Text Normalization

Chapter 2
(2.1 – 2.4)

Basic Text Processing

Regular Expressions
Regular expressions

• A formal language for specifying text strings
• How can we search for any of these?
  • woodchuck
  • woodchucks
  • Woodchuck
  • Woodchucks
  • Ill vs. illness
  • color vs. colour

Example

• Does $grep "elect" news.txt$ return every line in a file called news.txt that contains the word “elect”
  elect
  Misses capitalized examples
  [eE]lect
  Incorrectly returns select or electives
  [^a-zA-Z][eE]lect[^a-zA-Z]
Errors

• The process we just went through was based on fixing two kinds of errors
  • Matching strings that we should not have matched (there, then, other)
    • False positives (Type I)
  • Not matching things that we should have matched (The)
    • False negatives (Type II)

Errors cont.

• In NLP we are always dealing with these kinds of errors.
• Reducing the error rate for an application often involves two antagonistic efforts:
  • Increasing accuracy or precision (minimizing false positives)
  • Increasing coverage or recall (minimizing false negatives).
Summary

• Regular expressions play a surprisingly large role
  • Sophisticated sequences of regular expressions are often the first model for any text processing text
  • I am assuming you know, or will learn, in a language of your choice
• For many hard tasks, we use machine learning classifiers
  • But regular expressions are used as features in the classifiers
  • Can be very useful in capturing generalizations

Basic Text Processing

Word tokenization
Text Normalization

- Every NLP task needs to do text normalization:
  1. Segmenting/tokenizing words in running text
  2. Normalizing word formats
  3. Segmenting sentences in running text

How many words?

- I do uh mainly business data processing
  - Fragments, filled pauses
- Terminology
  - **Lemma**: same stem, part of speech, rough word sense
    - *cat* and *cats* = same lemma
  - **Wordform**: the full inflected surface form
    - *cat* and *cats* = different wordforms
How many words?

they lay back on the San Francisco grass and looked at the stars and their

• **Type**: an element of the vocabulary.
• **Token**: an instance of that type in running text.
• How many?
  • 15 tokens (or 14)
  • 13 types (or 12) (or 11?)

---

How many words?

\[ N = \text{number of tokens} \]
\[ V = \text{vocabulary} = \text{set of types} \]

|                      | Tokens = \(N\) | Types = \(|V|\) |
|----------------------|----------------|----------------|
| Switchboard phone conversations | 2.4 million   | 20 thousand    |
| Shakespeare          | 884,000       | 31 thousand    |
| Google N-grams       | 1 trillion    | 13 million     |
Issues in Tokenization

• Finland’s capital → Finland Finlands Finland’s ?
• what’re, I’m, isn’t → What are, I am, is not
• state-of-the-art → state of the art ?
• San Francisco → one token or two?
Tokenization: language issues

- Chinese and Japanese no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - Sharapova now lives in US southeastern Florida
Normalization

- Need to “normalize” terms
  - Information Retrieval: indexed text & query terms must have same form.
    - We want to match \textit{U.S.A.} and \textit{USA}
  - We implicitly define equivalence classes of terms
    - e.g., deleting periods in a term
  - Alternative: asymmetric expansion:
    - Enter: \textit{windows}  
    Search: \textit{Windows, windows, window}
    - Potentially more powerful, but less efficient

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., \textit{General Motors}
    - \textit{Fed} vs. \textit{fed}
    - \textit{SAIL} vs. \textit{sail}
  - For sentiment analysis, MT, Information extraction
    - Case is helpful (\textit{US} versus \textit{us} is important)
Lemmatization

• Reduce inflections or variant forms to base form
  • *am, are, is* → *be*
  • *car, cars, car's, cars'* → *car*
  • *the boy's cars are different colors* → *the boy car be different color*
  • Lemmatization: have to find correct dictionary headword form

Morphology

• **Morphemes:**
  • The small meaningful units that make up words
  • **Stems:** The core meaning-bearing units
  • **Affixes:** Bits and pieces that adhere to stems
    • Often with grammatical functions
Stemming

• Reduce terms to their stems in information retrieval
• *Stemming* is crude chopping of affixes
  • language dependent
  • e.g., *automate(s), automatic, automation* all reduced to *automat*.

```plaintext
for example compressed and compression are both accepted as equivalent to compress.
```

Sentence Segmentation

• !, ? are relatively unambiguous
• Period “.” is quite ambiguous
  • Sentence boundary
  • Abbreviations like Inc. or Dr.
  • Numbers like .02% or 4.3
• Build a binary classifier
  • Looks at a “.”
  • Decides EndOfSentence/NotEndOfSentence
  • Classifiers: hand-written rules, regular expressions, or machine-learning

```plaintext
for example compress and compress ar both accept as equivel to compress
```
Minimum Edit Distance

- Not assigned, but fyi, quantifies similarity of two strings
  - Word similarity is useful for spelling correction