Amirkabir University of Technology
(Tehran Polytechnic)

Natural Language Processing

Lecture 17: Text Preprocessing

Amirkabir University of Technology
Dr Momtazi

Variations of Corpora

- Language: 7097 languages in the world
- Variety, like African American Language varieties.
  - AAE Twitter posts might include forms like "iont" (I don't)

- Code switching, e.g., Spanish/English, Hindi/English:

  S/E: Por primera vez veo a @username actually being hateful! It was beautiful:)

  [For the first time I get to see @username actually being hateful! it was beautiful:)]

  H/E: dost tha or ra- hega ... dont wory ... but dherya rakhe

  [“he was and will remain a friend ... don’t worry ... but have faith”]

- Genre: newswire, fiction, scientific articles, Wikipedia
- Author Demographics: writer's age, gender, ethnicity, SES

Outline

- **Tokenization**
- Normalization
- Lemmatization
- Stemming
- Stopword Removal

Space-based Tokenization

- A very simple way to tokenize
  - For languages that use space characters between words
    - Arabic, Cyrillic, Greek, Latin, etc., based writing systems
  - Segment off a token between instances of spaces

- Unix tools for space-based tokenization
  - The "tr" command
  - Inspired by Ken Church's UNIX for Poets
  - Given a text file, output the word tokens and their frequencies

More Counting

- Merging upper and lower case

```
tr 'A-Z' 'a-z' < shakes.txt | tr -sc 'A-Za-z' '\n' | sort | uniq -c
```

- Sorting the counts

```
tr 'A-Z' 'a-z' < shakes.txt | tr -sc 'A-Za-z' '\n' | sort | uniq -c | sort -n -r
```

```
  23243 the
  22225 i
  18618 and
  16339 to
  15687 of
  12780 a
  12163 you
  10839 my
  10005 in
   8954 d
```

What happened here?

Issues in Tokenization

- Can't just blindly remove punctuation:
  - m.p.h., Ph.D., AT&T, cap’n
  - prices ($45.55)
  - dates (01/02/06)
  - URLs (http://www.stanford.edu)
  - hashtags (#nlproc)
  - email addresses (someone@cs.colorado.edu)
- Clitic: a word that doesn't stand on its own
  - "are" in we're, French "je" in j'ai, "le" in l'honneur
- When should multiword expressions (MWE) be words?
  - New York, rock ’n’ roll

Tokenization in NLTK

Bird, Loper and Klein (2009), *Natural Language Processing with Python*. O'Reilly

```
>>> text = 'That U.S.A. poster-print costs $12.40...'
>>> pattern = r'''(?x)    # set flag to allow verbose regexps
...     ([A-Z]\.)+        # abbreviations, e.g. U.S.A.
...   | \w+(-\w+)*        # words with optional internal hyphens
...   | \$?\d+(\.\d+)?%?  # currency and percentages, e.g. $12.40, 82%
...   | \.\.\.            # ellipsis
...   | [][.,;"'?():-_`]  # these are separate tokens; includes ], [
... '''
>>> nltk.regexp_tokenize(text, pattern)
['That', 'U.S.A.', 'poster-print', 'costs', '$12.40', '...']
```

Tokenization in Languages without Spaces

- Many languages (like Chinese, Japanese, Thai) don't use spaces to separate words!

- How do we decide where the token boundaries should be?

How to do word tokenization in Chinese?

姚明进入总决赛 “Yao Ming reaches the finals”

How to do word tokenization in Chinese?

姚明进入总决赛 “Yao Ming reaches the finals”

- 3 words?

姚明　　进入　　总决赛

YaoMing reaches finals

How to do word tokenization in Chinese?

姚明进入总决赛 “Yao Ming reaches the finals”

- 3 words?

姚明　　进入　　总决赛
YaoMing reaches finals

- 5 words?

姚　　明　　进入　　总　　决赛
Yao　Ming　reaches　overall　finals

- 7 characters? (don't use words at all):

姚　明　　进　　入　　总　　决　　赛
Yao Ming enter enter overall decision game

Sentence Segmentation

- !, ? mostly unambiguous but **period** "." is very ambiguous
  - Sentence boundary
  - Abbreviations like Inc. or Dr.
  - Numbers like .02% or 4.3

- Common algorithm: Tokenize first: use rules or ML to classify a period as either (a) part of the word or (b) a sentence-boundary.
  - An abbreviation dictionary can help

- Sentence segmentation can then often be done by rules based on this tokenization.

Outline

- Tokenization
- **Normalization**
- Lemmatization
- Stemming
- Stopword Removal

Word Normalization

- Putting words/tokens in a standard format
  - U.S.A. or USA
  - uhhuh or uh-huh
  - Fed or fed
  - am, is, be, are

Case Folding

- Applications like IR: reduce all letters to lower case
  - Since users tend to use lower case
  - Possible exception: upper case in mid-sentence?
    - e.g., ***General Motors***
    - ***Fed*** vs. ***fed***
    - ***SAIL*** vs. ***sail***

- For sentiment analysis, MT, Information extraction
  - Case is helpful (***US*** versus ***us*** is important)

Outline

- Tokenization
- Normalization
- **Lemmatization**
- Stemming
- Stopword Removal

Lemmatization

- Represent all words as their lemma, their shared root
  = dictionary headword form:
  - *am, are, is* → *be*
  - *car, cars, car's, cars'* → *car*
  - Spanish quiero (‘I want’), quieres (‘you want’)

    → querer ‘want'
  - *He is reading detective stories*

    → *He be read detective story*

Lemmatization is done by Morphological Parsing

- Morphemes:
  - The small meaningful units that make up words
  - **Stems**: The core meaning-bearing units
  - **Affixes**: Parts that adhere to stems, often with grammatical functions

- Morphological Parsers:
  - Parse *cats* into two morphemes *cat* and *s*
  - Parse Spanish *amaren* ('if in the future they would love') into morpheme *amar* 'to love', and the morphological features *3PL* and *future subjunctive*.

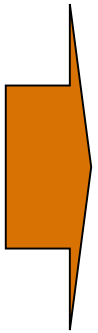
Outline

- Tokenization
- Normalization
- Lemmatization
- **Stemming**
- Stopword Removal

Stemming

- Reduce terms to stems, chopping off affixes crudely

This was not the map we found in Billy Bones's chest, but an accurate copy, complete in all things-names and heights and soundings-with the single exception of the red crosses and the written notes.



Thi wa not the map we found in Billi Bone s chest but an accur copi complet in all thing name and height and sound with the singl except of the red cross and the written note .

Porter Stemmer

- Based on a series of rewrite rules run in series
  - A cascade, in which output of each pass fed to next pass
- Some sample rules:

$$\begin{aligned} \text{ATIONAL} &\rightarrow \text{ATE} \quad \text{(e.g., relational} \rightarrow \text{relate)} \\ \text{ING} &\rightarrow \epsilon \quad \text{if stem contains vowel (e.g., motoring} \rightarrow \text{motor)} \\ \text{SSES} &\rightarrow \text{SS} \quad \text{(e.g., grasses} \rightarrow \text{grass)} \end{aligned}$$

Dealing with complex morphology

- Dealing with complex morphology is necessary for many languages
- e.g., the Turkish word:
  - Uygarlastiramadiklarimizdanmissinizcasina
  - `(behaving) as if you are among those whom we could not civilize'
  - Uygar `civilized' + las `become'
    + tir `cause' + ama `not able'
    + dik `past' + lar 'plural'
    + imiz 'p1pl' + dan 'abl'
    + mis 'past' + siniz '2pl' + casina 'as if'

Outline

- Tokenization
- Normalization
- Lemmatization
- Stemming
- **Stopword Removal**

Stopwords

- Many of the most frequently used words in English are likely to be useless for text mining
- These words are called *Stopwords*
  - Examples: the, of, and, to, an, is, that, …
  - Typically text contains about 400 to 500 such words
  - For an application, an additional domain specific stopwords list may be constructed

Stopword Removal

- Why should we remove stopwords?
  - Reduce data set size
  - Stopwords account for 20-30% of total word count
  - Improve effectivity of text mining methods
  - Stopwords may confuse the mining algorithm

- Challenges
  - *to be or not to be*

Further Reading

- Speech and Language Processing (3rd ed. draft)
  - Chapter 2