

Master in Artificial Intelligence

Morphology

Morphological
analysis

Spell checkers
and spell
correctors

Introduction to Human Language Technologies 3 - Morphology



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Outline

Morphology

Morphological
analysis

Spell checkers
and spell
correctors

- 1 Morphology
 - Motivation
 - Definitions
 - Types of morphology
- 2 Morphological analysis
 - Finite-state automata
 - Finite-state transducers
- 3 Spell checkers and spell correctors

Outline

Morphology

Motivation

Morphological
analysis

Spell checkers
and spell
correctors

- 1 Morphology
 - Motivation
 - Definitions
 - Types of morphology
- 2 Morphological analysis
 - Finite-state automata
 - Finite-state transducers
- 3 Spell checkers and spell correctors

Motivation

There are lots of NLP tools and applications in which dealing with the morphology of the words is relevant, for instance:

- IR is based on the canonical forms of the words.
 - 'Normally, **houses** in the **Pyrenees** are made of stone.'
 - 'A typical **pyrenean house** has little windows.'
- Spell checkers are based on checking whether words in a document are well-formed or not.
 - 'This could be an **alterantive** remedy'
- Syntactic parsing requires lexical information derived from morphological analysis
 - '**Children are** very intelligent'
 - '**Children is** very intelligent'

Morphology

Motivation

Morphological
analysis

Spell checkers
and spell
correctors

Outline

Morphology

Definitions

Morphological
analysis

Spell checkers
and spell
correctors

- 1 Morphology
 - Motivation
 - Definitions
 - Types of morphology
- 2 Morphological analysis
 - Finite-state automata
 - Finite-state transducers
- 3 Spell checkers and spell correctors

Definition of morphology

Morphology
Definitions

Morphological
analysis

Spell checkers
and spell
correctors

- Study of the structure of words
 - Phonology: word as a combination of phonemes
 - Orthography: word as a combination of graphemes
 - Morphology: word as a combination of morphemes
- Types of morphemes:
 - Stems: (e.g., 'work', 'of', 'mak'[e])
 - Affixes: always occur combined with other morphemes (e.g., -s, 'in-', '-able')
 - Prefixes: in + frequent
 - Suffixes: work + s
 - Infixes: [Arabic] ktb + CuCuC → kutub (books)
 - Circumfixes: en+light+en
- The resulting words can be classified into categories known as Part of Speech (POS): Noun, Verb, Adjective, Adverb, Preposition, . . .

Outline

Morphology

Types of morphology

Morphological
analysis

Spell checkers
and spell
correctors

- 1 Morphology
 - Motivation
 - Definitions
 - Types of morphology
- 2 Morphological analysis
 - Finite-state automata
 - Finite-state transducers
- 3 Spell checkers and spell correctors

Types of morphology

- Concatenative morphology: builds words up by concatenating morphemes (prefixes, suffixes). Frequent in the Indo-European languages.
 - **Inflectional morphology**: *stem* → *different forms of the same word*
Ex: work → worked
 - **Derivational morphology**: *stem* → *new words*
Ex: frequent → infrequent
 - **Compositional morphology**: *N words* → *new word*
Ex: fire + man → fireman
- Non-concatenative morphology: builds words by other mechanism (infixes). Frequent in the Semitic languages.
 - Ex: Root-Pattern morphology
Ex: [Arabic] ktb + CaCaCa → kataba [en: he wrote]

Inflectional morphology

Inflectional morphemes provide morphological information depending on the POS and language of the input word

- Nouns (N):

- Genre: [Spanish] niñ-o (M), niñ-a (F)
- Number: [Italian] italian-o (SG), italian-i (PL)
- Case: [German] die Rolle des Mann-es (Genitive)

- Verbs (V):

- Tens: want-ed (PAST), will want (no morpho. mark for future)
- Mode: [Spanish] com-er (indicative), com-ed (imperative)
- Aspect: want-ed (perfective), I am waiting (no morpho mark for imperfective)
- Voice: [Sweden] servera-s (PAS) [en: is served]

- Adjectives (A):

- Genre: [Spanish] blanc-o (M), blanc-a (F) [en: white]
- Number: [Spanish] blanco (SG), blanco-s (PL) [en: white]
- Comparison: cheap-er, more similar (not for all adjectives)

Derivational morphology

Morphology

Types of morphology

Morphological
analysis

Spell checkers
and spell
correctors

Derivational morphemes can change the POS and the meaning of the word

- Adjectivization: $V \rightarrow A$ or $N \rightarrow A$

Ex: react \rightarrow react-ive, employ \rightarrow employ-able
medicine \rightarrow medicin-al, use \rightarrow use-ful

- Nominalization: $V \rightarrow N$ or $A \rightarrow N$

Ex: watch \rightarrow watch-er, react \rightarrow react-ion
useful \rightarrow useful-ness

- Negativization:

Ex: frequent \rightarrow in-frequent, do \rightarrow un-do

Outline

Morphology

Morphological
analysis

Spell checkers
and spell
correctors

- 1 Morphology
 - Motivation
 - Definitions
 - Types of morphology
- 2 Morphological analysis
 - Finite-state automata
 - Finite-state transducers
- 3 Spell checkers and spell correctors

Goal of morphological analysis

Morphology

Morphological
analysis

Spell checkers
and spell
correctors

- Morphological recognition

Does word w belong to language L ?

- Morphological parsing

What is the morphological information related to word $w \in L$?

Ex: *word POS+Gen+Num+Case+Tense+... LEMMA (stem)*
men Noun+M+PL man

Resources required for morphological analysis

- Lists of regular (Reg) stems (ambiguities)
 - EX: Reg_V: walk
Reg_N: cat, fox, walk
- Lists of irregular (Irreg) stems (ambiguities)
 - EX: Irreg_pres_V: sing ... Irreg_past_V: sang sing
Irreg_sg_N: mouse ... Irreg_pl_N: mice mouse
- List of suffixes and prefixes (dealing with concatenative morphology)
 - EX: Inflec: s suffix, ing suffix
Deriv: able suffix, un prefix
- Morphotactics: general rules for combining morphemes
 - EX: Reg_N + s \rightarrow PL
Reg_V + ing \rightarrow Gerund
- Spelling rules: orthographic rules for combining letters
 - EX: E-insertion: $-(z,x,s,sh,ch)^s \rightarrow -(z,x,s,sh,ch)es$
Consonant-doubling: $-l^{ing} \rightarrow -lling$

Morphology

Morphological
analysis

Spell checkers
and spell
correctors

Types of morphological processors

Morphology

Morphological
analysis

Spell checkers
and spell
correctors

- Based on dictionaries: list of word forms [with their corresponding morphological information]
 - Ex: (write VPrI write, writes VPrI3S write, wrote VPsl write, ...)
 - + efficiency
 - + can be automatically generated/maintained from the resources
 - + language with 'simple' morphology (e.g., English)
 - languages with complex morphology (e.g., German, Finish, ...)
- Based on finite state automata (FSAs)
 - languages with complex morphology
- Based on finite state transducers (FSTs)

Outline

Morphology

Morphological
analysis

Finite-state
automata

Spell checkers
and spell
correctors

- 1 Morphology
 - Motivation
 - Definitions
 - Types of morphology
- 2 Morphological analysis
 - Finite-state automata
 - Finite-state transducers
- 3 Spell checkers and spell correctors

Finite state automata (FSA)

A FSA defines a function over words w of a regular language L .

$$M_L : w \rightarrow \{true, false\}$$

$$M = \langle Q, \Sigma, q_0, F, \sigma \rangle$$

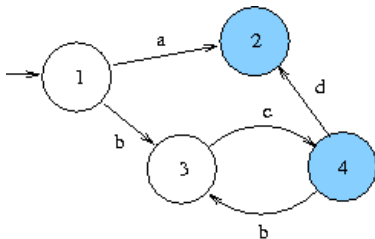
$Q = \{q_0, \dots, q_n\}$ finite set of states

$\Sigma = \{s_0, \dots, s_k\}$ finite set of symbols

$q_0 \in Q$ start state

$F \subset Q$ final states

$\sigma : Q \times \Sigma \rightarrow [Q \cup 2^Q]$ deterministic \vee non-det. transition function



$\frac{a|(bc)+d\{0,1\}}{a}$
bc
bcd
bcbcd
...

Morphology

Morphological
analysis

Finite-state
automata

Spell checkers
and spell
correctors

FSAs for lexical recognition

Morphology

Morphological
analysis

Finite-state
automata

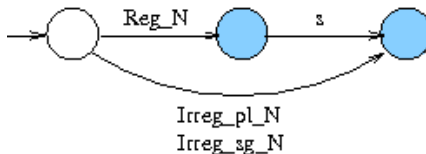
Spell checkers
and spell
correctors

An FSA can be the union/concatenation of different FSAs:

- FSAs generated from morphological rules
- FSAs generated from spelling rules
- FSAs generated from derivational rules
- FSAs generated from compositional rules

FSAs for lexical recognition

Example: FSA for English number nominal inflection



Examples of lists of stems

Reg_N	Irreg_sg_N	Irreg_pl_N
dog	mouse	mice
fox	foot	feet
tax		
donkey		

Morphology

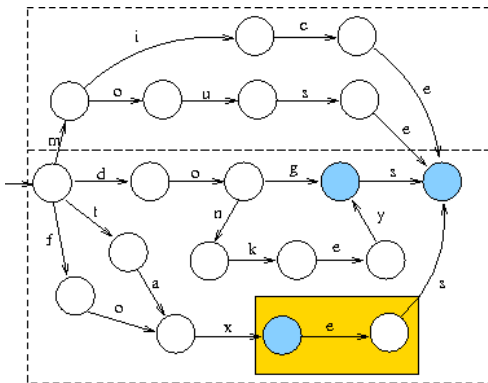
Morphological
analysis

Finite-state
automata

Spell checkers
and spell
correctors

FSAs for lexical recognition

Example: FSA for English number nominal inflection



Morphotactics: List Irreg_N

Morphotactics: noun + s = PL
over list Reg_N

SHOULD CORRECT WITH:

Spelling rule:
[s,x,z,sh,ch]^s=[s,x,z,sh,ch]es
over list Reg_N

Morphology

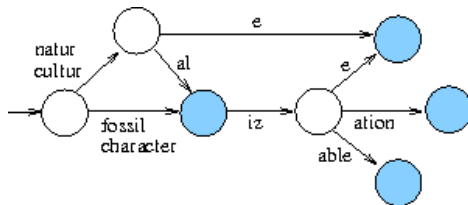
Morphological
analysis

Finite-state
automata

Spell checkers
and spell
correctors

FSAs for lexical recognition

Example: FSA derived from derivational rules



Not so productive as inflectional rules: 'jail', 'window', ... ?

Morphology

Morphological
analysis

Finite-state
automata

Spell checkers
and spell
correctors

FSAs for lexical recognition

- FSAs can be useful for recognising words
- FSAs are not able to output a word analysis

Input word (surface form)	Output analysis (lexical form)
dog dogs (word form)	dog+N+SG dog+N+PL (lemma+Features)

- A more sophisticated technique is required: FSTs

Morphology

Morphological
analysis

Finite-state
automata

Spell checkers
and spell
correctors

Outline

Morphology

Morphological
analysis

Finite-state
transducers

Spell checkers
and spell
correctors

- 1 Morphology
 - Motivation
 - Definitions
 - Types of morphology
- 2 Morphological analysis
 - Finite-state automata
 - Finite-state transducers
- 3 Spell checkers and spell correctors

Finite state transducers (FSTs)

A FST defines a relation between regular languages L_1 and L_2 .

$$T = \langle Q, \Sigma, \Delta, q_0, F, \sigma, \delta \rangle$$

$Q = \{q_0, \dots, q_n\}$ finite set of states

$\Sigma = \{s_0, \dots, s_k\}$ finite set of input symbols

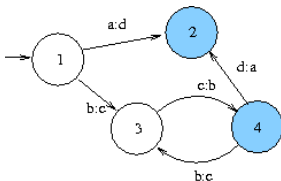
$\Delta = \{t_0, \dots, t_m\}$ finite set of output symbols

$q_0 \in Q$ start state

$F \subset Q$ final states

$\sigma : Q \times \Sigma \rightarrow 2^Q$ transition function

$\delta : Q \times \Sigma \rightarrow \Delta$ output function



$a (bc)+d\{0,1\}$	$d (cb)+a\{0,1\}$
a	d
bc	cb
bcd	cba
bcbc	cbcb
bcbcd	cbcba
...	

Morphology

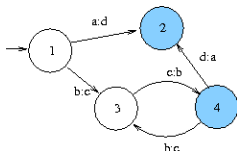
Morphological
analysis

Finite-state
transducers

Spell checkers
and spell
correctors

Finite state transducers (FSTs)

- Inversion: $T : L_1 \rightarrow L_2 \implies T^{-1} : L_2 \rightarrow L_1$



$T : b:c \implies c \rightarrow b \implies \text{Ex: } cbc b \rightarrow bcb c$

$T^{-1} : b:c \implies b \rightarrow c \implies \text{Ex: } bcb c \rightarrow cbc b$

- Composition: $T_a : L_1 \rightarrow L_2 \wedge T_b : L_2 \rightarrow L_3 \implies T_a \circ T_b : L_1 \rightarrow L_3$
- $x:x \equiv x$
- Non-consumption symbol: $\epsilon \in \Sigma, \epsilon \in \Delta$

Morphology

Morphological
analysis

Finite-state
transducers

Spell checkers
and spell
correctors

FSTs for morphological analysis

Morphology

Morphological
analysis

Finite-state
transducers

Spell checkers
and spell
correctors

We want a FST being a relation between

- Surface form: $L_1 = \{w | w \text{ is word form}\}$
- Lexical form: $L_2 = \{ \langle I, F \rangle | I \text{ is lemma} \wedge F \text{ are morphological features} \}$

So that we get a morphological parser

- Ex: $\text{dogs} \rightarrow \text{dog} + \text{N} + \text{PL}$
Ex: $\text{dog} \rightarrow \text{dog} + \text{N} + \text{SG}$

Inverting that FST, we get a word forms generator

- Ex: $\text{dog} + \text{N} + \text{PL} \rightarrow \text{dogs}$
Ex: $\text{dog} + \text{N} + \text{SG} \rightarrow \text{dog}$

FSTs for morphological analysis

Two-level construction:

1 T_{lex} : A FST that computes morphotactics

Ex: $\text{Reg_N}^{\wedge} \text{s\#} \rightarrow \text{Reg_N} + \text{N} + \text{PL}$.

Ex: $\text{dog}^{\wedge} \text{s\#} \rightarrow \text{dog} + \text{N} + \text{PL}$, $\text{fox}^{\wedge} \text{s\#} \rightarrow \text{fox} + \text{N} + \text{PL}$

2 T_{inter}^i : FSTs each computing a spelling rule (orthographic regularization)

Ex: $-\{z,x,s,sh,ch\}es \rightarrow -\{z,x,s,sh,ch\}^{\wedge} \text{s\#}$

Two-level processing:

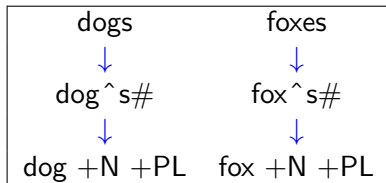
surface level

$T_{inter}^1, \dots, T_{inter}^k$

intermediate level

T_{lex}

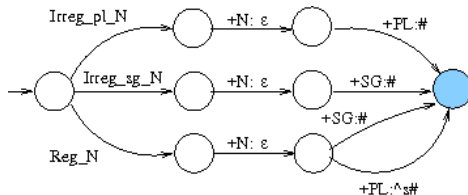
lexical level



FSTs for morphological analysis

- 1 T_{lex} : FST that computes morphotactics
Example: FST for English number nominal inflection

T_{num_nouns}



Examples of lists of stems/forms

Reg_N	Irreg_sg_N	Irreg_pl_N
dog	mouse	m o:i u:ε s:c e
fox	foot	f o:e o:e t
tax		
donkey		

Morphology

Morphological
analysis

Finite-state
transducers

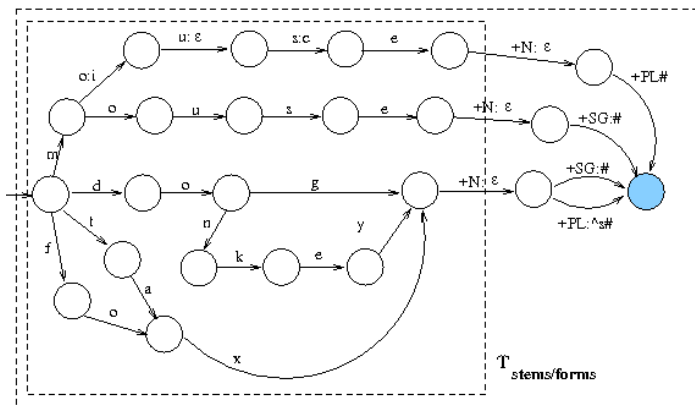
Spell checkers
and spell
correctors

FSTs for morphological analysis

1 T_{lex} : FST that computes morphotactics

Example: FST for English number nominal inflection

$$T_{lex} = T_{stems/forms} \circ T_{num_nouns}$$



$fox\wedge s\# \rightarrow fox+N+PL$!! (requires spelling rules)

Morphology

Morphological
analysis

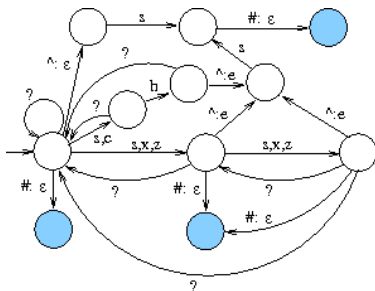
Finite-state
transducers

Spell checkers
and spell
correctors

FSTs for morphological analysis

2 T_{inter}^i : FSTs that compute spelling rules

Example: FST for E-insertion rule



'?': other symbol

e-insertion cases

$$\text{foxes} \rightarrow \text{fox}^s \#$$
$$\text{bosses} \rightarrow \text{boss}^s \#$$

flashes \rightarrow flash^s#

• • •

regular cases

$\text{dogs} \rightarrow \text{dog}^s \#$

• • •

FSTs for morphological analysis

2 T_{inter}^i : FST that computes spelling rules

Some other examples of spelling rules:

- **Consonant doubling**: two-syllable word stressed in the last one with ending CVC pattern double last consonant before *-ing/-ed*
EX: control → controlling
- **E-deletion**: Silent *-e* removed before *-ing/-ed*
EX: remove → removed
- **E-insertion**: *-e* added after ending *-s,-z,-x,-ch,-sh*, before *-s*
EX: flash → flashes
- **Y-replacement**: *-y* changes to *-ie* before *-s* or to *-i* before *-ed*
EX: cry → cries, cried
- **K-insertion**: verbs ending with *1-vowel+c* add *-k* before *-ed*
EX: panic → panicked

Morphology

Morphological
analysis

Finite-state
transducers

Spell checkers
and spell
correctors

Exercise

Morphology

Morphological
analysis

Finite-state
transducers

Spell checkers
and spell
correctors

- Generate a FST for the inflection of verbs *sing* and *work*
- Add the inflection of verb *make* to the previous FST

Outline

Morphology

Morphological
analysis

Spell checkers
and spell
correctors

- 1 Morphology
 - Motivation
 - Definitions
 - Types of morphology
- 2 Morphological analysis
 - Finite-state automata
 - Finite-state transducers
- 3 Spell checkers and spell correctors

Spell checkers

Morphology

Morphological
analysis

Spell checkers
and spell
correctors

- **Goal:** given a piece of text, recognise the word forms that do not belong to the text language L

- **Possible approach:**

FSA_L OR FST_L

$S = \text{Tokenizer}(\text{text})$ (sequence of forms)

for each $x \in S$

if $FSA_L(x)$ then print("x")

else print("**x**")

Spell correctors

- **Goal:** given a word form, provide a list of possible correct forms.

- **Possible approach:**

$D = \{y_i : y_i \in L\}$ generated by applying FST_L

$S = \text{Tokenizer}(\text{text})$ (sequence of forms)

for each $x \in S$

if $x \in D$ then print(x)

else

$D' = \{y \in D : |\text{length}(x) - \text{length}(y)| \leq \gamma\}$

$C = \emptyset$

for each $y \in D'$

$d = \text{distance}(x, y)$

if ($d \leq \delta$) then

$C = C + \{< y, d >\}$

print_Nbest_candidates(C, N)

$\delta = 2$ and $\gamma = 2$ seem to be enough for standard text

Spell correctors

Morphology

Morphological
analysis

Spell checkers
and spell
correctors

- Edit distance: minimum number of insertions, deletions, swaps to achieve y from x
- **Weighted edit distance**: minimum **cost** of insertions, deletions, swaps to achieve y from x
 - Cost of insertion/deletion = 1
 - Cost of swap = $s(a, b)$: (typo - Manhattan distance in a keyboard)
 - Total cost = $d(x, y)$:
 - Compute cost matrix E , with dimension $m \times n$ (lengths of x and y) using dynamic programming
 - $d(x, y) = E(m, n)$

Spell correctors

Cost matrix computation

	y1	y2	y3	y4	
	0	1	2	3	4
x1	1				
x2	2				
x3	3				

deletion (+1)

insertion (+1)

swap

$+s(x_i, y_j)$

$$E(i, j) = \min(\text{Cost}_{del}, \text{Cost}_{ins}, \text{Cost}_{swap})$$

$$\begin{cases} \text{Cost}_{del} = E(i-1, j) + 1 \\ \text{Cost}_{ins} = E(i, j-1) + 1 \\ \text{Cost}_{swap} = E(i-1, j-1) + s(x_i, y_j) \end{cases}$$

$s(x_i, y_j)$	a	b	c	d	e
a	0				
b	0.5	0			
c	0.3	0.3	0		
d	0.2	0.2	0.1	0	
e	0.3	0.4	0.2	0.1	0

$s(x_i, y_j)$ normalised to 1.0

Morphology

Morphological analysis

Spell checkers and spell correctors

Exercise

Morphology

Morphological
analysis

Spell checkers
and spell
correctors

- Compute the weighted edit distance between 'dom' and 'come'