CIS5300 - Speech and Language Processing - Chapter 2 Notes

Jonathon Delemos - Chris Callison Burch

June 21, 2025

0.1 Abstract - Regular Expressions, Tokenization, Edit Distance

Text Normalization is a process that involves using regular expressions to convert words to a more convenient, standard form. This requires **Tokenization**, which is a fancy way to say categorizing things. Another part of text normalization is **lemmatization**, the task of determining whether two words have the same root, despite their differences. Think of a conjugation machine. Finally, we will focus on **edit distance**. That is to say what would it take to translate one string into another?

0.2 2.1 - Regular Expressions

The most common regular expressions is **concatenation**. That means using the regular expression to find a group of chosen letters. We can also use **disjunction** to match case sensitve words. Another technique is **ranges**. This means finding any range of number.

- **Literal Match (Concatenation)**:
  - `/woodchuck/` matches "woodchuck"
  - `/Buttercup/` matches strings containing "Buttercup"
- **Character Classes (Disjunction)**:
  - `/[wW]oodchuck/` matches "woodchuck" or "Woodchuck"
  - `/[abc]/` matches "a", "b", or "c"
  - `/[0-9]/` matches any digit
  - `/[a-z]/`, `/[A-Z]/` match lowercase or uppercase letters
  - `/[b-g]/` matches any character from b to g
- **Negated Character Classes**:
  - `/â/` matches any character except "a"
- **Optional Characters (?)**:
  - ? means the preceding character or nothing
  - `/woodchucks?/` matches "woodchuck" or "woodchucks"
  - `/colou?r/` matches "color" or "colour"
- **Repetition (Kleene Star and Plus)**:
  - `/a*/` matches zero or more "a" characters
  - `/aa*/` matches one or more "a" characters
  - `/[ab]*/` matches any string of a's and b's (including empty)
  - `/[0-9]+/` matches one or more digits (shorthand for integer)

- **Wildcard (.)**:
  - `/beg.n/` matches "begin", "beg'n", "begun", etc.
- **Anchors and Boundaries**:
  - `\B` matches non-word boundary
- **Grouping and Precedence**
  - `/cat|dog/` matches cat *or* dog
- **Special Characters**
  - `*` — Zero or more occurrences of the preceding expression
  - `+` — One or more occurrences of the preceding expression
  - `?` — Zero or one occurrence (optional) of the preceding expression
  - `{n}` — Exactly `n` occurrences
  - `{n,m}` — Between `n` and `m` occurrences (inclusive)
  - `{n,}` — At least `n` occurrences
  - `{,m}` — Up to `m` occurrences

So /a$\dot{2}$4z/ will match a followed by 24 dots followed by z (but not a followed by 23 or 25 dots followed by a z).