Tools and applications I

Morphology, syntax, semantics, pragmatics
Tokenization

- Identify words
- Identify sentences, abbreviations
- Identify symbols (numbers, addresses, markup codes, special characters)
- Normalize orthography (spelling, caps, hyphenation, etc.)
ABERNETHY, WILLIAM, Wallingford, m. 1673 or 4, Sarah, d. of William Doolittle, had William, and Samuel, and d. 1718, when his two s. admin. on his est. Early this name was writ. Ebenetha, or Abbenatha, acc. Hinman; but in mod. days the descend. use the spell. here giv.

ABINGTON, WILLIAM, Maine, 1642. Coffin.

ABORNE. See Eborne.

ACRERLY, ACCORLEY, or ACRELY, HENRY, New Haven 1640, Stamford 1641 to 53, Greenwich 1656, d. at S. 17 June 1668, wh. is the date of his will. His wid. Ann, was 75 yrs. old in 1662. Haz. II. 246.

ROBERT, Brookhaven, L. I. 1655, adm. freem. of Conn. jurisdict. 1664. See Trumbull, Col. Rec. I. 341, 428. SAMUEL, Brookhaven, 1655, perhaps br. of the preced.
Taggers

- Process source text, mark for part-of-speech
- Different approaches
  - Statistical modeling
  - Rules
  - Analogical modeling
- Sample output: Portuguese tagged text
Computational morphology

- **Applications**
  - Search engines (web, corpora)
  - Speech recognition, generation
  - Text understanding (parsing)

- **Approaches**
  - Exhaustive listing (inflected lexicon)
  - Cut-and-paste
    - Ad-hoc, limited usefulness (fair for Engl.)
  - Finite-state techniques
Morphology engine output (1)

PC-KIMMO>recognize ducks

`duck+s  `duck+PL

1:

<table>
<thead>
<tr>
<th>Stem</th>
<th>INFL</th>
</tr>
</thead>
<tbody>
<tr>
<td>`duck</td>
<td>+s</td>
</tr>
</tbody>
</table>

ROOT +PL

`duck

Word:

[ cat: Word
  head: [ agr:
    number:PL
    pos: N ]

root: `duck
root_pos:N
clitic:-
drvstem:- ]

1 parse found

`duck+s  `duck+3SG

1:

Word

<table>
<thead>
<tr>
<th>Stem</th>
<th>INFL</th>
</tr>
</thead>
<tbody>
<tr>
<td>`duck</td>
<td>+s</td>
</tr>
</tbody>
</table>

ROOT +3SG

`duck

Word:

[ cat: Word
  head: [ agr:
    3sg: + ]

finite:+
pos: V
tense: PRES
vform: S ]

root: `duck
root_pos:V
clitic:-
drvstem:- ]

1 parse found
Morphology engine output (2)

PC-KIMMO> recognize supercooled

super+`cool+ed     DEG9/LOC2+`cool+ED

3 parses found:

Word

<table>
<thead>
<tr>
<th>Stem</th>
<th>INFL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+ed</td>
</tr>
</tbody>
</table>

PREFIX | Stem | +ED
super+ |     | `cool
DEG9/LOC2+ | ROOT | `cool

Word:

[ cat:   Word
  head:   [ finite:+
            pos:    V
            tense: PAST
            vform: ED ]
  root:   `cool
  root_pos:V
  clitic:-
  drvstem:- ]
**Morphology engine output (3)**

```
1:

    Word_1
      _____|______
         Stem_2      INFL_6+
               |     +ed
      PREFIX_3+ Stem_4+  +EN
    super+         |
      DEG9/LOC2+    Word_10
                      ___|____
                         Stem_4+ INFL_6+
    super+         |
      DEG9/LOC2+    ROOT_5+
                      `cool
                         `cool

Word:
[ cat: Word
  head: [ finite:-
     pos: V
     vform: EN ]
  root: `cool
  root_pos: V
  clitic:-
  drvstem:- ]

table
<table>
<thead>
<tr>
<th>Word</th>
<th>Head</th>
<th>Cat</th>
<th>Aform</th>
<th>Pos</th>
<th>Verbal</th>
</tr>
</thead>
</table>
| super+`cool+ed | [ finite:-
  stem: `cool  | V | ABS | AJ | + ]
| DEG9/LOC2+`cool+EN | |

2:

    Word_7
      _____|______
         Stem_8
               |     +ed
      PREFIX_3+ Stem_9
    super+         |
      DEG9/LOC2+    Word_10
                      ___|____
                         Stem_4+ INFL_6+
    super+         |
      DEG9/LOC2+    ROOT_5+
                      `cool
                         `cool

Word:
[ cat: Word
  head: [ aform: ABS
     pos: AJ
     verbal:+ ]
  root: `cool
  root_pos: V
  clitic:-
  drvstem:- ]
```
Armenian word graph

Word
| NDet
    | NDecl
        | NBase
            | ROOT
                | tjpax'dowt'iwn
woe_tribulation

| ART
  +s
| CASE
  +1sPoss.
  +ov
  +Inst
| PLURAL
  +ny'r
  +plural
;;; Genitive epenthesis rule

;;; #Fransa0+i#    #T’oxio’0+i#

;;; #Fransah’i#    #T’oxioh’0i#

RULE 0:h' <= [a|o':o] __ +:0 i

<table>
<thead>
<tr>
<th></th>
<th>o'</th>
<th>i</th>
<th>a</th>
<th>@</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>o'</td>
<td>i</td>
<td>a</td>
<td>@</td>
</tr>
</tbody>
</table>

h' o 0 i a @

1: 0 2 1 1 2 1
2: 4 2 3 1 2 1
3: 0 2 1 0 2 1
4: 0 0 5 0 0 0
5: 0 0 0 1 0 0
Farsi morphology

PC-KIMMO> recognize nmibinmC
n+mi+bin+m+C      NEG+DUR+see.PRES+1S+3s.object

Top
  |
Verb
    __________|__________
   |        |        
VNEGPREFIX       VNStem
   n+      |      
   NEG+    |        
   VPREFIX |        
   mi+     |        
   DUR+    |        
   V1Stem  |        
   VOSUFFIX | +C
   V2Stem  |        
   VPSUFFIX +3s.object
   | +m
   V3Stem  |        
   +1S
   |
   V
   bin
   see.PRES
Optional syncope rule
Note: free variation
L: Lu+ad+s+pastEd
S: L00ad0s0pastEd
RULE
"u:0 => [L|T'] ___ +:@ VW" 4
6

u  L  +  VW @  T'
0  L  @  VW @  T'
1:  0  2  1  1  1  2
2:  3  2  1  1  1  2
3.  1  0  4  0  0  0
4.  1  0  0  1  0  0
Syntactic parsing

- Build syntax for phrases, sentences
- Constructs categories, constituents, trees
- Phrase-structure grammar rules
- Top-down vs. bottom-up
- Chart: collect all possibilities
- Related to compiler design, implementation
- Grammar engineering
Parsing

- Start with text (e.g. sentence)
- Label each of the elements (e.g. words)
- Diagram the relationships between elements
- Why?
  - Shows constituency
  - Visual representation of content
  - Useful for future reference (e.g. treebanks)
Different kinds of parse trees (ways of viewing a sentence)

(5) The noticeably overweight tourists ate grilled shark in a thick buttery sauce.
Linguistic theories and parsing

- LFG (KANT)
- GB/P&P (NL-Soar)
- SFG (NIGEL)
- HPSG (Verbmobil)
- Categorial grammar (ALE)
- RST (PENMAN)
- TAG (XTAG)
- STATISTICS (CANDIDE)
- etc. etc.
Parsers

- Take input sentences, perform morpho-syntactic/semantic analysis, output structural representations of content
- Many different syntactic theories \(\Rightarrow\) many different kinds of parsers
Parse tree

```
CP
  |
  C
  |
  CP
  |
  C'
  |
  IP
  |
  NP
  |
  det
  |
  the
  |
  N'
  |
  N
  |
  doctor
  |
  N'
  |
  CP
  |
  C'
  |
  IP
  |
  N
  |
  t_x
  |
  who
  |
  N'
  |
  C'
  |
  IP
  |
  I
  |
  t_x
  |
  I'
  |
  t_j
  |
  VP
  |
  V
  |
  V
  |
  past
  |
  called
  |
  V
  |
  V
  |
  PRESENT
  |
  works at
  |
  det
  |
  a
  |
  N'
  |
  N
  |
  hospital
```
Sample parses
I've been majoring in Material engineering at my University in Korea.

but probably the best class for me was medicine and first aid principles.
LG parsing (Arabic and Farsi)

و لم يقدم المصدر أي إيضاحات حول أهداف الزبارة

\[ \text{w lm y qdm Al m8dr y <yDAH At Hw1 >hdAf Al zyAr p} \]

سجد ١٠٢٠٣ـة ٢٠٠٦ـهـ

آنها ديروز ست نه كتابه ين شما زدند

\[ \text{w conj lm neg y.IV3M8 qdm.IV Al.d m8dr.Ndu y.quant <yDAH.NAt At.NFEMPL Hw1.PREP >hdAf.N Al.d zyAr.NapAt p.NFEMSG} \]

١٠٧٠٣١٠٠٣

\[ \text{jnhA.pn diruz.av dst.nk bh.pp ktAb.n hA.nms e.ez CmA.pn z.vk d.vmt nd.vmp} \]
Persian: **<tu midAni kh mn mirum>** “you know that I am going”
++++Time 0.02 seconds (0.30 total)
Found 1 linkage (1 had no P.P. violations)
Unique linkage, cost vector = (UNUSED=0 DIS=4 AND=0 LEN=24)

+----------------------------Xp---------------------------+
|   +-------------------SOo-------------------+   |
|   +------EX------+------P-----+             |   |
| +-----Wd----+---SOs--+     |  +----DT---+             |   |
|      +-PRF+-TX+    |     |  |   +--NZ-+      +--DT--+   |
|      |    |   |    |     |  |   |     |      |      |   |
Visualizing lexical relations

- WordNet (of course)
- The Visual Thesaurus
- Text clusterers
  - clusty.com
  - mooter.com
- The Lexical Freenet
IE and the Semantic Web

The search query

Java

island

an island in Indonesia S of Borneo; one of the world's most densely populated regions

coffee java beverage

a beverage consisting of an infusion of ground coffee beans; "he ordered a cup of coffee"

Java object-oriented programming language

a simple platform-independent object-oriented programming language used for writing applets that are downloaded from the World Wide Web by a client and run on the client's machine
IE and the Semantic Web

- Ranking based on content data and structure (XML,...)
- Using hierarchies for similarity search
- Grouping results by their topics: WSD is required!
Encoding pragmatics

- OpenMind
  - Enter data to represent commonsense real-world information (examples)
- Cycorp’s FACTory
- Never-Ending Language Learner (NELL)
- Lots of others...
Specify, manipulate dialogue/discourse turns

Manage model of total information state
- Private beliefs, plans, discourse agenda
- Shared knowledge: content, context, common ground
- Accommodation of goals, partial and out-of-sequence info

V-commerce, call center management, conversation tracking, intelligent tutorial dialogues
1-6: Eagle 6, this is 1-6. The situation here is growing more serious. We've spotted weapons in the crowd. Over.

Base: 1-6, this is Eagle 6. Eagle 2-6 is in the vicinity of Celic right now and enroute to your location.

1-6: Eagle 2-6, this is 1-6. I need your assistance here ASAP. Things are really starting to heat up here.

Lt: What happened?

Lt (2-6) to 1-6: They just shot out from the side streets, sir... Our driver couldn't see 'em coming.

Lt: How bad? Is he okay?

Medic: Driver's got a cracked rib, but the boy's—Sir, we gotta get a Medevac here ASAP.

Lt: Base, request Medevac.

Lt: Base: Standby. Eagle 2-6, this is Eagle base. Medevac launching from operating base Alicia. Time: Now. ETA your location 03. Over.

Sgt: Yes Sir! Squad leaders, listen up! I want 360 degree security here. First squad 12 to 4. Second squad 4 to 8. Third squad 8 to 12. Fourth squad, secure the accident site. Follow our standard procedure.
Dialogue application: training

- SOF mission rehearsal and cultural adaptation exercises
- Soar-based agents interact to create scenario
- 3D virtual reality including spoken language
  - Discourse modeling, conversational pragmatics, dialogue strategies
- Human trains on how to manage situation, build rapport
Computer programs (usually agents) that hold conversations with people
Most limited to specific domains (e.g. Hampy)
The Loebner contest
- “An average interrogator will not have more than a 70% chance of making the right identification after five minutes of questioning” (1950)
- 6 c-bots, 4 humans (teacher, MBA, minister, yogi), 10 judges (linguist, chemist, 2 phil’is, musician, psych’ist, journalist, u’grad, author, grad)
- 91% correct after 5 minutes, 93% after 15
- No computer was mistaken for a human.
A: I like techno, especially retro-electronic like Kraftwerk. I like 70's bands like ABBA. Also I like Opera.

J: ALICE LIKES ABBA

A: That's me. My favorite group is Kraftwerk.

J: FIGURES

A: No but I know a lot of female humans.

J: DO YOU SPEAK ANY SWEDISH

A: I speak only English right now, but I'd like to learn some other languages. Can you teach me a few phrases in Any Swedish.

J: SURE

A: OK let's do it. Tell me more.

J: SWEDISH FOR COMPUTOR IS DATOR

A: How do you usually introduce yourself?

J: WOMAN IS KVINNA SO I GUESS YOU ARE A KVINNLIG DATOR

A: What do you look like?

J: I LOOK TONS DIFFERENT FROM YOU

A: You sound very attractive.

J: HOW WOULD YOU DESCRIBE YOURSELF BY THE WAY

A: It is a good question. Let me think about it. I can ask someone about it.