Stemming

- Words can be viewed as consisting of:
  - A STEM
  - One or more AFFIXes
- MORPHOLOGICAL ANALYSIS in its general form involves recovering the LEMMA of a word an all its affixes, together with their grammatical properties
- STEMMING a simplified form of morphological analysis – simply find the stem

The Porter Stemmer (Porter, 1980)

- A simple rule-based algorithm for stemming
- An example of a HEURISTIC method
- Based on rules like:
  - ATIONAL -> ATE (e.g., *relational* -> *relate*)
- The algorithm consists of seven sets of rules, applied in order

The Porter Stemmer: definitions

- Definitions:
  - CONSONANT: a letter other than A, E, I, O, U, and Y preceded by consonant
  - VOWEL: any other letter
- With this definition, all words are of the form:

  $(C)(VC)^m(V)$

  C=string of one or more consonants (con+)

  V=string of one or more vowels
- E.g.,
  - Troubles
  - C V CVC

The Porter Stemmer: rule format

- The rules are of the form:

  (condition) S1 -> S2

  Where S1 and S2 are suffixes

- Conditions:

m	The measure of the stem
*S	The stem ends with S
v	The stem contains a vowel
*d	The stem ends with a double consonant
*o	The stem ends in CVC (second C not W, X, or Y)

The Porter Stemmer: Step 1

- SSES -> SS
  - *caresses -> caress*
- IES -> I
  - *ponies -> poni*
  - *ties -> ti*
- SS -> SS
  - *caress -> caress*
- S -> ε
  - *cats -> cat*

The Porter Stemmer: Step 2a (past tense, progressive)

- (m>1) EED -> EE
  - Condition verified: *agreed -> agree*
  - Condition not verified: *feed -> feed*
- (*V*) ED -> ϵ
  - Condition verified: *plastered -> plaster*
  - Condition not verified: *bled -> bled*
- (*V*) ING -> ϵ
  - Condition verified: *motoring -> motor*
  - Condition not verified: *sing -> sing*

The Porter Stemmer: Step 2b (cleanup)

- (These rules are ran if second or third rule in 2a apply)
- AT-> ATE
  - *conflat(ed) -> conflate*
- BL -> BLE
  - *Troubl(ing) -> trouble*
- (*d & ! (*L or *S or *Z)) -> single letter
  - Condition verified: *hopp(ing) -> hop, tann(ed) -> tan*
  - Condition not verified: *fall(ing) -> fall*
- (m=1 & *o) -> E
  - Condition verified: *fil(ing) -> file*
  - Condition not verified: *fail -> fail*

The Porter Stemmer: Steps 3 and 4

- Step 3: Y Elimination (*V*) Y -> I
  - Condition verified: *happy -> happi*
  - Condition not verified: *sky -> sky*
- Step 4: Derivational Morphology, I
  - (m>0) ATIONAL -> ATE
    - *Relational -> relate*
  - (m>0) IZATION -> IZE
    - *generalization-> generalize*
  - (m>0) BILITI -> BLE
    - *sensibiliti -> sensible*

The Porter Stemmer: Steps 5 and 6

- Step 5: Derivational Morphology, II
  - (m>0) ICATE -> IC
    - *triplicate -> triplic*
  - (m>0) FUL -> ε
    - *hopeful -> hope*
  - (m>0) NESS -> ε
    - *goodness -> good*
- Step 6: Derivational Morphology, III
  - (m>0) ANCE -> ε
    - *allowance-> allow*
  - (m>0) ENT -> ε
    - *dependent-> depend*
  - (m>0) IVE -> ε
    - *effective -> effect*

The Porter Stemmer: Step 7 (cleanup)

- Step 7a
  - (m>1) E -> ε
    - *probate -> probat*
  - (m=1 & !*o) NESS -> ε
    - *goodness -> good*
- Step 7b
  - (m>1 & *d & *L) -> single letter
    - Condition verified: *controll -> control*
    - Condition not verified: *roll -> roll*

Examples

- *computers*
  - Step 1, Rule 4: -> *computer*
  - Step 6, Rule 4: -> *compute*
- *controlling*
  - Step 2a, Rule 3: -> *controll*
  - Step 7b : -> *control*
- *generalizations*
  - Step 1, Rule 4: -> *generalization*
  - Step 4, Rule 11: -> *generalize*
  - Step 6, last rule: -> *general*

Problems

- *elephants -> eleph*
  - Step 1, Rule 4: -> *elephant*
  - Step 6, Rule 7: -> *eleph*
- *doing - > doe*
  - Step 2a, Rule 3: -> *do*

References

- The Porter Stemmer home page (with the original paper and code): http://www.tartarus.org/~martin/PorterStemmer/
- Jurafsky and Martin, chapter 3.4
- The original paper: Porter, M.F., 1980, An algorithm for suffix stripping, *Program*, **14**(3) :130-137.