Introduction to Morphology

Stefan Trausan-Matu

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

fereastra, ferestrele, fereastro, ferestre

Morphemes

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

Morphological analysis

- Identification of the root , afixes 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 → CHURCH + ES
- SPOUSES → SPOUSE + S
- FLIES → FLY + IES
- PIES → PIE + S
- GROOVES → GROOVE + S

Exceptions:

- MICE → MOUSE
- FISH → FISH
- ROOVES → ROOF + VES
- BOOK ENDS → BOOK END + S
- LIEUTENANTS GENERAL → 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

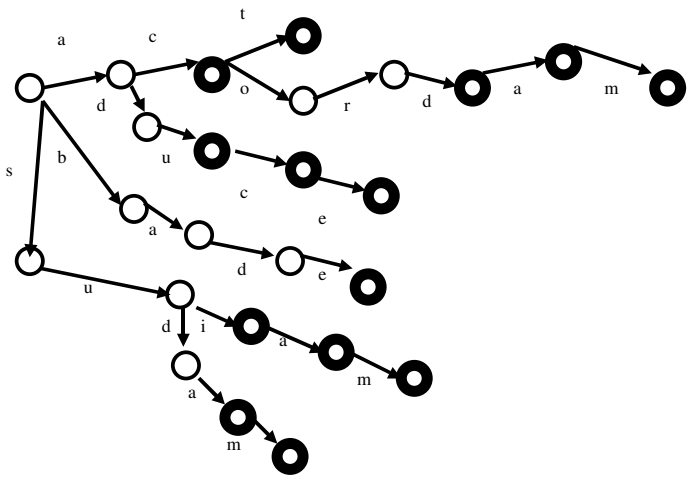
Synthesis

“try”+verb+present+ 3rd person +singular ➔ tries

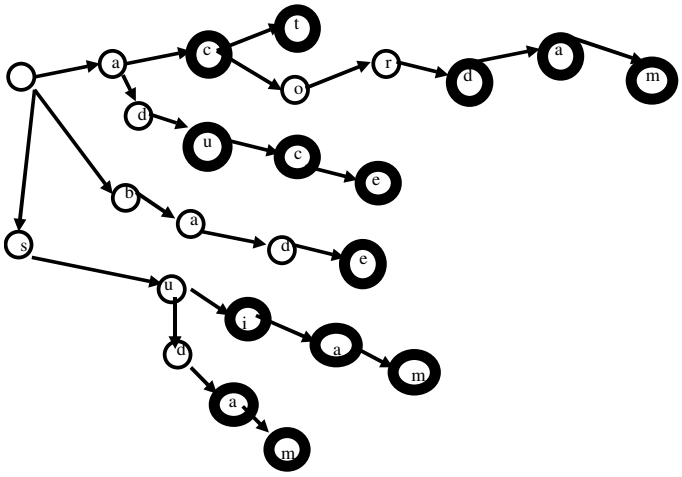
Computational aspects

- Lexicon
- Formation rules
  - Morphotactics
  - Inflexion rules
  - Phonological rules

Finite automata



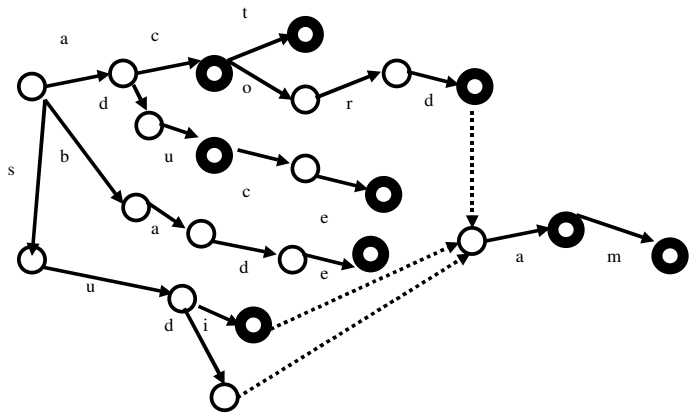
TRIE trees



Linear representation

- ((**a**(**c** subst sing neutru
- (**t** subst sing neutru)
- (**o**(**r**(**d** subst sing neutru
- (**a** verb infinitiv
- (**m** verb imperfect prezent pers1 sg-pl))))))
- (**d**(**u** verb imperativ
- (**c** verb prezent pers3 pl
- (**e** verb infinitiv)))))
- (**b**(**a**(**d**(**e** subst sing masc))))
- (**s**(**u**(**i** verb infinitiv
- (**a** verb imperfect prezent pers3 sg
- (**m** verb imperfect prezent pers1 sg-pl)))
- (**d**(**a** verb infinitiv
- (**m** 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))))
```


t
a
c
d
o
r
d
u
s
b
c
a
e
d
e
a
m
u
d
i

Regular expressions (Xerox FST)

Prefix=[[u n +] | [d i s +] | [l n +]]]
Root=[[t i e] | [e m b a r k] | [h a p p y] | [d e c e n t] | [f a s t e n]]
Sufix=[[+ s] | [+ I n g] | [+ e r] | [+ e d]]
Lexicon=[([u n +]) [[t i e] | [f a s t e n]] ([+ s] | [+ i n g] | [+ e d])]
| [([i n +]) [d e c e n t]]
| [([u n +]) [h a p p y]] ([+ e r])]
| [([d i s +]) [e m b a r k] ([[+ s] | [+ i n g] | [+ e d]])]
can generate: un+tie, dis+embark+ing ….

Xerox morphological cathegories (a fragment)

[redacted] major categories (part-of-speech)
minor categories (number, tense, subclass, ...)

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

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

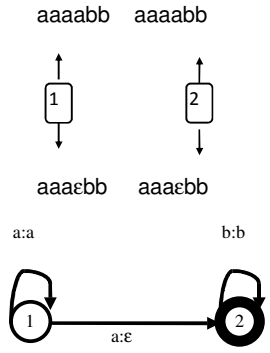
Implementation of the formation rules

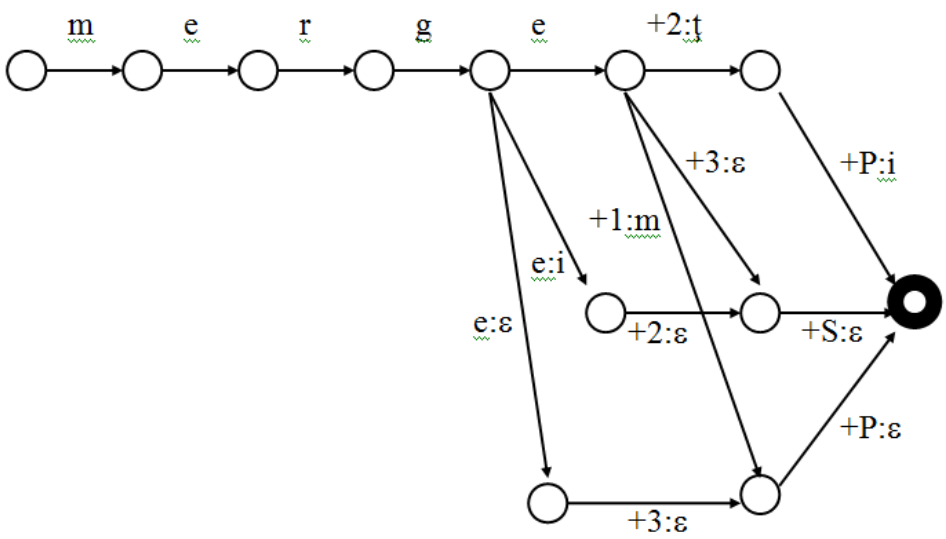
Finite state transducer

aaaabb

aaaɛbb

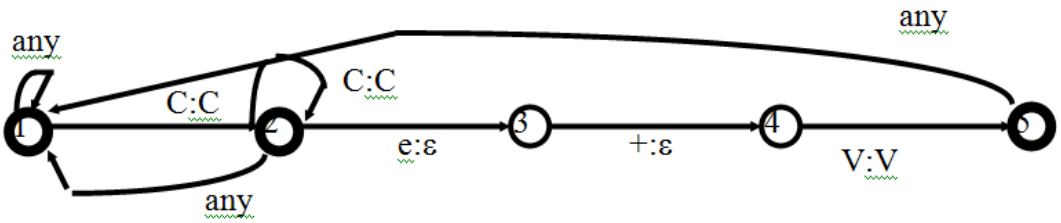
	a : a	A : ɛ	b : b
1	1	2	0
2	0	0	2




m
e
r
g
e
+2:t
+3:ε
+P:i
+1:m
e:i
e:ε
+2:ε
+S:ε
+P:ε
+3:ε

move+ed move+ing seize+ure dye+ed tie+ing

movεεed movεεing seizεεure dyεεed tyεεing



	C : C	V : V	e : ε	+ : ε	others
1	2				1
2	2		3		1
3				4	
4		5			
5					
6					1

Two levels morphology (Koskeniemi)

1) **Context restrictioning rules**

a:b ➔ CS _ CD

2) Surface form retrictioning

a:b ← CS _ CD

3) Composed rules

a:b ←➔ CS _ CD

4) Exclusion rules

a:b /← 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-> triplic, 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.