CS124 Video Notes

Basic Text Processing

Regular Expressions

- **Regular Expression**: a formal language for specifying text strings
  - **Brackets**: [].
    - Any letter inside square bracket `[wW]`
    - Ranges: `[A-Z]`
  - **Negations**: ^ first character inside disjunction
    - Carat means negation only when **first** in []
    - `[^A-Z]` means not an upper case letter
  - **Pipe**: |
    - or
  - **Special Characters**:
    - `?`: optional previous character
    - `*`: 0 or more previous character
    - `+`: 1 or more of previous character
    - `.`: any character
    - `\d` any digit
    - `\D` any non-digit
    - `\w` any alphanumeric/underscore
    - `\W` a non-alphanumeric
    - `\s` whitespace
    - `\S` Non whitespace
  - Anchors:
    - `^` is start of line
    - `$` is end of line
    - `\`: real character for period
    - `\b`: word boundary

Word Tokenization

- Every NLP task needs to do text normalization
- **Lemma:** Same stem, part of speech, rough word sense (cat = cats)
- **Wordform**: the full inflected surface form (cat ≠ cats)
- **Type**: an element of the vocabulary
- **Token**: an instance of that type in running text
- **Standard Unix tools**:

    - `tr`: takes every instance of a character and replaces it with new character
        - E.g. `tr -sc A-za-z '\n' < shakes.txt` takes every nonalphabetic character and replaces it with a `\n`
    - `sort`: sorts
    - `uniq`: creates `Counter`
- Maximum Matching Word Segmentation Algo:
    1. Start a pointer at the beginning of the string
    2. While not end of string:
        1. Find the longest word in dictionary that matches the string starting at pointer
        2. Move the pointer over the word in string

Word Normalization and Stemming

- Need to normalize terms (U.S.A. → USA)
- Case folding:
    - Applications like IR reduce all letters to lower case
    - Lemmatization: Reduce inflections or variant forms to base form
    - Morphology: small meaningful units make up words
        - **Stems:** Core meaning-bearing units
        - **Affixes**: Bits and pieces that adhere to stems, often used for grammatical functions
    - Stemming: crude chopping of affixes
        - Porter's algorithm: a series of replacement rules

Sentence Segmentation

- Build a binary classifier that decides whether `.` is end of sentence or not end of system
- Utilize binary decision tree
    - More sophisticated feature such as period's wordshape