Natural Language Processing 1

Lecture 3: Morphological processing

Katia Shutova

ILLC
University of Amsterdam

4 November 2020

Stems and affixes

- morpheme: the minimal information carrying unit
- affix: morpheme which only occurs in conjunction with other morphemes
- words made up of stem (more than one for compounds) and zero or more affixes.
  e.g., *dog+s*, *book+shop+s*
- *slither*, *slide*, *slip* etc have somewhat similar meanings, but *sl-* not a morpheme.

Affixation

- **suffix**: *dog +s*, *truth +ful*
- **prefix**: *un+ wise* (derivational only)
- **infix**: Arabic stem *k_t_b*: *kataba* (he wrote); *kotob* (books)
  In English: *sang* (stem *sing*): not productive
  e.g., (maybe) *absobloodylutely*
- **circumfix**: not in English
  German *ge+kauf+t* (stem *kauf*, affix *ge-t*)

Productivity

productivity: whether affix applies generally, whether it applies to new words

sing, *sang*, *sung*

ring, *rang*, *rung*

BUT: *ping*, *pinged*, *pinged*

So this infixation pattern is not productive:

sing, *ring* are irregular

Productivity

productivity: whether affix applies generally, whether it applies to new words
sing, *sang*, *sung*
ring, *rang*, *rung*
BUT: *ping*, *pinged*, *pinged*
So this infixation pattern is not productive:
sing, *ring* are irregular

Inflectional morphology

- e.g., plural suffix *+s*, past participle *+ed*
- sets slots in some paradigm
  e.g., tense, aspect, number, person, gender, case
- inflectional affixes are not combined in English
- generally fully productive (except irregular forms)
  e.g., *texted*

Derivational morphology

- e.g., *un-*, *re-*, *anti-*, *-ism*, *-ist* etc
- broad range of semantic possibilities, may change part of speech
- indefinite combinations
  e.g., *antiantidisestablishmentarianism*
  anti-anti-dis-establish-ment-arian-ism
- generally semi-productive: e.g., *escapee*, *textee*, *?dropee*, *?snoree*, **cricketee* (* and ?)
- zero-derivation: e.g. *tango*, *waltz*

Guess the structure...

- ruined
- settlement
- inventive
- archive
- unionised

Guess the structure...

- ruined
- settlement
- inventive
- archive
- unionised

Guess the structure...

- ruined
- settlement
- inventive
- archive
- unionised

Guess the structure...

- ruined
- settlement
- inventive
- archive
- unionised

Guess the structure...

- ruined
- settlement
- inventive
- archive
- unionised

Internal structure and ambiguity

Morpheme ambiguity: stems and affixes may be individually ambiguous: e.g. *dog* (noun or verb), *+s* (plural or 3persg-verb)

Structural ambiguity: e.g., *shorts* or *short -s*
unionised could be *union -ise -ed* or *un- ion -ise -ed*

Bracketing: *un- ion -ise -ed*

- **((un- ion) -ise) -ed*
- *un- ((ion -ise) -ed)*

Using morphological processing in NLP

- compiling a full-form lexicon
- stemming for IR (not linguistic stem)
- lemmatization, i.e. morphological analysis:
  - finding stems and affixes as a precursor to parsing (often inflections only)
- generation
  - Morphological processing may be bidirectional: i.e., parsing and generation.

```
party + PLURAL <-> parties
sleep + PAST_VERB <-> slept
```

Compiling a full form lexicon

run
runs
ran
running

Compiling a full form lexicon

run
runs
ran
running

Бегаю
Бегу
Бегаешь
Бежишь
Бегает
Бежит
Бегаем
Бежим
Бегаете
Бежите
Бегают
Бегут

Бегал
Бежал
Побежал
Бегала
Бежала
Побежала
Бегало
Бежало
Побежало
Бегали
Бежали
Побежали
Бегай
Беги
Побеги
Бегайте
Бегите
Побегите

Побегу
Побежишь
Побежит
Побежим
Побежите
Побегут
Бегущий
Бежавший
Бежавшая
Бегущая
Бегущее
Бежавшее
Побежавший
Побежавшая
Побежавшее
Побежав
Побегав
Бегая

Using morphological processing in NLP

- compiling a full-form lexicon
- stemming for IR (not linguistic stem)
- lemmatization, i.e. morphological analysis:
  - finding stems and affixes as a precursor to parsing (often inflections only)
- generation
  - Morphological processing may be bidirectional: i.e., parsing and generation.

```
party + PLURAL <-> parties
sleep + PAST_VERB <-> slept
```

Morphological processing

Surface form mapped to stem(s) and affixes (or abstractions of affixes):

OPTION 1	*pinged* / *ping-ed*
OPTION 2	*pinged* / *ping PAST_VERB* *pinged* / *ping PSP_VERB* *sang* / *sing PAST_VERB* *sung* / *sing PSP_VERB*

Lexical requirements for morphological processing

- affixes, plus the associated information conveyed by the affix

```
ed PAST_VERB
ed PSP_VERB
s PLURAL_NOUN
```

- irregular forms, with associated information similar to that for affixes

```
began PAST_VERB begin
begun PSP_VERB begin
```

- stems with syntactic categories
  e.g. to avoid *corpus* being analysed as *corpu -s*

Spelling rules

- English morphology is essentially concatenative
  - irregular morphology — inflectional forms have to be listed
- regular phonological and spelling changes associated with affixation, e.g.
  - *-s* is pronounced differently with stem ending in *s*, *x* or *z*
  - spelling reflects this with the addition of an *e* (*boxes* etc)
- in English, description is independent of particular stems/affixes

e-insertion

e.g. *box^s* to *boxes*

$$\varepsilon \rightarrow \text{e} / \left\{ \begin{array}{c} \text{s} \\ \text{x} \\ \text{z} \end{array} \right\} \text{\^{}} \_ \; s$$

- map 'underlying' form to surface form
- mapping is left of the slash, context to the right
- notation:

_	position of mapping
ε	empty string
^	affix boundary — stem ^ affix

- same rule for plural and 3sg verb
- formalisable/implementable as a finite state transducer

e-insertion

e.g. *box^s* to *boxes*

$$\varepsilon \rightarrow \text{e} / \left\{ \begin{array}{c} \text{s} \\ \text{x} \\ \text{z} \end{array} \right\} \text{\^{}} \_ \; s$$

- map 'underlying' form to surface form
- mapping is left of the slash, context to the right
- notation:

_	position of mapping
ε	empty string
^	affix boundary — stem ^ affix

- same rule for plural and 3sg verb
- formalisable/implementable as a finite state transducer

e-insertion

e.g. *box^s* to *boxes*

$$\varepsilon \rightarrow \text{e} / \left\{ \begin{array}{c} \text{s} \\ \text{x} \\ \text{z} \end{array} \right\} \text{\^{}} \_ \; s$$

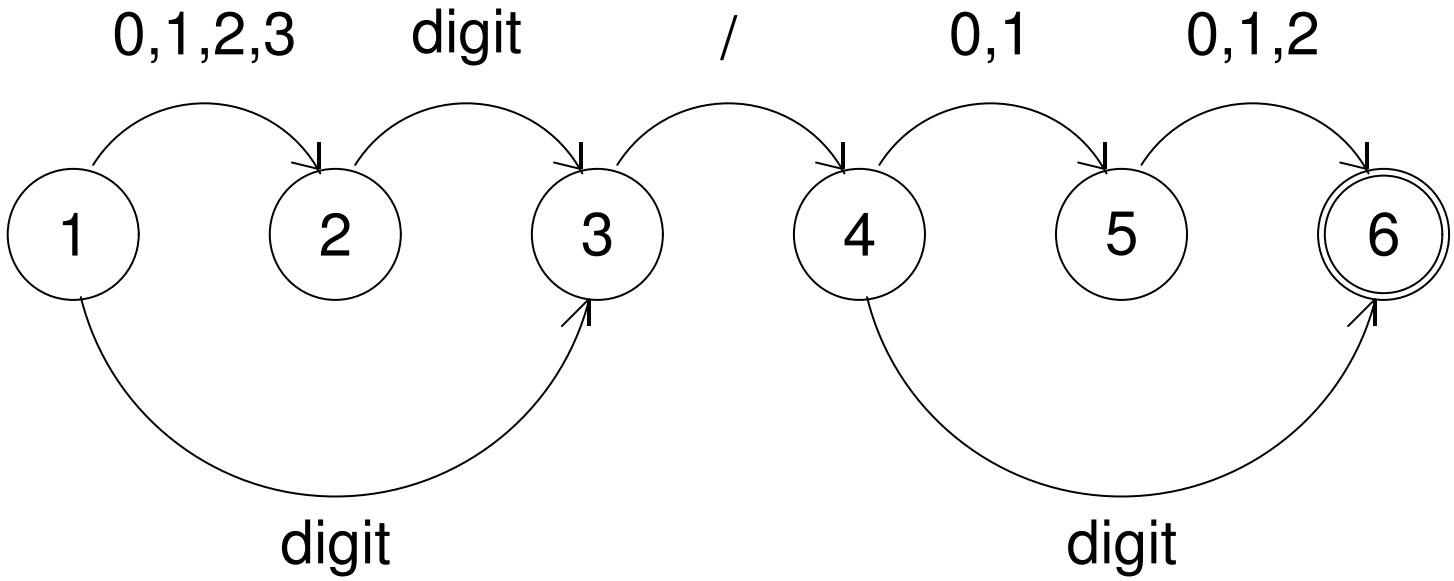
- map 'underlying' form to surface form
- mapping is left of the slash, context to the right
- notation:

_	position of mapping
ε	empty string
^	affix boundary — stem ^ affix

- same rule for plural and 3sg verb
- formalisable/implementable as a finite state transducer

Finite state automata for recognition

day/month pairs: e.g. 12/2, 1/12 etc.



- non-deterministic — after input of '2', in state 2 and state 3.
- double circle indicates accept state
- accepts e.g., 11/3 and 3/12
- also accepts 37/00 — overgeneration

e-insertion

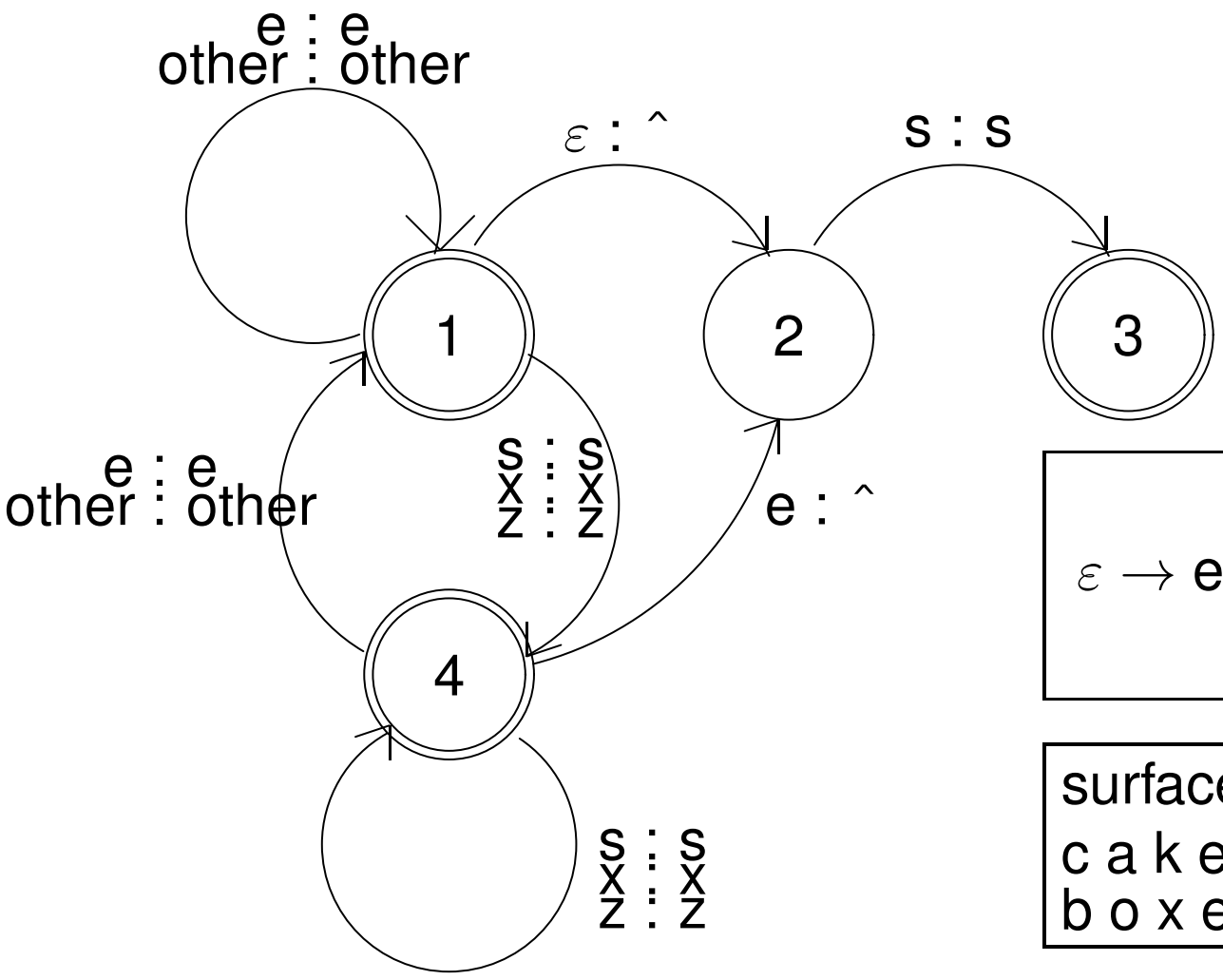
e.g. *box^s* to *boxes*

$$\varepsilon \rightarrow \mathrm{e} / \left\{ \begin{array}{c} \mathrm{s} \\ \mathrm{x} \\ \mathrm{z} \end{array} \right\} \hat{} \_ \, s$$

- notation:

_	position of mapping
ε	empty string
^	affix boundary — stem ^ affix

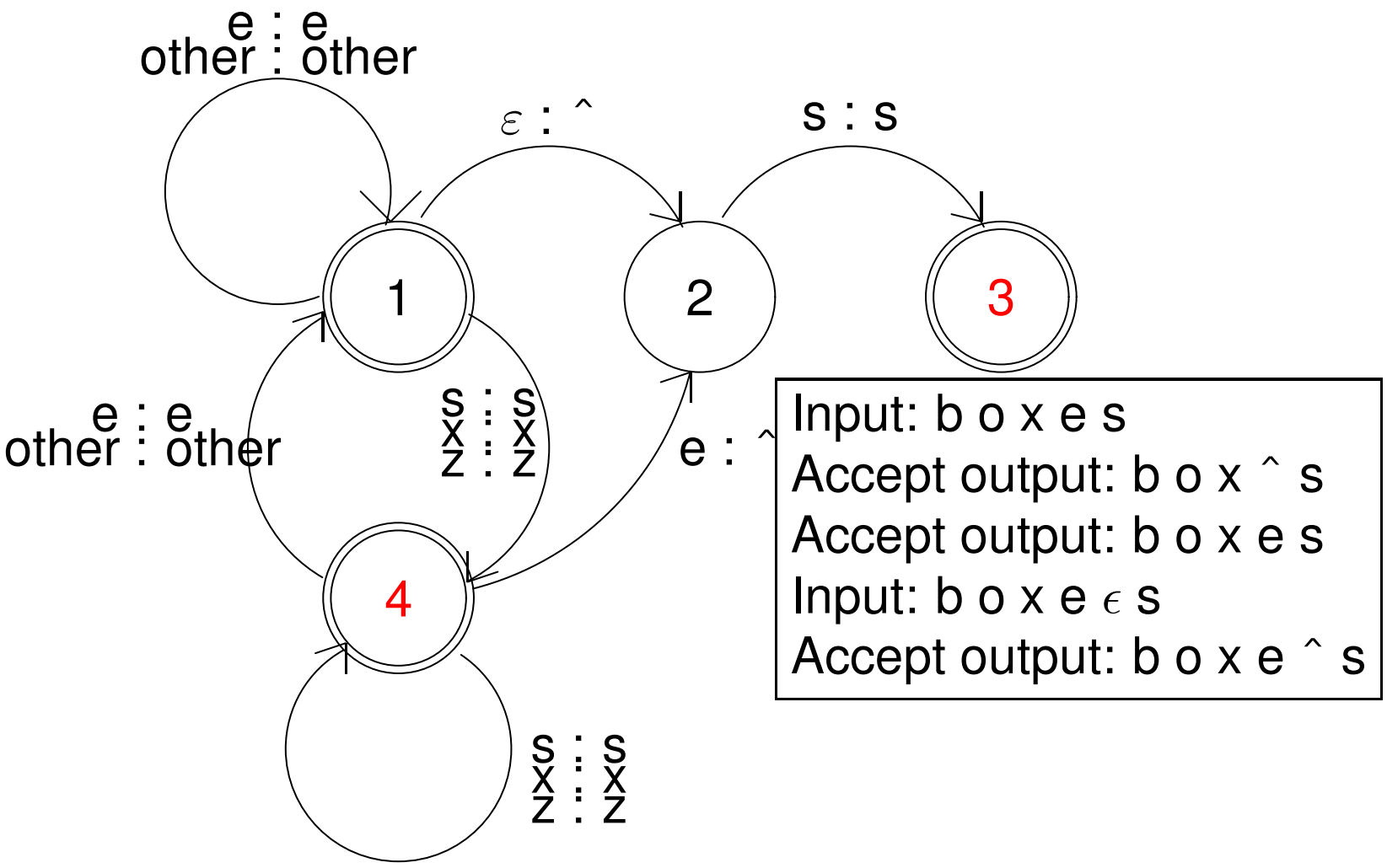
Finite state transducer



$$\varepsilon \rightarrow e / \left\{ \begin{array}{c} s \\ x \\ z \end{array} \right\} \hat{} \_ s$$

surface : underlying
c a k e s ↔ c a k e ^ s
b o x e s ↔ b o x ^ s

Analysing *b o x e s*



Using FSTs

- FSTs assume tokenization (word boundaries) and words split into characters. One character pair per transition!
- **Analysis**: return character list with affix boundaries, so enabling lexical lookup.
- **Generation**: input comes from stem and affix lexicons.
- One FST per spelling rule: either compile to big FST or run in parallel.
- FSTs do not allow for internal structure:
  - can't model *un- ion -ize -d* bracketing.

How is morphological processing implemented?

- rule-based methods, e.g. the Porter stemmer
  - part of NLTK toolkit
  - used in the practical
- probabilistic models for morphological segmentation
- neural models with character-level input
  (discussed later in the course)

Acknowledgement

Some slides were adapted from Ann Copestake and Tejaswini Deoskar