CS 181: Natural Language Processing

Lecture 3: Morphology & Tagging

Kim Bruce
Pomona College
Spring 2008

Disclaimer: Slide contents borrowed from many sources on web!

Last nltk examples

```
>>> print nltk.corpus.treebank.parsed_sents('wsj_0001')[0]
(S
  (NP-SBJ
    (NP (NNP Pierre) (NNP Vinken))
    (, ,)
    (ADJP (NP (CD 61) (NNS years)) (JJ old))
    (, ,))
  (VP
    (MD will)
    (VP
      (VB join)
      (NP (DT the) (NN board))
      (PP-CLR (IN as) (NP (DT a) (JJ nonexecutive) (NN director)))
      (NP-TMP (NNP Nov.) (CD 29))))
  (. .))
```

Reading Files

- f = open('afile','rU') # ignores diff in \n
- f.readline() # returns one line (including \n) at a time.
- contents = f.read() # reads whole file into string

Morphology

Morphology

- Study of sub-word units of meaning
  - Ex: disconnect - "not" + "attach"
- Construct plurals:
  - regular: add s
  - word ends in y: change y to i and add es
  - word ends in x or ch: add es
  - ...
- Breaking word ("churches") into morphemes "church" and "es" called morphological parsing

More Morphology

- Morpheme: minimal meaning-bearing unit
  - Stem: main morpheme, e.g., church
  - Affixes: add "additional" meanings, e.g. es, un, anti, ize, ization, ...
  - not always concatenative - add in middle or use other templates
- Inflection: stem + morpheme in same class
  - usually for agreement
- Derivation: stem + morpheme in diff class
  - e.g., add "ly" or "ize"
- Lemma: Set of lexical forms with same stem.

Regular Verb Forms

Morphological classes				
stem	walk	merge	try	map
-s form	walks	merges	tries	maps
-ing participle	walking	merging	trying	mapping
Past form	walked	merged	tried	mapped

Irregular verbs

Morphological classes			
stem	eat	catch	cut
-s form	eats	catches	cuts
-ing participle	eating	catching	cutting
past form	ate	caught	cut
-ed/-en participle	eaten	caught	cut

Two Goals

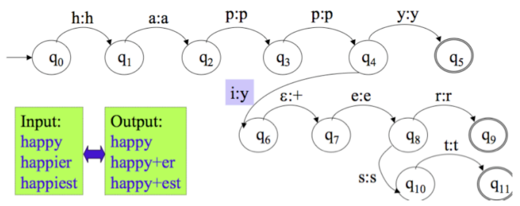
- Recognize word as legal or not, and return lexical form: lemma + tags:
  - dogs ⇒ dog + N + PL
  - children ⇒ child + N + PL
- Generate correct surface form from lemma plus tags *(reverse of above)*

Finite-State Transducers

Finite State Transducers

- Generalize FSA's to generate output
- Finite state transducer: 2 tape automaton that generates output on second tape while reading input on first.
- Ideally, can run either direction
- Want to cascade simple transformations into more complex ones.

Example Transducer

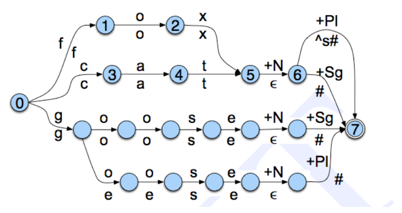


- Translates surface form of word to morphemes.
- Lexicon can be encoded as a FST

Translating Between Forms

- Use "^" as morpheme boundary, "#" as word boundary.
- Three forms:
  - Lexical: church+N+PL
  - Intermediate: church^s#
  - Surface: churches#

Example From Text

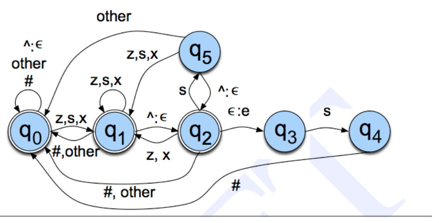


Top is lexical form
Bottom is corresponding intermediate form
Not deterministic! Not nec. equiv. to DFST.
May have to search!

Rules for Plurals

- Add an "e" after "s", "z", "x", "ch", or "sh" before adding "-s"
- "y" changes to "ie" before "-s"

Converting Plurals



Intermediate to Surface

Cascasade FST's

- Compose Lexical ⇔ Intermediate and Intermediate ⇔ Surface FST's to get Lexical ⇔ Surface
- Can also intersect FST's
- Ambiguity a problem: Is "dogs"
  - dog + N + Pl or
  - dog + V + 3Sg
- Return both and decide later based on contextual information

Porter Stemmer

- First approach depended on lexicon plus rules.
- Approaches just based on rules.
- Stemmers return base form of word.
- Porter (1980) is a good one, though improvements possible.

Porter Stemmer

- Rules depend on "measure" of the stem:
  - consonant is any but a, e, i, o, u, or y preceded by consonant.
  - C represents one or more consonants
  - V represents one or more vowels
  - Write words uniquely as [C](VC)^m[V] where [...] means optional.
  - Measure is m.
- Examples: by (0), tree (0), trees(1), private(2), trouble(1), troubles(2)
- Other conditions, e.g., (*v*) = contains vowel

Porter Stemmer Rules

- Seven sets of rewrite rules of form (cond) S1 ⇒ S2 to get stem:

1. Plural nouns & 3Sg V:

SSES ⇒ SS	*asses ⇒ ass*
IES ⇒ I	*ponies ⇒ poni*
SS ⇒ SS	*assess ⇒ assess*
S ⇒ ϵ	*dogs ⇒ dog*

2. a. Verbal Past tense & Progressive forms

(m > 1) EED ⇒ EE	*feed ⇒ feed, agreed ⇒ agree*
(*v*) ED ⇒ ϵ	*planted ⇒ plant, fed ⇒ fed*
(*v*) ING ⇒ ϵ	*telling ⇒ tell, ring ⇒ ring*

Porter Stemmer Rules

- Seven sets of rewrite rules not always successful
  - *organization ⇒ organ*
  - *doing ⇒ doe*
  - *analysis ⇒ analysi*
  - *matrices ⇒ matric*
  - *matrix ⇒ matrix*

Stemmer in NLTK

```
import nltk

stemmer = nltk.PorterStemmer()

words = ['assess','ass','assesses','analyze','analysis']
stems = []
for word in words:
   stem_word = stemmer.stem(word)
   stems.append(stem_word)

print sorted(set(stems))
```

Any Questions?