Morphology

11-711 Algorithms for NLP
1 November 2018 – Part I

(Some slides from Lori Levin, David Mortenson)
Types of Lexical and Morphological Processing

• **Tokenization**
  • Input: raw text
  • Output: sequence of tokens normalized for further processing

• **Recognition**
  • Input: a string of characters
  • Output: is it a legal word? (yes or no)

• **Morphological Parsing**
  • Input: a word
  • Output: an analysis of the structure of the word

• **Morphological Generation**
  • Input: an analysis of the structure of the word
  • Output: a word
But first: What is a word?

• The things that are in the dictionary?
  • But how did the lexicographers decide what to put in the dictionary?
• The things between spaces and punctuation?
• The smallest unit that can be uttered in isolation?
  • You could say this word in isolation: *Unimpressively*
  • This one too: *impress*
  • But you probably wouldn’t say these in isolation, unless you were talking about morphology:
    • *un*
    • *ive*
    • *ly*
So what is a word?

- Can get pretty tricky:
  - didn’t
  - would’ve
  - gonna
  - shoulda woulda coulda
  - Ima
  - blackboard (vs. school board)
  - baseball (vs. golf ball)
  - the person who left’s hat; Jim and Gregg’s apartment
  - acct.
  - LTI
About 1000 pages. $139.99

You don’t have to read it.

The point is that it takes 1000 pages just to survey the issues related to what words are.
So what is a word?

• It is up to you or the software you use for processing words.
• Take linguistics classes.
• Make good decisions in software design and engineering.
Tokenization
Tokenization

*Input*: raw text

*Output*: sequence of **tokens** normalized for easier processing.
• Some Asian languages have obvious issues:

利比亚“全国过渡委员会”执行委员会主席凯卜22日在首都的黎波里公布“过渡政府”内阁名单，宣告过渡政府正式成立。
Tokenization

• Some Asian languages have obvious issues:

利比亚“全国过渡委员会”执行委员会主席凯卜22日在首都的黎波里公布“过渡政府”内阁名单，宣告过渡政府正式成立。

• But German too: Noun-noun compounds:

_Gesundheitsversicherungsgesellschaften_
Tokenization

• Some Asian languages have obvious issues:
  利比亚“全国过渡委员会”执行委员会主席凯卜22日在首都的黎波里公布“过渡政府”内阁名单，宣告过渡政府正式成立。

• But German too: Noun-noun compounds:
  *Gesundheits-versicherungs-gesellschaften (health insurance companies)*
Tokenization

• Some Asian languages have obvious issues:
  利比亚“全国过渡委员会”执行委员会主席凯卜22日在首都的黎波里公布“过渡政府”内阁名单，宣告过渡政府正式成立。

• But German too: Noun-noun compounds:
  Gesundheitsversicherungsgesellschaften

• Spanish clitics: Darmelo
• Some Asian languages have obvious issues:
  利比亚“全国过渡委员会”执行委员会主席凯卜22日在首都的黎波里公布“过渡政府”内阁名单，宣告过渡政府正式成立。

• But German too: Noun-noun compounds:
  *Gesundheitsversicherungsgesellschaften*

• Spanish clitics: *Dar-me-lo (To give me it)*
Tokenization

• Some Asian languages have obvious issues:
  利比亚“全国过渡委员会”执行委员会主席凯卜22日在首都的黎波里公布“过渡政府”内阁名单，宣告过渡政府正式成立。

• But German too: Noun-noun compounds:
  Gesundheitsversicherungsgesellschaften

• Spanish clitics: Darmelo

• Even English has issues, to a smaller degree: Gregg and Bob’s house
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.


```plaintext
Dr./NNP Smith/NNP said/VBD tokenization/NN of/IN English/NNP is/VBZ ``/`` harder/JJR than/IN you/PRP 've/VBP thought/VBN ./. 
''/

When/WRB in/IN New/NNP York/NNP ,/, he/PRP paid/VBD $/$ 12.00/CD a/DT day/NN for/IN lunch/NN and/CC wondered/VBD what/WP it/PRP would/MD be/VB like/JJ to/TO work/VB for/IN AT&T/NNP or/CC Google/NNP ,/, Inc./NNP ./.
```
Morphological Phenomena
What is Linguistic Morphology?

• Morphology is the study of the internal structure of words.

  • **Derivational morphology.** How new words are created from existing words.
    • [grace]
    • [[grace]ful]
    • [un[grace]ful]]

  • **Inflectional morphology.** How features relevant to the syntactic context of a word are marked on that word.
    • This example illustrates number (singular and plural) and tense (present and past).
    • Green indicates irregular. Blue indicates zero marking of inflection. Red indicates regular inflection.
    • This student walks.
    • These students walk.
    • These students walked.

• **Compounding.** Creating new words by combining existing words
  • With or without spaces: surfboard, golf ball, blackboard
Morphemes

- **Morphemes.** Minimal pairings of form and meaning.

- **Roots.** The “core” of a word that carries its basic meaning.
  - *apple* : ‘apple’
  - *walk* : ‘walk’

- **Affixes (prefixes, suffixes, infixes, and circumfixes).** Morphemes that are added to a base (a root or stem) to perform either derivational or inflectional functions.
  - *un-* : ‘NEG’
  - *-s* : ‘PLURAL’
Language Typology
Types of Languages:

• In order of morphological complexity:
  • Isolating (or Analytic)
  • Fusional (or Inflecting)
  • Agglutinative
  • Polysynthetic
  • Others
Isolating Languages: Chinese
Little morphology other than compounding

- **Chinese inflection**
  - few affixes (prefixes and suffixes):
    - 们: 我们, 你们, 他们, 。。。同志们
      - plural: we, you (pl.), they comrades, LGBT people
    - “suffixes” that mark aspect: 着 -zhě ‘continuous aspect’

- **Chinese derivation**
  - 艺术家 yìshùjiā ‘artist’

- Chinese is a champion in the realm of compounding—up to 80% of Chinese words are actually compounds.

<table>
<thead>
<tr>
<th>毒</th>
<th>+</th>
<th>贩</th>
<th>→</th>
<th>毒贩</th>
</tr>
</thead>
<tbody>
<tr>
<td>dù</td>
<td>fàn</td>
<td>→</td>
<td>dùfàn</td>
<td></td>
</tr>
<tr>
<td>‘poison, drug’</td>
<td>‘vendor’</td>
<td>‘drug trafficker’</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Agglutinative Languages: Swahili

Verbs in Swahili have an average of 4-5 morphemes, [http://wals.info/valuesets/22A-swa](http://wals.info/valuesets/22A-swa)

<table>
<thead>
<tr>
<th>Swahili</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>m-tu a-li-lala</em></td>
<td>‘The person slept’</td>
</tr>
<tr>
<td><em>m-tu a-ta-lala</em></td>
<td>‘The person will sleep’</td>
</tr>
<tr>
<td><em>wa-tu wa-li-lala</em></td>
<td>‘The people slept’</td>
</tr>
<tr>
<td><em>wa-tu wa-ta-lala</em></td>
<td>‘The people will sleep’</td>
</tr>
</tbody>
</table>

- Words written without hyphens or spaces between morphemes.
- Orange prefixes mark noun class (like gender, except Swahili has nine instead of two or three).
  - Verbs agree with nouns in noun class.
  - Adjectives also agree with nouns.
  - Very helpful in parsing.
- Black prefixes indicate tense.
Turkish
Example of extreme agglutination
But most Turkish words have around three morphemes

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 ablative case (“from/among”)
+mış past
+sınız second person plural (“y’ all”)
+casına finite verb → adverb (“as if”)

Behaving: uygarlaştıramadıklarımızdanmışsınızcasına
 Chunga!
Operationalization

- operate (opus/opera + ate)
- ion
- al
- ize
- ate
- ion
Polysynthetic Languages: Yupik

• Polysynthetic morphologies allow the creation of full “sentences” by morphological means.
• They often allow the incorporation of nouns into verbs.
• They may also have affixes that attach to verbs and take the place of nouns.

• Yupik Eskimo
  *Untu-ssur-qatar-ni-ksaite-ngqiggte-uq*
  reindeer-hunt-FUT-say-NEG-again-3SG.INDIC
  ‘He had not yet said again that he was going to hunt reindeer.’
## Fusional Languages: Spanish

<table>
<thead>
<tr>
<th>Tense</th>
<th>Singular</th>
<th>Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st</td>
<td>2nd</td>
</tr>
<tr>
<td>Present</td>
<td><em>am-o</em></td>
<td><em>am-as</em></td>
</tr>
<tr>
<td>Imperfect</td>
<td><em>am-ab-a</em></td>
<td><em>am-ab-as</em></td>
</tr>
<tr>
<td>Preterit</td>
<td><em>am-é</em></td>
<td><em>am-aste</em></td>
</tr>
<tr>
<td>Future</td>
<td><em>am-aré</em></td>
<td><em>am-arás</em></td>
</tr>
<tr>
<td>Conditional</td>
<td><em>am-aría</em></td>
<td><em>am-arias</em></td>
</tr>
</tbody>
</table>
Indo-European: 4000BC

From Wikipedia
Indo-European: 3000BC
Indo-European: 2000BC
Indo-European: 500BC
Indo-European: “hand”
A Brief History of English

• 900,000 BC? Humans invade British Isles
• 800 BC? Celts invade (Gaelic) [first Indo-Europeans there]
• 40 AD Romans invade (Latin)
• 410 AD Anglo-Saxons invade (West German)
• 790 AD Vikings invade (North German)
• 1066 AD Normans invade (Norman French/Latin)

• The English spend a few hundred years invading rest of British Isles
• A little later, British start invading everyone else
  • North America, India, China, ...
Root-and-Pattern Morphology: Arabic

• **Root-and-pattern.** A special kind of fusional morphology found in Arabic, Hebrew, and their cousins.

• Root usually consists of a sequence of consonants.

• Words are derived and, to some extent, inflected by patterns of vowels intercalated among the root consonants.
  
  • kitaab ‘book’
  • kaatib ‘writer; writing’
  • maktab ‘office; desk’
  • maktaba ‘library’
Other Non-Concatenative Morphological Processes

**Non-concatenative morphology** involves operations other than the concatenation of affixes with bases.

- **Infixation.** A morpheme is inserted inside another morpheme instead of before or after it.

- **Reduplication.** Can be prefixing, suffixing, and even infixing.
  - **Tagalog:**
    - sulat (write, imperative)
    - susulat (reduplication) (write, future)
    - sumulat (infixing) (write, past)
    - sumusulat (infixing and reduplication) (write, present)

- **Apophony,** including the umlaut in English *tooth* → *teeth; subtractive morphology,* including the truncation in English nickname formation (*David* → *Dave*); and so on.

- **Tone change; stress shift.** And more...
Type-Token Curves

Finnish is agglutinative
Iñupiaq is polysynthetic

Types and Tokens:
“I like to walk. I am walking now. I took a long walk earlier too.”

The type walk occurs twice. So there are two tokens of the type walk.

Walking is a different type that occurs once.
Morphological Processing
Recognizing the words of a language

• Input: a string (from some alphabet)
• Output: is it a legal word? (yes or no)
FSA for English Noun inflections

Lexicon:

<table>
<thead>
<tr>
<th>reg-noun</th>
<th>irreg-pl-noun</th>
<th>irreg-sg-noun</th>
<th>plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>fox</td>
<td>geese</td>
<td>goose</td>
<td>-s</td>
</tr>
<tr>
<td>cat</td>
<td>sheep</td>
<td>sheep</td>
<td></td>
</tr>
<tr>
<td>aardvark</td>
<td>mice</td>
<td>mouse</td>
<td></td>
</tr>
</tbody>
</table>

Note: “fox” becomes plural by adding “es” not “s”. We will get to that later.
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
- $\Sigma$: a finite alphabet
- Transitions:
  
  - Encodes a \textbf{set} of strings that can be recognized by following paths from $q_0$ to some state in $F$. 

\[ q_i \quad \xrightarrow{s \in \Sigma^*} \quad q_j \]
FSA for English Adjective derivations

Big, bigger, biggest
Happy, happier, happiest, happily
Unhappy, unhappier, unhappiest, unhappily
Clear, clearer, clearest, clearly
Unclear, unclearly

Cool, cooler, coolest, coolly
Red, redder, reddest
Real, unreal, really

But note that this accepts words like “unbig”.

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

How big do these automata get? Reasonable coverage of a language takes an expert about two to four months.

What does it take to be an expert? Study linguistics to get used to all the common and not-so-common things that happen, and then practice.
Morphological Parsing

*Input*: a word

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

*Example*:  
- geese $\rightarrow$ goose $+N$ $+\text{Pl}$
- gooses $\rightarrow$ goose $+V$ $+3\text{P}$ $+\text{Sg}$
- dog $\rightarrow$ $\{\text{dog} +N +\text{Sg}, \text{dog} +V\}$
- leaves $\rightarrow$ $\{\text{leaf} +N +\text{Pl}, \text{leave} +V +3\text{P} +\text{Sg}\}$
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
• $\Sigma$ and $\Delta$: two finite alphabets
• Transitions:

\[ q_i \xrightarrow{s : t} q_j \]

$s \in \Sigma^*$ and $t \in \Delta^*$
Two-level Morphology

upper side or underlying form

talk+Past

FST

talked

lower side or surface form
Morphological Parsing with FSTs

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

Note “same symbol” shorthand.

^ denotes a morpheme boundary.

# denotes a word boundary.
Getting back to fox+s = foxes

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

Generate a normally spelled word from an abstract representation of the morphemes:

Input: fox^s# (fox^es#)
Output: foxes# (foxes#)

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 #:e, or by postprocessing the output to remove word boundaries.
The E Insertion Rule as a FST

Input: foxes# (fox$\varepsilon$es#)
Output: fox$^s$# (fox$^\varepsilon$es#)

---

Parse a normally spelled word into an abstract representation of the morphemes:

Input: foxes# (fox$\varepsilon$es#)
Output: fox$^s$# (fox$^\varepsilon$es#)

---

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 #:e, or by postprocessing the output to remove word boundaries.
Combining FSTs

Figure 3.19 Generating or parsing with FST lexicon and rules
FST Operations

Input: fox +N +pl
Output: foxes#
Language Type Comparison wrt FSTs

• Morphologies of all types can be analyzed using finite state methods.
• Some present more challenges than others:
  • **Analytic languages.** Trivial, since there is little or no morphology (other than compounding).
  • **Agglutinating languages.** Straightforward—finite state morphology was “made” for languages like this.
  • **Polysynthetic languages.** Similar to agglutinating languages, but with blurred lines between morphology and syntax.
  • **Fusional languages.** Easy enough to analyze using finite state method as long as one allows “morphemes” to have lots of simultaneous meanings and one is willing to employ some additional tricks.
  • **Root-and-pattern languages.** Require some very clever tricks.
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*
The Good News

• More than almost any other problem in computational linguistics, morphology is a solved problem (as long as you can afford to write rules by hand).

• Finite state methods provide a simple and powerful means of generating and analyzing words (as well as the phonological alternations that accompany word formation/inflection).

• Finite state morphology is one of the great successes of natural language processing.

• One brilliant aspect of using FSTs for morphology: the same code can handle both analysis and generation.
Conclusion

• Finite state methods provide a simple and powerful means of generating and analyzing words (as well as the phonological alternations that accompany word formation/inflection).

• Straightforward concatenative morphology is easy to implement using finite state methods.

• Other phenomena are easiest to capture with extensions to the finite state paradigm.
  • Co-occurrence restrictions—flag diacritics.
  • Non-concatenative morphology—compile-replace algorithm. Pure finite state, but computed in a novel fashion.
Tools

• There are special finite state toolkits for building morphological tools (and other linguistic tools).

• The best-known of these is the Xerox Finite State Tool or XFST, which originated at Xerox PARC.

• There are open source reimplementations of XFST called HFST (Helsinki Finite State Technology) and Foma, which are not as fully optimized as XFST but which are sometimes more pleasant to use.

• None of these tools allow the construction of weighted FSTs.
Can you make a list of all the words in a language?

Productivity

In the Oxford English Dictionary (OED)

(www.oed.com, accessible for free from CMU machines)

- drinkable
- visitable

Not in the OED

- mous(e)able
- stapl(e)able

In NLP, you need to be able to process words that are not in the dictionary.

But could you make a list of all possible words, taking productivity into account?
Can you make a list of all the words in a language?

A trie representing a list of words (lexicon)
Telugu, Tamil, Kannada, Malayalam
Dravidian languages

• Agglutinating like Turkish, Finnish, and Swahili
Hindi, Urdu, Bengali, Marathi, Punjabi, etc.

Indo-european

• A little richer than English
• Like English, uses auxiliary verbs and separate words to express things that are affixes on the verbs in Dravidian languages.
  • want, have, be, make, etc.
Mapudungun compared to Spanish
Mapudungun is polysynthetic
Spanish is fusional