Natural Language Processing

Lecture 2: Words and Morphology

Tokenization

Input: raw text

Output: sequence of **tokens** normalized for easier processing.

Dr. Smith said tokenization of English is “harder than you’ve thought.” When in New York, he paid $12.00 a day for lunch and wondered what it would be like to work for AT&T or Google, Inc.

Morphology

- Morpheme
- Inflectional morphology
- Irregularity
- Derivational morphology

Morphological Parsing

Input: a word

Output: the word's stem(s) and features expressed by other morphemes.

Example: geese → goose +N +Pl

gooses → goose +V +3P +Sg

dog → {dog +N +Sg, dog +V}

leaves → {leaf +N +Pl, leave +V +3P +Sg}

Turkish Example

uygarlaştıramadıklarımızdanmışsınızcasına

“(behaving) as if you are among those whom we were not able to civilize”

uygar “civilized”

+laş “become”

+tır “cause to”

+ama “not able”

+dık past participle

+lar plural

+ımız first person plural possessive (“our”)

+dan second person plural (“y’ all”)

+mış past

+sınız ablative case (“from/among”)

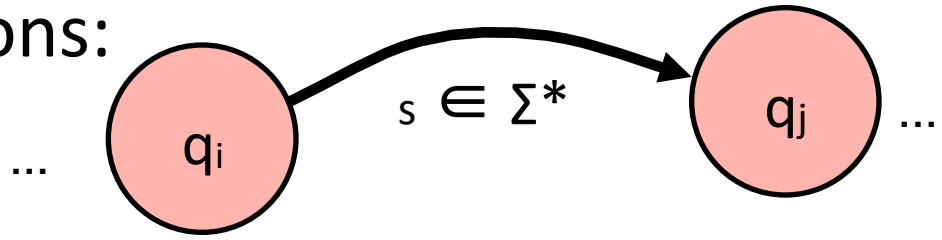
+casına finite verb → adverb (“as if”)

Four Solutions

1. Table
2. Trie
3. Finite-state automaton
4. Finite-state transducer

Finite-State Automaton

- Q: a finite set of states
- $q_0 \in Q$: a special start state
- $F \subseteq Q$: a set of final states
- Σ: a finite alphabet
- Transitions:



- Encodes a **set** of strings that can be recognized by following paths from q_0 to some state in F.

FSA for English Nouns

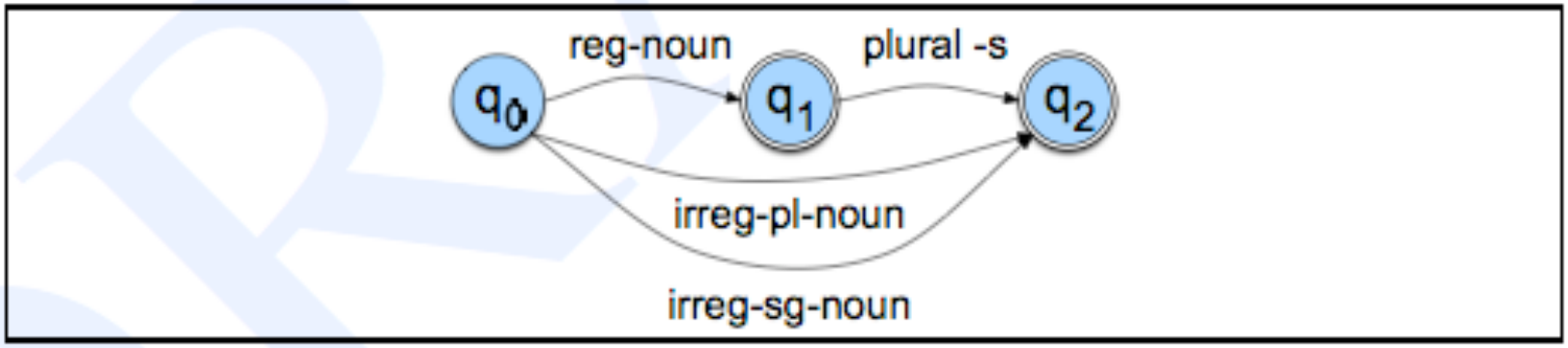


Figure 3.3 A finite-state automaton for English nominal inflection.

reg-noun	irreg-pl-noun	irreg-sg-noun	plural
fox	geese	goose	-s
cat	sheep	sheep	
aardvark	mice	mouse	

FSA for English Adjectives

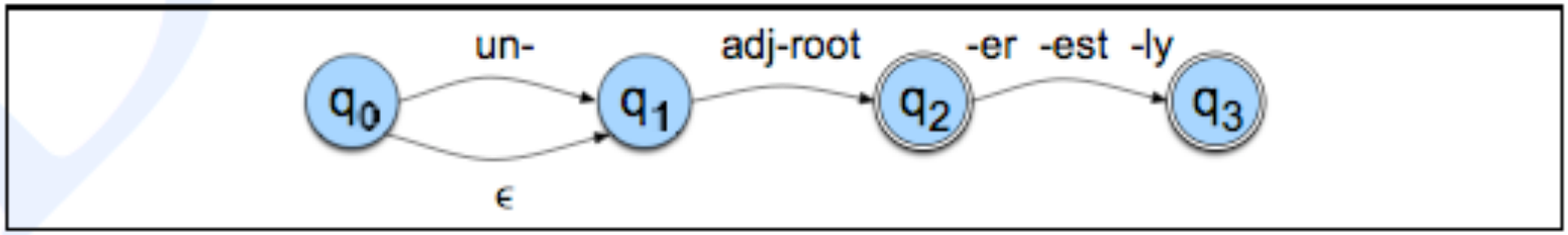


Figure 3.5 An FSA for a fragment of English adjective morphology: Antworth's Proposal #1.

FSA for English Derivational Morphology

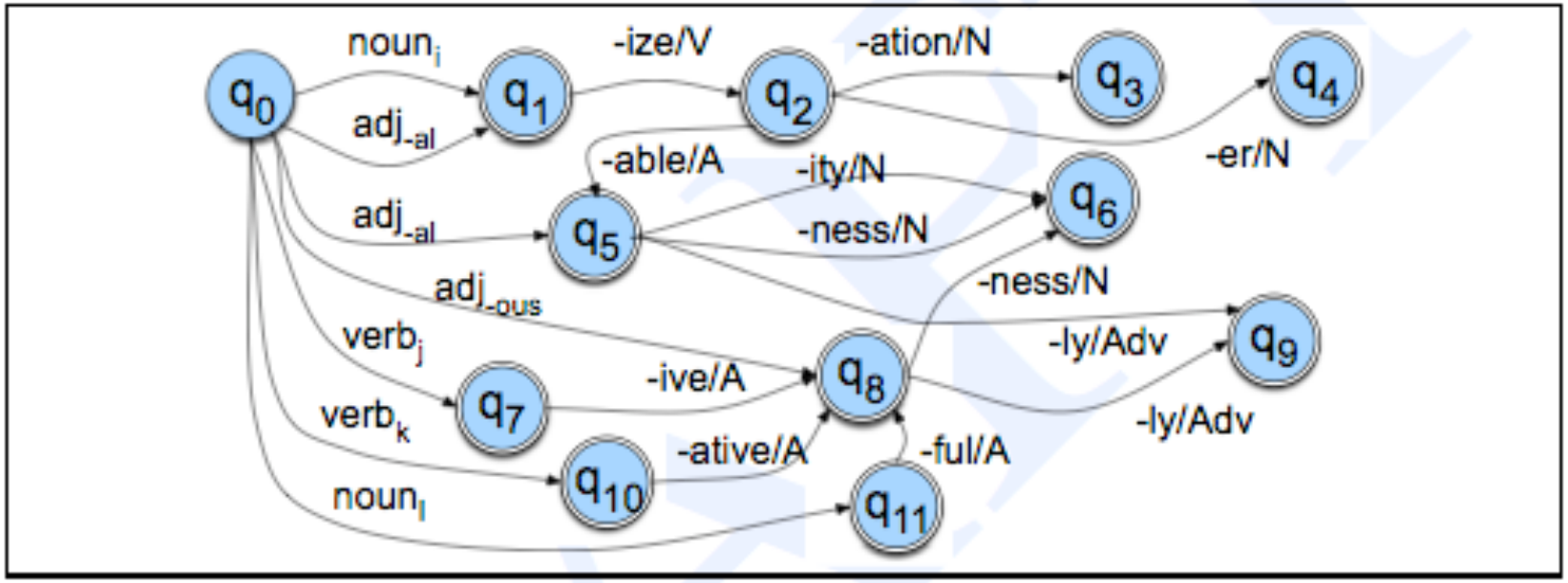


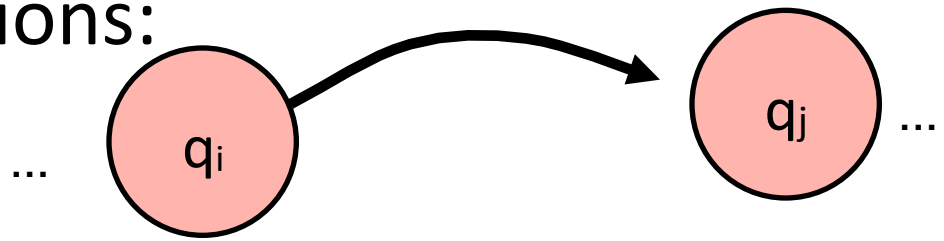
Figure 3.6 An FSA for another fragment of English derivational morphology.

Four Solutions

1. Table
2. Trie
3. Finite-state automaton
4. Finite-state transducer

Finite State Transducers

- Q: a finite set of states
- $q_0 \in Q$: a special start state
- $F \subseteq Q$: a set of final states
- Σ and Δ: two finite alphabets
- Transitions:



s : t

$s \in \Sigma^*$ and $t \in \Delta^*$

Morphological Parsing with FSTs

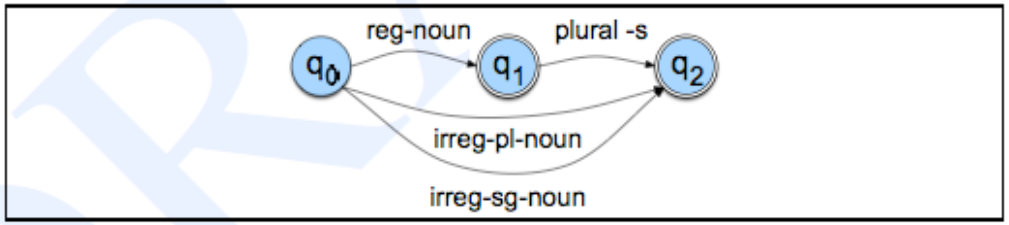


Figure 3.3 A finite-state automaton for English nominal inflection.

reg-noun	irreg-pl-noun	irreg-sg-noun	plural
fox	geese	goose	-s
cat	sheep	sheep	
aardvark	mice	mouse	

reg-noun	irreg-pl-noun	irreg-sg-noun
fox	g o:e o:e s e	goose
cat	sheep	sheep
aardvark	m o:i u:ϵ s:c e	mouse

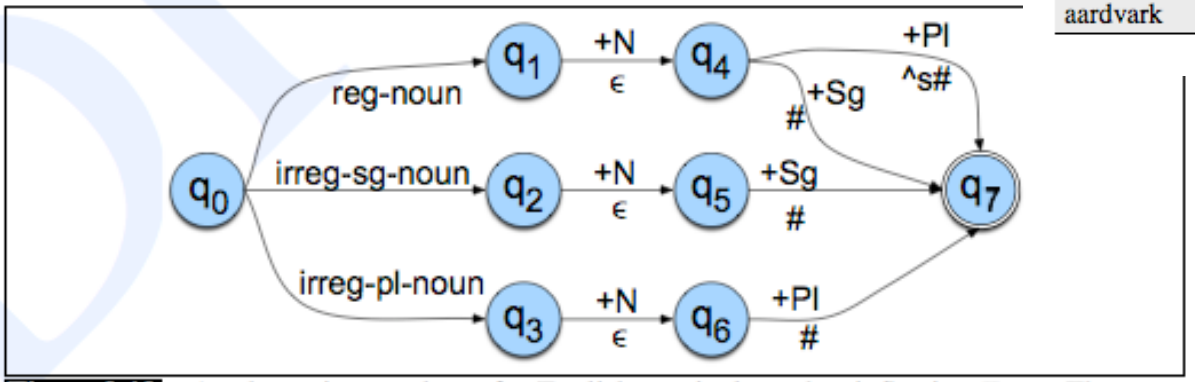


Figure 3.13 A schematic transducer for English nominal number inflection T_{num}. The symbols above each arc represent elements of the morphological parse in the lexical tape; the symbols below each arc represent the surface tape (or the intermediate tape, to be described later), using the morpheme-boundary symbol ^ and word-boundary marker #. The labels on the arcs leaving q_0 are schematic, and need to be expanded by individual words in the lexicon.

Note "same symbol" shorthand.

^ denotes a morpheme boundary.

# denotes a word boundary.

English Spelling

Name	Description of Rule	Example
Consonant doubling	1-letter consonant doubled before *-ing/-ed*	beg/begging
E deletion	Silent e dropped before *-ing* and *-ed*	make/making
E insertion	e added after *-s,-z,-x,-ch, -sh* before *-s*	watch/watches
Y replacement	*-y* changes to *-ie* before *-s*, *-i* before *-ed*	try/tries
K insertion	verbs ending with *vowel* + *-c* add *-k*	panic/panicked

The E Insertion Rule as a FST

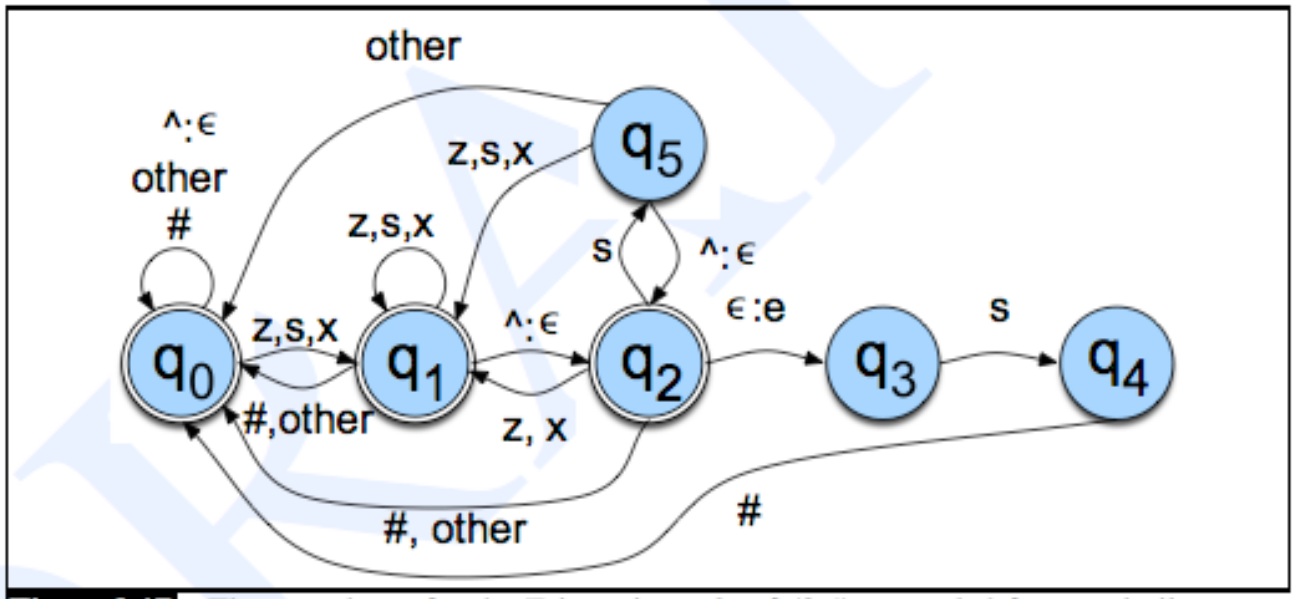


Figure 3.17 The transducer for the E-insertion rule of (3.4), extended from a similar transducer in Antworth (1990). We additionally need to delete the # symbol from the surface string; this can be done either by interpreting the symbol # as the pair #:ϵ, or by postprocessing the output to remove word boundaries.

$$\epsilon \rightarrow \mathrm{e} / \left\{ \begin{array}{c} \mathrm{s} \\ \mathrm{x} \\ \mathrm{z} \end{array} \right\} \wedge \_\mathrm{s}\#$$

Combining FSTs

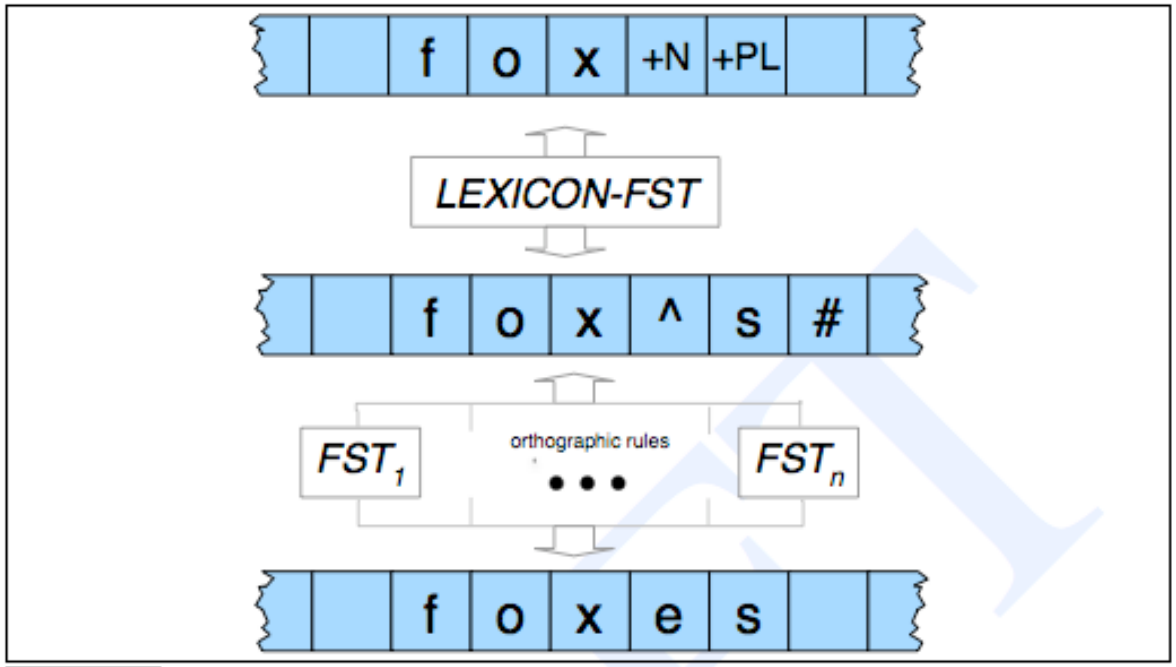


Figure 3.19 Generating or parsing with FST lexicon and rules

parse

generate

FST Operations

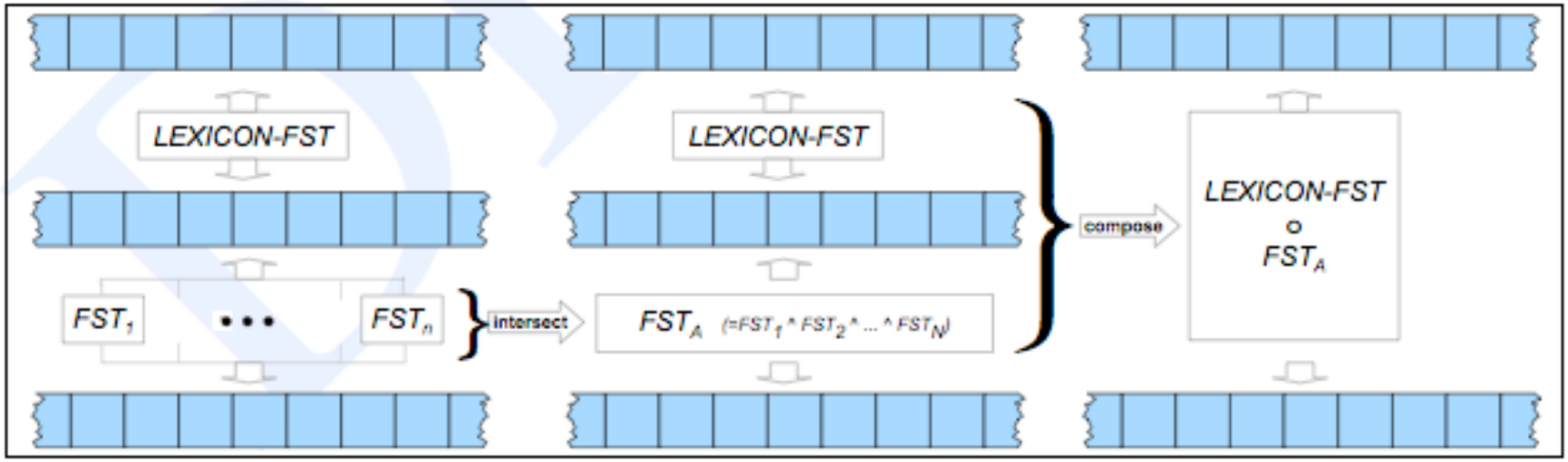


Figure 3.21 Intersection and composition of transducers.

Stemming (“Poor Man’s Morphology”)

Input: a word

Output: the word’s stem (approximately)

Examples from the Porter stemmer:

- -sses → -ss
- -ies → i
- -ss → s

no	no
noah	noah
nob	nob
nobility	nobil
nobis	nobi
noble	nobl
nobleman	nobleman
noblemen	noblemen
nobleness	nobl
nobler	nobler
nobles	nobl
noblesse	nobless
noblest	noblest
nobly	nobli
nobody	nobodi
noces	noce
nod	nod
nodded	nod
nodding	nod
noddle	noddl
noddles	noddl
noddy	noddi
nods	nod