Tokenization

Tokenization

- Given input text, split into words (or sentences).
- Tokens: words, numbers, punctuation marks
- Example:
  - Input: He said reaction has been “very positive.”
  - Output: He said reaction has been “ very positive . ”
- Why tokenize?
  - Identify basic units for downstream processing

Tokenization

- Proposal #1: Split on whitespace
- Good enough? No
- Why not?
  - Multi-linguality:
    - Languages without whitespace delimiters: Chinese, Japanese
    - Agglutinative languages (Hungarian, Korean)
      - meggazdagiithatnok
    - Compounding nouns (German):
      - Lebensversicherungsgesellschaftsangestellter

      “Life insurance company employee”
  - Even with English, this approach does not handle punctuation properly.

Tokenization

- Proposal #2: Split on whitespace and punctuation marks only
  - For English
- Good enough? No
- Problems: Non-splitting punctuation
  - 1.23 ➔ 1 . 23 1,234,456 ➔ 1 , 234 , 456
  - don’t ➔ don ‘ t E-mail ➔ E - mail
- Problems: no-splitting whitespace
  - Names: New York; Collocations: pick up
- What’s a word?

Tokenization in Chinese, Japanese, Thai, etc.

- No whitespace in written languages

- Baseline: maximum match
  - Use a dictionary
  - Cannot handle unknown words or ambiguity (e.g., “ABC” can be “AB C” or “A BC”)

- Current approach: treat it as a sequence labeling task
  - Input: c1c2c3c4c5c6
  - Output: c1c2 c3 c4c5c6
  - Output: c1/B c2/E c3/S c4/B c5/I c6/E
  - “BIO” scheme and its extension: B (beginning), I (inside), O (outside), E (ending), S (single)

What counts as a word?

- In theory:
  - Phonological word, syntactic words, lexeme, etc.
  - Ex: “On the definition of word” (Di Sciullo and Williams, 1987)

- In practice:
  - $22
  - Hyphenated words
  - Named entity
  - …
  - This can depend on your applications: e.g., MT (for which lang pairs)

Specification of the tokenizer

- The tokenizer should not separate \W from surrounding chars in the following cases:
  - \w matches a “word” character, i.e., [a-zA-Z0-9_]; \W matches non-word chara
  - Period: when it is part of a number, an abbreviation, an url, an email address, the ellipsis (e.g., Ph.D., 1.23, xx@uw.edu)
  - Comma: when it is part of a number (e.g., 1,245,789)
  - Colon: when it is part of an url or a path (e.g., http://washington.edu/)
  - Slash and back slash: when it is part of a fraction, an url or a path (e.g., C:\dropbox\22-23\, 3/14)
  - Hyphen: when it is used as a minus sign (e.g., -4), part of a dash (“--”), e-file
  - Tilde: part of an url or a path (e.g., ~/hw1/)
  - %: percentage sign (e.g., -2.3%)

Other special cases

- Apostrophe: isn’t ➔ is n’t, I’d ➔ I ’d, doesn’t ➔ does n’t
  fund’s ➔ fund ’s, funds’ ➔ funds’

- Dollar sign ($) is separated from the following number:
  $12.5 ➔ $ 12.5

- Abbreviation:
  E.g., U.S.A. Inc. Ph.D. e-file Mr.

Similar task: Sentence Segmentation

- Proposal: Split on period, !, ?

- Problems?
  - Non-boundary periods:
    - 1.23
    - Mr. Sherword
    - U.S.A.
    - Ph.D.
    - Etc.

- Solutions?
  - Rule-based approach: e.g., using heuristics and dictionaries.
  - Labeled data + machine learning

- What if the text is ASR output which has no punctuation marks?