L90: Overview of Natural Language Processing
Lecture 2: Morphology and Finite State Techniques

Weiwei Sun

Department of Computer Science and Technology
University of Cambridge

Michaelmas 2020/21
Some yinkish dripners blorked quastofically into the nindin with the pidibs. ... dripn+ER+S blork+ED quastofical+LY into the nindin with the pidib+S

words have internal structures

Lecture 2: Morphology and Finite State Techniques
1. A brief introduction to morphology
2. Using morphology in NLP
3. Aspects of morphological processing
4. Finite state techniques
Morphology
**Morpheme**

*Morphemes* are the *smallest meaningful units* of language. Words are composed of morpheme(s).
Morphemes are the smallest meaningful units of language. Words are composed of morpheme(s).

**Affix**: morpheme which only occurs in conjunction with other morphemes.
- suffix (*units*), prefix (*in*complete), infix, circumfix
**Morpheme**

*Morphemes* are the *smallest meaningful units* of language. Words are composed of morpheme(s).

**Affix**: morpheme which only occurs in conjunction with other morphemes.
- suffix (*units*), prefix (*in*complete), infix, circumfix

**Root**: nucleus of the word that affixes attach too.
Infix Tagalog (Philippines)

Infix: -um-

basa  b-um-asa  sulat  s-um-ulet
read  read.PAST  write  wrote

Source: J Hana & A Feldman. ESSLLI 2013: Computational Morphology.
Infix Tagalog (Philippines)

**Infix:** -um-

- **basa** → **b-um-asa** → read
- **sulat** → **s-um-ulat** → write

**Circumfix:** occur on both sides

- **berg** → **ge-berg-te** → mountain
- **ge-berg-te** → **ge-berg-te** → mountains

**Dutch collectives**

- *ge-berg*
- *berg-te*


**Productivity**: whether affix applies generally, whether it applies to new words

- **sing**, **sang**, **sung**
- **ring**, **rang**, **rung**
Productivity: whether affix applies generally, whether it applies to new words

- **sing, sang, sung**
- **ring, rang, rung**

- But, **ping, pinged, pinged**

This infixation pattern is not productive: **sing, ring** are **irregular**
Inflection and derivation

**Inflection** creates new forms of the same word

- e.g. *bring*, *brought*, *brings*, *bringing*
- generally fully productive (modulo irregular forms)
- tends to affect only its *syntactic function*

**Derivation** creates new words

- e.g. *logic*, *logical*, *illogical*, *illogicality*, *logician*, etc.
- generally semi-productive: e.g., *escapee*, *textee*, *?dropee*, *?snoree*, *cricketee* (* and ?*)
- tends to be more irregular; the meaning is more idiosyncratic and less compositional.
- tends to affects the *meaning* of the word, and may change part-of-speech
Compound and multiword expression

Root: nucleus of the word that affixes attach too.

**Compounds** contain more than one root.

(1) a. railway  
    b. beam-width  
    c. sunset

**Multiword expression**: combinations of two or more words that exhibit syntactic and semantic idiosyncratic behavior.

<table>
<thead>
<tr>
<th>Fixed</th>
<th>(Syntactically) flexible</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>by and large</em></td>
<td><em>put on the clothes</em></td>
</tr>
<tr>
<td></td>
<td><em>put the clothes on</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-compositional</th>
<th>Semi-compositional</th>
<th>Compositional</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>kick the bucket</em></td>
<td><em>spill the beans</em></td>
<td><em>strong tea</em></td>
</tr>
<tr>
<td></td>
<td>(reveal the secret)</td>
<td></td>
</tr>
</tbody>
</table>
**Stem**: word without its inflectional affixes = root + all derivational affixes.
**Stem**: word without its inflectional affixes = root + all derivational affixes.

*bookshopped*
**Stem**: word without its inflectional affixes = root + all derivational affixes.

*bookshopped*

```
stem: bookshopp
```
**Stem:** word without its inflectional affixes = root + all derivational affixes.

*bookshopped*

**Lexeme:** the set of all forms related by inflection (but not derivation).

\{ *bookshops*, *bookshopped*, *bookshopping*, \ldots \}

**Lemma:** the *canonical/base/dictionary/citation* form of a lexeme chosen by convention.

*bookshop* (cf. the stem—*bookshopp*)
Etymology

*slither, slide, slip* etc have somewhat similar meanings; but *sl-* is not a morpheme.

*slith, slid* and *slip* are historically related.

See [www.etymonline.com/word/slide](http://www.etymonline.com/word/slide)
Internal structure: order

The order of morphemes matters

- talk-ed $\neq$ *ed-talk
- re-write $\neq$ *write-re
- un-kind-ly $\neq$ *kind-un-ly

Suffixing is more frequent than prefixing and far more frequent than infixing/circumfixing

- Postpositional and head-final languages use suffixes and no prefixes. cf. harmonic order: $\langle VO, PO \rangle$, $\langle OV, OP \rangle$
- Prepositional and head-initial languages use not only prefixes but also suffixes.
- Many languages use exclusively suffixes and no prefixes
- Very few languages use only prefixes and no suffixes
Internal structure: ambiguity

*dog*

Source of photo: commons.wikimedia.org/w/index.php?curid=73851814

**Morpheme ambiguity**: stems and affixes may be individually ambiguous.

*dogs*
Internal structure: ambiguity

*Structural ambiguity*: different combinations of morphemes

- **unlockable**
  - un-
  - lock

  Capable of being unlocked.

- **un-**
  - lock
  - -able

  Not capable of being locked.

*Cross word boundaries*: syntax all the way down

- **beautiful dancer**
  - beautiful
  - dance
  - -er


  More about *unlockable*: en.wiktionary.org/wiki/unlockable

http://semantics.uchicago.edu/kennedy/classes/f11/na/docs/larson08.pdf
Using Morphology in NLP
Abstraction

Surface form $\rightarrow$ Abstraction

- Indefinite article: *an* orange, *a* building
- Negation: *un*happy, *in*complete, *im*possible, *ir*rational
- Irregular: *sing*, *sang*, *sung*

The same morpheme may have different variants, which are called *allomorphs*. Allomorphs have the same function but different forms.
Computational tasks

- **Lemmatization** → word *saw*
- **Tagging** → contextualized word *saw @ J saw M*
- **Segmentation** → word *meaningful*
- **Generation** ← word *saw*

A lexicon is being compiled from natural language expressions. For example, the word *saw* is remapped to its lemma *see* and its past tense tag *VERB.PAST*.

---

Compiling a full-form lexicon, stemming for Information Retrieval, preprocessing for parsing, . . .
Segmentation

antidisestablishmentarianism ⇒ anti- dis- e- stabl -ish -ment -arian -ism
antidisestablishmentarianism

en.wikipedia.org/wiki/Antidisestablishmentarianism
www.etymonline.com/word/antidisestablishmentarianism

bioinformatics
Text normalization

- Not using any punctuation at all
  
  *Eh speak english mi malay not tt good* (Eh, speak English! My Ma-ly is not that good.)

- Using spell-ing/punctuation for emphasis
  
  *gooollllllld Sunday morning !!!!!!!* (Good Sunday morning!)

- Using phonetic spelling
  
  *dat iz enuf* (That is enough)

- Dropping vowel
  
  *i hv cm to c my luv.* (I have come to see my love.)

- Introducing local flavor
  
  *yar lor where u go juz now* (yes, where did you go just now?)

- Dropping verb
  
  *l hv 2 go. Dinner w parents.* (I have to go. Have dinner with parents.)


More: noisy-text.github.io/norm-shared-task.html
Aspects of Morphological Processing
Cross-lingual variants

- **English** morphology is essentially concatenative
cf. duplication in **Chinese**, e.g.

```
高兴 → 高高高兴兴

happy
```

---

<table>
<thead>
<tr>
<th>Form</th>
<th>Root Pattern</th>
<th>PoS</th>
<th>Phonological Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>ktb</td>
<td>C</td>
<td>C</td>
<td>Cv</td>
</tr>
<tr>
<td>ktbhi</td>
<td>CC</td>
<td>i</td>
<td>Cv</td>
</tr>
<tr>
<td>ktbmi</td>
<td>CC</td>
<td>a</td>
<td>Cn</td>
</tr>
<tr>
<td>ktb</td>
<td>CC</td>
<td>a</td>
<td>Cn</td>
</tr>
</tbody>
</table>

from E. Bender's tutorial ([faculty.washington.edu/ebender/papers/100things.pdf](faculty.washington.edu/ebender/papers/100things.pdf))
Cross-lingual variants

- **English** morphology is essentially concatenative
cf. duplication in **Chinese**, e.g.

  高 兴 → 高 高 兴 兴

  *happy*

- The phones making up a morpheme don’t have to be contiguous, e.g. in **Hebrew**,

<table>
<thead>
<tr>
<th>Root</th>
<th>Pattern</th>
<th>PoS</th>
<th>Phonological Form</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>ktb</td>
<td>CaCaC</td>
<td>v</td>
<td>katav</td>
<td>‘wrote’</td>
</tr>
<tr>
<td>ktb</td>
<td>hiCCiC</td>
<td>v</td>
<td>hixtiv</td>
<td>‘dictated’</td>
</tr>
<tr>
<td>ktb</td>
<td>miCCaC</td>
<td>n</td>
<td>mixtav</td>
<td>‘a letter’</td>
</tr>
<tr>
<td>ktb</td>
<td>CCaC</td>
<td>n</td>
<td>ktav</td>
<td>‘writing, alphabet’</td>
</tr>
</tbody>
</table>

from E. Bender’s tutorial ([faculty.washington.edu/ebender/papers/100things.pdf](faculty.washington.edu/ebender/papers/100things.pdf))
Spelling rules

- 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)

morphophonology

- In English, description is independent of particular stems/affixes
Knowledge

affixes, plus the associated information conveyed by the affix

- ed VERB.PAST
- ed VERB.PSP
- s NOUN.PLURAL

irregular forms, with associated information similar to that for affixes

began VERB.PAST begin
begun VERB.PSP begin
Finite State Techniques
- Circles are states of the automaton.
- Arrows are called transitions.
- The automaton changes states by following transitions.
- The double circle indicates that this state is an accepting state. The automaton accepts the string if it ends in an accepting state.
Circles are **states** of the automaton.

Arrows are called **transitions**.

The automaton changes states by following transitions.

The double circle indicates that this state is an **accepting state**. The automaton accepts the string if it ends in an accepting state.

**Form Transformation**: augmenting transitions

\[ \text{input} \rightarrow \text{input:output} \]
Finite state transducer

- cakes $\rightarrow$ cake#s
- boxes $\rightarrow$ box#s
Analysing boxes

OUTPUT
INPUT

| b | o | x | e | s |

```
q1 - e:e
    ^
   /   other:other
q4 - e:e
    ^
   /   other:other

q1 - e: #
q2 - s:s
q3 - s:s

q4 - s:s
    | x:x
    | z:z
```

21 of 23
Analysing boxes

OUTPUT

INPUT

| b | o | x | e | s |

\[ q_1 \] \quad \epsilon: \# \quad \Rightarrow \quad \frac{e:e}{\text{other:other}} \quad \Rightarrow \quad q_2

\[ q_2 \] \quad s:s \quad \Rightarrow \quad q_3

\[ q_4 \] \quad \frac{e:e}{\text{other:other}} \quad \Rightarrow \quad \frac{s:s}{x:x} \quad \frac{z:z}{\epsilon: \#} \quad \Rightarrow \quad q_1
Analysing boxes

INPUT

OUTPUT

b

boxes

\[ q_1 \quad \epsilon: \# \quad q_2 \quad s:s \quad q_3 \]

\[ q_4 \quad s:s \quad x:x \quad z:z \quad \epsilon: \# \]

\[ q_1 \quad \epsilon: e \quad other: other \]

\[ q_2 \quad s:s \quad \epsilon: \# \]

\[ q_3 \quad s:s \quad \epsilon: \# \]

\[ q_4 \quad s:s \quad x:x \quad z:z \]
Analysing boxes

INPUT

OUTPUT

| b | o | x | e | s |

\[
\begin{array}{c}
q_1 \\
q_2 \\
q_3 \\
q_4
\end{array}
\]

\[
\begin{array}{c}
e : e \\
other : other
\end{array}
\]

\[
\begin{array}{c}
s : s \\
x : x \\
z : z
\end{array}
\]

\[
\begin{array}{c}
\epsilon : \# \\
s : s
\end{array}
\]
Analysing *boxes*
Analysing boxes

OUTPUT

INPUT

| b | o | x | e | s |

\[
\begin{array}{c}
q_1 \xrightarrow{e,e} q_4 \xrightarrow{s,s} q_2 \\
q_4 \xrightarrow{s,s} q_3
\end{array}
\]
Analysing boxes
Analysing boxes

OUTPUT
INPUT

\begin{array}{cccc}
\text{b} & \text{o} & \text{x} & \# & \text{s} \\
\text{b} & \text{o} & \text{x} & \text{e} & \text{s} \\
\end{array}

\begin{tikzpicture}
\node[state] (q1) {$q_1$};
\node[state] (q2) [right of=q1] {$q_2$};
\node[state] (q3) [right of=q2] {$q_3$};
\node[state] (q4) [below of=q1] {$q_4$};

\draw[->] (q1) -- node {$\epsilon:\#$} (q2);
\draw[->] (q2) -- node {$s:s$} (q3);
\draw[->] (q1) -- node {$e:e$} (q4);
\draw[->] (q4) -- node {$s:s$} (q1);
\draw[->] (q4) -- node {$e:e$} (q2);
\draw[->] (q4) -- node {$s:s$} (q3);
\draw[->] (q4) -- node {$x:x$} (q1);
\draw[->] (q4) -- node {$z:z$} (q2);
\draw[->] (q4) -- node {$e:\#$} (q3);
\end{tikzpicture}
Finite-state machine

- A symbolic system that can recognize or transform forms.
A symbolic system that can recognize or transform forms.

An automaton remembers only a finite amount of information.
Finite-state machine

- A symbolic system that can recognize or transform forms.
- An automaton remembers only a finite amount of information.
- Information is represented by its states.
Finite-state machine

- A symbolic system that can recognize or **transform forms**.
- An automaton remembers only a finite amount of information.
- Information is represented by its states.
- State changes in response to inputs and may trigger outputs.
A symbolic system that can recognize or **transform forms**.

- An automaton remembers only a finite amount of information.
- Information is represented by its states.
- State changes in response to inputs and may trigger outputs.
- **Transition rules define how the state changes in response to inputs.**
Finite-state machine

- A symbolic system that can recognize or transform forms.
- An automaton remembers only a finite amount of information.
- Information is represented by its states.
- State changes in response to inputs and may trigger outputs.
- Transition rules define how the state changes in response to inputs.
- Given a sequence of input symbols, a recognition process starts in the start state and follow the transitions in turn. Input is accepted if this process ends up in an accepting state.
Finite-state machine

- A symbolic system that can recognize or transform forms.
- An automaton remembers only a finite amount of information.
- Information is represented by its states.
- State changes in response to inputs and may trigger outputs.
- Transition rules define how the state changes in response to inputs.
- Given a sequence of input symbols, a recognition process starts in the start state and follow the transitions in turn. Input is accepted if this process ends up in an accepting state.
- Partial grammars for text preprocessing, tokenization, named entity recognition etc.
Readings

Required

- Ann’s lecture notes.
  https://www.cl.cam.ac.uk/teaching/1920/NLP/materials.html
- E. Bender. 100 Things You Always Wanted to Know about Linguistics But Were Afraid to Ask. NAACL-HLT 2012 tutorial.
  faculty.washington.edu/ebender/papers/100things.pdf

Optional