1.1 Recap last Lecture

- Übung
  - Fragen?
  - Bearbeitungszeit unterschiedlich (1.5h - 8h)
  - Beispiellösungen aufgeschaltet, andere Wege möglich
  - ungewollte Hürde: locate nicht standardmässig installiert
- Frequenzanalysen
  - Übersicht gewinnen: Wo liegen Schwerpunkte?
  - komparative Vergleich über Zeit oder Akteure (z.B. Partei)
  - vergleichbar mit Ngram-Viewer

1.2 Outline

- Halbzeit von Semester, langsam gehts ans Eingemachte
- Simpler Plan, RegEx allerdings mühsam
- uralt, aber unumgänglich für Data Cleaning
- je nach Zeit, nächstes Mal nochmals RegEx + Übungszeit

2.2 Text as Pattern
2.3 Formal Search Patterns

- Was meint Text als Pattern?
  - am einfachsten an Problemstellung zu sehen
  - Email-Adressen sind immer nach dem gleichen Muster aufgebaut
  - ganze Sprache ist voller Muster –> Grammatik
- Frage an Studis: Wie macht ihr das?
  - Bsp. Marketing-Analyse oder Wistleblower Korpus
- allen bekannt: Suche in Text
  - Suche nach @ findet alle Adressen
  - wie aber extrahieren und welche Teile gehören genau dazu?
- kryptisch + hässlich, aber beliebig expressive Beschreibungssprache

2.4 What are patterns for?

- RegEx mit breiter Anwendung
  - für Preprocessing Textanalysen unverzichtbar
  - Data Cleaning
- funktioniert genau gleich in Python, R und anderen Programmiersprachen

2.5 Data Cleaning is paramount!

- Aufbereitung braucht viel Zeit
- einfaches Modell mit ein paar Zeilen Code, Bereinigung immer spezifisch für Datenquelle

src

2.6 What are Regular Expressions (RegEx)?

- Regex = Muster = generalisierende Beschreibung
- Erklären von String = Zeichensequenz
- zwei Arten von Zeichen
- Literale = Zeichen steht für tatsächliches Zeichen (buchstabentreue Repräsentation)
  - wie letztes Mal
- Meta-Zeichen = Zeichen mit spezieller Bedeutung
  - anfänglich verwirrend
  - Thema heutiger Sitzung
- genaue mathematische Definition hier nicht Thema

2.7 Finding + Extracting
3.7 check a regular expression quickly
4.7 extract raw match only to allow for subsequent counting

- Empfehlung: egrep benutzen statt grep

4.8 Quantifiers

- erste Klasse von Meta-Symbolen: Quantifikatoren
- definieren Anzahl von vorangehendem Zeichen
- in Regex beziehen sich Operatoren auf vorderes Zeichen, in Wildcard nicht

4.9 Character Sets
5.9 match the capitalized and non-capitalized form
6.9 match sequences of 3 vowels
7.9 extract all bigrams (sequence of two words)
7.10 Special Symbols
8.10 match anything between brackets

- Klammern sind auch Metasymbole

8.11 The power of `.*` ...
8.12 More Complex Examples
9.12 extract basename of URLs
10.12 extract valid email adresses (case-insensitive)

- bei Erstellung von Online-Accounts prüfen RegEx Validität von Email

10.13 Combining RegEx with Frequency Analysis
11.13 count political areas by looking up words ending with “politik”
12.13 count ideologies/concepts by looking up words ending with “ismus”

- bis jetzt Spielerei, um RegEx zu lernen
- Grundlage für Seminararbeit
  - systematisches Suchen, quantifizieren und analysieren von Begriffsverwendung

12.14 Start simple, add complexity subsequently.
12.15 In-class: Exercise
13.15 Some not so random hints

- `egrep -roh "[A-Z]" **/*.txt | sort | uniq -c | sort -h`
- **Pause**
- CTRL+C um Befehl abzubrechen (falls länger als eine Sekunde dauert, ist etwas falsch)
- Start mit einfachem grep-Befehl, dann schauen, was gematcht wird und dann auszählen

13.16 In-class: Solution
13.17 Replacing + Removing
14.17 by setting the g flag in "s/llo/y/g",
15.17 sed replaces all occurences, not only the first one

- egrep für Extraktion, sed für Manipulation
  - wichtig um Daten aufzubereiten
- wie Suchen-Ersetzen-Funktion von Word, nur mächtiger dank Regex
- Löschen = Ersetzen mit leeren Sequenz
- flag "global"
- Demonstration mit
- `_echo "hello hell" | sed "s/l\b/lo/g"`

15.18 Contextual Replacing
16.18 swap order of name (last first -> first last)
17.18 matching also supports grouping
18.18 match any pair of two identical digits

- Teilausdruck gruppieren zur Wiederverwendung
- Klammern sind ebenfalls Metazeichen

18.19 More Meta-Symbols

- diese Symbole sind leer, sie matchen keine Zeichen
- spezifizieren Positon von regulärem Ausdruck
- line start hilfreich für übung

18.20 Greediness Trap
19.20 greedy: an apple, other apple
20.20 non-greedy: an apple

- `.*` = jegliche Zeichen, beliebige Länge

20.21 Assignment 2 :writing_hand:

-

20.22 In-class: Exercises I
20.23 In-class: Solution I
20.24 In-class: Self-Check