SFST
- PC-Kimmo
- Phonetisaurus
- NLTK
- OpenGrm Pynini
- Lots of others!
Task #5

• Just like #4, except for phonology (not morphology)
  • Same tools, plus a few others possible
  • Recommended: use a different tool than you did for #4
• Build an engine that can perform some (morpho)phonologically interesting task
• If you did a mostly phonological task for #4 (because you didn’t know the difference yet), you can do a morphological one for this task.