Morphology

• Morphology studies the formation and transformation of words

• Words are formed from Morphemes

• Lexicon: an organized collection of words in a language
Morpheme

Smallest unit of language that carries information about meaning or function

know; know-ing; know-s; know-er;
  a-know-ledge;
  anti-dis-establish-ment-ari-an-ism
Lexeme (lemma)

• The minimal unit of language which
  – has a semantic interpretation and
  – embodies a distinct cultural concept

• A lexeme is conventionally listed in a dictionary as a separate entry

(www.sil.org)
Morphology

**Derivational**: creates new words (new lexemes)
compute > computer > computerize > computerization

**Inflectional (grammatical)**: creates different forms of a word for different persons, numbers, tenses, cases, modes ...

talk, talks, talked, talking
merg, mergi, mergem, mergeam
ferastra, ferestrele, fereastro, ferestre
Morphemes

- **Root** (\textit{un-know-able})
- **Affixes** (\textit{un-know-able})
  - prefixes - “un-”, “anti-”, “pre-”, etc. (\textit{un}-explicable, \textit{anti}-terrorist).
  - sufixes - “-able” (read-\textit{able}), “-er” (read-\textit{er}), etc.
  - infixes
- **May be**
  - Lexical
  - Gramatical for inflexion
Morphological analysis

• Identification of the root, affixes and and maybe the grammatical form

• “moved” ➞ “move” + “ed”

• books ➞ “book”+Noun+plural
  ➞ “book”+verb+present+3rd person+singular

• tries ➞ “try”+verb+present+3rd person+singular
Analysis of plurals

- CHURCHES \(\rightarrow\) CHURCH + ES
- SPOUSES \(\rightarrow\) SPOUSE + S
- FLIES \(\rightarrow\) FLY + IES
- PIES \(\rightarrow\) PIE + S
- GROOVES \(\rightarrow\) GROOVE + S

Exceptions:
- MICE \(\rightarrow\) MOUSE
- FISH \(\rightarrow\) FISH
- ROOVES \(\rightarrow\) ROOF + VES
- BOOK ENDS \(\rightarrow\) BOOK END + S
- LIEUTENANTS GENERAL \(\rightarrow\) LIEUTENANT (+S) GENERAL
Analysis of inflexions

- LODGING → LODGE + ING
- BANNED → BAN + NED
- FUMED → FUME + D
- BREACHED → BREACH + ED
- TAKEN → TAKE + N

Irregularities

- TAUGHT → TEACH
- FAUGHT → FIGHT
- TOOK → TAKE
Synthesis

“try” + verb + present + 3rd person + singular ➞ tries
Computational aspects

• Lexicon
• Formation rules
  – Morphotactics
  – Inflexion rules
  – Phonological rules
Finite automata
TRIE trees
Linear representation

- \((a(c\text{ subst sing neutru})\]
- \((t\text{ subst sing neutru})\]
- \((o(r(d\text{ subst sing neutru})\]
- \((a\text{ verb infinitiv}\]
- \((m\text{ verb imperfect prezent pers1 sg-pl}))\]
- \((d(u\text{ verb imperativ})\]
- \((c\text{ verb prezent pers3 pl}\]
- \((e\text{ verb infinitiv})\]
- \((b(a(d(e\text{ subst sing masc})\]
- \((s(u(i\text{ verb infinitiv}\]
- \((a\text{ verb imperfect prezent pers3 sg}\]
- \((m\text{ verb imperfect prezent pers1 sg-pl}))\]
- \((d(a\text{ verb infinitiv})).\]
- \((m\text{ verb imperfect prezent pers1 sg-pl}))))))\]
(a(c subst neutru
   (t subst neutru)
   (o(r(d subst sing neutru
      (a verb)))))
   (d(u(c(e verb)))))
(b(a(d(e subst masc)))))
(s(u(i verb
   (d(a verb)))))
Regular expressions (Xerox FST)

Prefix=[(un+) | (dis+) | (ln+)]
Root=[(tie) | (embark) | (happy) | (decent) | (fasten)]
Sufix=[(+s) | (+ing) | (+er) | (+ed)]
Lexicon=([(un+)] [(tie) | (fasten)] [(+s) | (+ing) | (+ed)] )
| [ ((in+)] [decent] )
| [([un+)] [happy] [(+er)] )
| [([dis+)] [embark] [(+s) | (+ing) | (+ed)] )

can generate: un+tie, dis+embark+ing ....
Xerox morphological categories (a fragment)

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Word</th>
<th>Examples</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>abbreviation</td>
<td>ea.</td>
<td>ea.</td>
<td>ea.+Quant+Abbr</td>
<td></td>
</tr>
<tr>
<td>acronym</td>
<td>USA</td>
<td>USA</td>
<td>USA+Prop+Misc+Acron</td>
<td></td>
</tr>
<tr>
<td>adjective</td>
<td>blue</td>
<td>blue</td>
<td>blue+Adj</td>
<td></td>
</tr>
<tr>
<td>adverb</td>
<td>today</td>
<td>today</td>
<td>today+Adv</td>
<td></td>
</tr>
<tr>
<td>auxiliary (verb)</td>
<td>will</td>
<td>will</td>
<td>will+Aux</td>
<td></td>
</tr>
<tr>
<td>business name</td>
<td>Xerox</td>
<td>Xerox</td>
<td>Xerox+Prop+Bus</td>
<td></td>
</tr>
<tr>
<td>cardinal (number)</td>
<td>ten</td>
<td>ten</td>
<td>ten+Num+Card</td>
<td></td>
</tr>
<tr>
<td>city name</td>
<td>London</td>
<td>London</td>
<td>London+Prop+Place+City</td>
<td></td>
</tr>
<tr>
<td>punctuation comma</td>
<td></td>
<td>.</td>
<td>+Punct+Comma</td>
<td></td>
</tr>
<tr>
<td>comparative</td>
<td>better</td>
<td>better</td>
<td>good+Adj+Comp</td>
<td></td>
</tr>
<tr>
<td>conjunction</td>
<td>because</td>
<td>because</td>
<td>because+Conj+Sub</td>
<td></td>
</tr>
<tr>
<td>continent name</td>
<td>Europe</td>
<td>Europe</td>
<td>Europe+Prop+Place+Continent</td>
<td></td>
</tr>
<tr>
<td>coordinating (conjunction)</td>
<td>or</td>
<td>or</td>
<td>or+ Conj+Coord</td>
<td></td>
</tr>
<tr>
<td>country name</td>
<td>Scotland</td>
<td>Scotland</td>
<td>Scotland+Prop+Place+Country</td>
<td></td>
</tr>
<tr>
<td>decimal (number)</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23+Dig+Dec</td>
<td></td>
</tr>
<tr>
<td>definite (determiner)</td>
<td>these</td>
<td>these</td>
<td>these+Det+Deg+Pi</td>
<td></td>
</tr>
<tr>
<td>(academic) degree</td>
<td>M.A.</td>
<td>M.A.</td>
<td>M.A.+Deg+Abbr</td>
<td></td>
</tr>
<tr>
<td>(determiner)</td>
<td>the</td>
<td>the</td>
<td>the+Det+Def+SP</td>
<td></td>
</tr>
<tr>
<td>digital number</td>
<td>123</td>
<td>123</td>
<td>123+Dig+Card</td>
<td></td>
</tr>
<tr>
<td>amount of dollars</td>
<td>$100</td>
<td>$100</td>
<td>$100+Dig+DlrAmt</td>
<td></td>
</tr>
</tbody>
</table>
Example

- This this+Adv
- This this+Det+Sg
- This this+Pron+NomObl+3P+Sg
- is be+Verb+Pres+3sg
- only only+Adj
- only only+Adv
- only only+Conj+Sub
- a a+Let
- a a+Det+Indef+Sg
- simple simple+Adj
- simple simple+Noun+Sg
- sample sample+Noun+Sg
- sample sample+Verb+Pres+Non3sg
- sentence sentence+Noun+Sg
- sentence sentence+Verb+Pres+Non3sg
Implementation of the formation rules

Finite state transducer

aaaabb

aaaεbb
move+ed move+ing seize+ure dye+ed tie+ing
moveεεed moveεεing seizeεεure dyeεεed tieεεing
Two levels morphology (Koskeniemi)

1) Context restrictioning rules
   \[ a:b \rightarrow CS \_ CD \]

2) Surface form restrictioning
   \[ a:b \leftarrow CS \_ CD \]

3) Composed rules
   \[ a:b \leftrightarrow CS \_ CD \]

4) Exclusion rules
   \[ a:b \leftrightarrow CS \_ CD \]
Stemmers

stem = word + inflection
Porter's Algorithm
(K.V. Lakshmi)

- The Porter Stemmer is a conflation Stemmer developed by Martin Porter at the University of Cambridge in 1980.
- Porter stemming algorithm (or 'Porter stemmer') is a process for removing the commoner morphological and inflexional endings from words in English.
- Most effective and widely used.
- Porter's Algorithm works based on number of vowel characters, which are followed be a consonant character in the stem (Measure), must be greater than one for the rule to be applied.
- A word can have any one of the forms: C……C, C…..V, V…..V, V…..C.
- These can be represented as [C](VC)\(^m\)[V].
Porter's Algorithm  contd..

- The rules in the Porter algorithm are separated into five distinct steps numbered from 1 to 5. They are applied to the words in the text starting from step 1 and moving on to step 5.
- Step 1 deals with plurals and past participles. The subsequent steps are much more straightforward.
  - Ex. plastered->plaster, motoring-> motor
- Step 2 deals with pattern matching on some common suffixes.
  - Ex. happy -> happi, relational -> relate, callousness ->  callous
- Step 3 deals with special word endings.
  - Ex. triplicate-> triplc, hopeful-> hope
Porter's Algorithm  contd..

- Step 4 checks the stripped word against more suffixes in case the word is compounded.
  Ex. revival -> reviv, allowance-> allow, inference-> infer etc.,

- Step 5 checks if the stripped word ends in a vowel and fixes it appropriately
  Ex. probate -> probat, cease -> ceas, controll -> control

The algorithm is careful not to remove a suffix when the stem is too short, the length of the stem being given by its measure, m. There is no linguistic basis for this approach.