LING 581: Advanced Computational Linguistics

Lecture Notes

April 2nd
Non-programmers can take more time on this assignment...

- you can combine my sample breadth-first search Perl code, e.g. `bfs3.perl`, and try to compute connections between words and meanings and store them into a table. Then you want to pair them up according to some objective function
  - Modify `bfs3.perl`:
    1. map (raw) words into word#pos#sense *(code snippet was given)*
    2. call the search for each word and meaning, store length (and maybe path as well) into a table
    3. write code to go through the table and output "best" matches

If your programming skills aren't quite up to the task, you may submit the table (step 3) and do the best matches by hand
# Answers

## Quiz 5:

<table>
<thead>
<tr>
<th>Word</th>
<th>Definition</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>arcane</td>
<td>outdated</td>
<td>a.</td>
</tr>
<tr>
<td>arduous</td>
<td>mysterious</td>
<td>b.</td>
</tr>
<tr>
<td>arabesque</td>
<td>hold one's attention</td>
<td>c.</td>
</tr>
<tr>
<td>asperity</td>
<td>impudent</td>
<td>d.</td>
</tr>
<tr>
<td>ascetic</td>
<td>strenuous</td>
<td>e.</td>
</tr>
<tr>
<td>arrant</td>
<td>complex design</td>
<td>f.</td>
</tr>
<tr>
<td>artless</td>
<td>austere</td>
<td>g.</td>
</tr>
<tr>
<td>archaic</td>
<td>harshness</td>
<td>h.</td>
</tr>
<tr>
<td>arrest</td>
<td>natural</td>
<td>i.</td>
</tr>
</tbody>
</table>

1. arcane  
2. arduous 
3. arabesque
4. asperity
5. ascetic
6. arrant
7. artless
8. archaic
9. arrest

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>b</td>
</tr>
<tr>
<td>2</td>
<td>e</td>
</tr>
<tr>
<td>3</td>
<td>f</td>
</tr>
<tr>
<td>4</td>
<td>h</td>
</tr>
<tr>
<td>5</td>
<td>g</td>
</tr>
<tr>
<td>6</td>
<td>d</td>
</tr>
<tr>
<td>7</td>
<td>i</td>
</tr>
<tr>
<td>8</td>
<td>a</td>
</tr>
<tr>
<td>9</td>
<td>c</td>
</tr>
</tbody>
</table>
## Answers

### Quiz 8:

<table>
<thead>
<tr>
<th></th>
<th>Word</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>bellicose</td>
<td>inclination</td>
</tr>
<tr>
<td>2</td>
<td>bent</td>
<td>misrepresent</td>
</tr>
<tr>
<td>3</td>
<td>blandish</td>
<td>carefree</td>
</tr>
<tr>
<td>4</td>
<td>bolster</td>
<td>belligerent</td>
</tr>
<tr>
<td>5</td>
<td>boisterous</td>
<td>pompous</td>
</tr>
<tr>
<td>6</td>
<td>blithe</td>
<td>support</td>
</tr>
<tr>
<td>7</td>
<td>belie</td>
<td>fawn</td>
</tr>
<tr>
<td>8</td>
<td>bedizen</td>
<td>loud</td>
</tr>
<tr>
<td>9</td>
<td>bombastic</td>
<td>adorn</td>
</tr>
</tbody>
</table>

1. d  2. a  3. g  4. f  5. h  6. c  7. b  8. i  9. e
# Answers

- **Quiz 16:**

<table>
<thead>
<tr>
<th></th>
<th>Word</th>
<th>Definition</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>discordant</td>
<td>harsh denunciation</td>
<td>a.</td>
</tr>
<tr>
<td>2</td>
<td>discomfit</td>
<td>intended to teach</td>
<td>b.</td>
</tr>
<tr>
<td>3</td>
<td>disapuse</td>
<td>tool used for shaping</td>
<td>c.</td>
</tr>
<tr>
<td>4</td>
<td>din</td>
<td>shy</td>
<td>d.</td>
</tr>
<tr>
<td>5</td>
<td>dilettante</td>
<td>stray from the point</td>
<td>e.</td>
</tr>
<tr>
<td>6</td>
<td>dilatory</td>
<td>causing delay</td>
<td>f.</td>
</tr>
<tr>
<td>7</td>
<td>digress</td>
<td>amateur</td>
<td>g.</td>
</tr>
<tr>
<td>8</td>
<td>diffident</td>
<td>loud noise</td>
<td>h.</td>
</tr>
<tr>
<td>9</td>
<td>die</td>
<td>undeceive</td>
<td>i.</td>
</tr>
<tr>
<td>10</td>
<td>didactic</td>
<td>frustrate</td>
<td>j.</td>
</tr>
<tr>
<td>11</td>
<td>diatribe</td>
<td>conflicting</td>
<td>k.</td>
</tr>
</tbody>
</table>
Digression

• Generative capacity of (natural) language ...
Chomsky Hierarchy

- Formal language theory
  [Chomsky, 1956]

  - Recursively enumerable (r.e.) languages
    - Context-sensitive languages
      - Context-free languages
        - Finite state languages
Chomsky Hierarchy

• grammar vs. machine characterization
  – $A \rightarrow aB$, $A \rightarrow a$
  – equivalent to a directed finite network of states and transitions

S → aB
B → bC
C → cD
D → dE
E → eF
F → fG
G → g
G → aH
H → bC
F → aI
I → bC

Birdsong: *Bengalese finch song* [Berwick et al., 2011]
Some useful notions

• Mathematical notion of a language as a set of strings
  \[ L(\text{Bengalese finch song}) = \{\text{abcdefg, abcdeabcdefg, abcdefgabcdefg, ..} \} \]

• Discrete infinity
  – “A fsm is the simplest type of grammar which, with a finite amount of apparatus, can generate an infinite number of sentences.”

Birdsong: Bengalese finch song [Berwick et al., 2011]
Discussion

• Finite state machine simply encodes **precedence** relations between elements of the string

• The (regular) grammar characterization adds in (possibly unintended) **hierarchical** relations between elements

---

S → aB
B → bC
C → cD
D → dE
E → eF
F → fG
G → g
G → aH
H → bC
F → al
I → bC

Birdsong: *Bengalese finch song* [Berwick et al., 2011]
Chomsky Hierarchy

• grammar vs. machine characterization
  – $A \to aB$, $A \to a$
  – equivalent to a directed finite network of states and transitions
• English is **not** a finite state language [(9) pg. 21, Chomsky, 1955]
• (11)
  (i) **If** $S_1$, **then** $S_2$
     *If $S_1$, or $S_2$
  (ii) **Either** $S_3$, **or** $S_4$
     *Either $S_3$, **then** $S_4$

“there are processes of sentence formation that fsm are intrinsically unable to handle.” [pg23, Chomsky, 1955]
Discussion (contd.)

• Grammatical construction
  – If .., then
  – Either .., or
  – “A grammatical construction is a syntactic template that is paired with conventionalized semantic and pragmatic content” [Wikipedia]

In some modern linguistic theories, constructions have little or no distinguished theoretical status
Chomsky Hierarchy

- English is not a context-free language [Higginbotham, 1984]
  - the woman such that (the man such that the man such that) she (gave this to him gave him to this) left is here

Construction: relative clause + extra dependencies
Discussion (contd.)

- “if a grammar does not have recursive devices (e.g. loops in fsm) it will be prohibitively complex. If it does have recursive devices of some sort, it will produced infinitely many sentences.” [pg24, Chomsky, 1995]

• Suppose human language is generated by some form of a recursive device:
  - human language admits infinitely many sentences
  - sentences can be arbitrarily long in principle
    (practical limitations on utterance length, working memory etc.)

Commonsense notion of grammaticality becomes a bit fuzzy
Chomsky Hierarchy (Fine Grained)

• Crossing dependencies (found in Dutch and Swiss-German)
  ... dat Wim Jan Marie de kinderen zag helpen leren zwemmen
  ... that Wim Jan Marie the children saw help teach swim

  ‘... that Wim saw Jan help Marie teach the children to swim’

(figures taken from [Kallmeyer, 2005])
Tree-adjoining Grammar (TAG) [Joshi]
Linear Indexed Grammar [Gazdar]
Combinatory Categorial Grammar [Steedman]
Chomsky Hierarchy (Fine Grained)

- **Indexed grammars**
  - Context-free languages
    - if-S-then-S
    - either-S-or-S
  - Finite state languages
    - Birdsong
  - Mildly context-sensitive languages
    - (linear indexed grammars)
  - Context-sensitive languages

**Crossing dependencies**

**Computation advantage:** polynomial time parsable
Primate language

- Northern muriquis
- Group studied extensively since 1982
- Data collected
  - 212 sequential exchanges vocalizations, emitted by several members:
    - 10 adult males (n=80),
    - 1 sub adult male (n=1),
    - 17 adult females (n=122), and
    - 2 sub adult females (n=3)
  - **sequential exchange** = vocalization of an individual and the response of different group members
  - Muriqui vocalizations have a quite long duration allowing to analyze them as sequences of different elements that are repeated or varied over time
  - It is possible to identify subparts in vocalizations that we designate by the term ‘segmental’ units
Primate language

- Sample utterance
Primate language

- Sequence:
  - tp tpp .. tpp..p Gt
  (monotonically increasing # p's)

String containing an arithmetic progression

Indexed grammars

Context-free languages

Finite state languages

Mildly context-sensitive languages (linear indexed grammars)

Crossing dependencies

Birdsong

If-S-then-S
Either-S-or-S

Arithmetic progression sequences
Primate language

• Sequence:
  – \( tp \; tpp \; .. \; tpp..\; p \; Gt \)
    (arithmetic progression)

• Indexed grammar [Aho, 1967]
  – equivalent to nested stack automaton
    – \( S[..] \rightarrow K[p..] \; Gt \)
    – \( K[..] \rightarrow tP[..] \; K[p..] \)
    – \( K[..] \rightarrow tP[..] \)
    – \( P[p..] \rightarrow pP[..] \)
    – \( P[\] \rightarrow [\] \)

Example derivation:
\[
S[\] \rightarrow K[p] \; Gt \rightarrow tP[p] \; K[pp] \; Gt \\
\rightarrow tp \; K[pp] \; Gt \rightarrow tp \; tP[pp] \; K[ppp] \; Gt \rightarrow \]
\[
tp \; tpp \; tP[ppp] \; K[pppp] \; Gt \rightarrow \]
\[
tp \; tpp \; tppp \; tP[pppp] \; Gt \rightarrow \]
\[
tp \; tpp \; tppp \; tpppp \; Gt \]
Primate language

- Sequence:
  - tp tpp .. tpp..p Gt
    (arithmetic progression)

Intuition:
indexed grammar with binary branching
1\textsuperscript{st} nonterminal generates \( k \)
p’s, and
2\textsuperscript{nd} nonterminal recursively generates sequences involving \( k+1 \) p’s

- Rely on a crucial distinction between \textbf{Indexed grammars (IG)} and the strictly smaller class of \textbf{Linear indexed grammars (LIG)}
  - at most one nonterminal in each production be specified as receiving the stack, whereas in a normal indexed grammar, all nonterminals receive copies of the stack

- [Vijay-Shanker and Weir, 1994] demonstrated that LIG, CCG, TAG are all (weakly) equivalent formalisms = “Mildly Context-Sensitive Languages”
Indexed grammars

Context-sensitive languages

Context-free languages

Finite state languages

Mildly context-sensitive languages (linear indexed grammars)

if-S-then-S

either-S-or-S

Birdsong

Computational disadvantage: not polynomial time parsable

Arithmetic progression sequences

Crossing dependencies
Discussion (contd.)

• Possible objection
  – did I pick out a hard subset of an easy language?

for example, a subset of a finite state language is not necessarily a finite state language
Discussion (contd.)

- Data

what counts as a “construction” here?
Discussion (contd.)

• Weak generative capacity:
  – rely on surface string patterning only


\[
\begin{align*}
S[..] &\rightarrow K[p..] \text{ Gt} \\
K[..] &\rightarrow tP[..] \text{ K[p..]} \quad \text{non-terminal} \\
K[..] &\rightarrow tP[..] \quad \text{and terminal} \\
P[p..] &\rightarrow pP[..] \quad \text{symbols} \\
P[] &\rightarrow []
\end{align*}
\]

• No intuition about the “constituents” of muriqui language
Finite state languages

Context-free languages

Mildly context-sensitive languages (linear indexed grammars)

Indexed grammars

if-S-then-S

either-S-or-S

Context-sensitive languages

Human Languages