hci+d lab.

Lab 11: Text Visualization

정보 비주얼라이제이션 2015 Fall

human-computer interaction + design lab.
Joonhwan Lee

Text Visualization: Wordcould

Text Visualization

https://www.jasondavies.com/wordcloud/

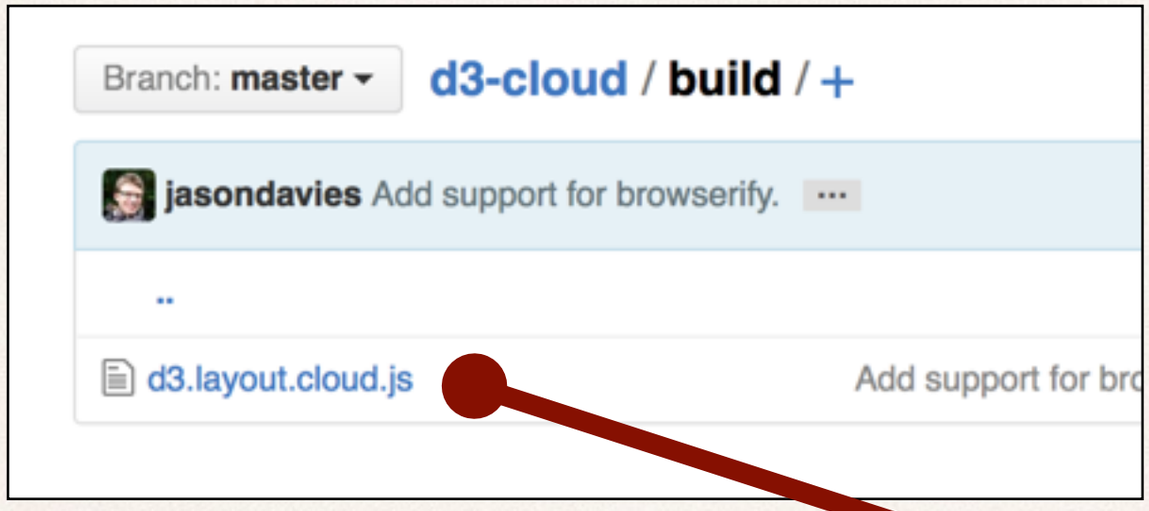
d3-cloud

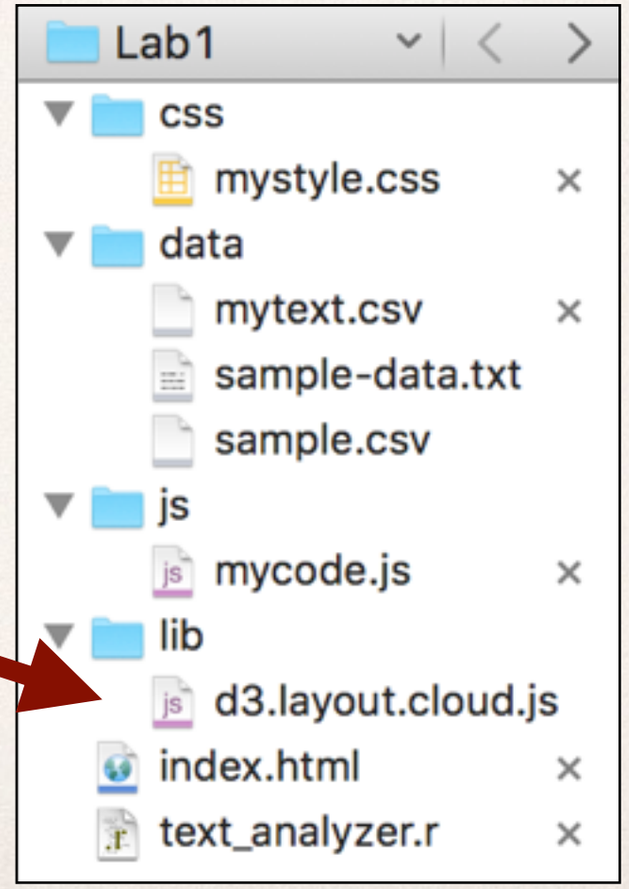
- d3.js의 기본 라이브러리는 워드클라우드 레이아웃을 제공하지 않음.
- 외부 API
  - https://github.com/jasondavies/d3-cloud
    - `d3.layout.cloud()` : cloud layout
    - `on(type, listener)` : 알고리즘을 작동하기 위한 리스너. 단어의 위치 계산
    - `start()` : 알고리즘 시작
    - `stop()` : 알고리즘 종료
    - `words([words])` : 레이아웃 계산을 위한 단어의 어레이

Lab 1: d3.js를 사용한 wordcloud

- Step1: 외부 라이브러리 등록

```
<script src="lib/d3.layout.cloud.js"
charset="utf-8"></script>
```





Lab 1: d3.js를 사용한 wordcloud

- Step2: Setup Variables

```
// 캔버스 사이즈
var width = 800, height = 800
// 칼라 라이브러리
var fill = d3.scale.category20()
// 단어들의 각도
var angle = 0
// 단어 출현 빈도수를 나누는 단계
var steps = 20
// 각 단계들간의 비율
var ratio = 2
// 최저 폰트 크기
var minsize = 3
// 스케일링 방법 (0:linear, 1:exponential)
var scaletype = 1
```

Lab 1: d3.js를 사용한 wordcloud

- Step3: Scale 선택
  - `d3.scale.linear()`
  - `d3.scale.quantile()`
  - `d3.scale.quantize()`

```
var scale = d3.scale.linear()
      .domain([0, d3.max(data, function (d)
            { return d.frq })])
      .range(d3.range(steps).map(function(i)
            { return i+minsize }))
```

Lab 1: d3.js를 사용한 wordcloud

- Step4: Layout 설정

```
d3.layout.cloud().size([width, height])
    .words(
        data.map(function(d) {
            if (scaletype == 0) {
                return { text: d.text, size:
                        scale(d.frq)*ratio }
            } else {
                return { text: d.text, size:
                    Math.pow(scale(d.frq), ratio) }
            }
        })
    )
```

Lab 1: d3.js를 사용한 wordcloud

- Step4 (cont.): Layout 설정

```
    .padding(0)
    .rotate(function() {
      // 0, 90
      return ~~(Math.random() * 2) * angle
    })
    .font("Impact")
    .fontSize(function(d) {
      return d.size;
    })
    .on("end", draw)
    .start()
```

Lab 1: d3.js를 사용한 wordcloud

- Step5: Wordcloud 그리기

```
function draw(words) {
 d3.select("body").append("svg")
        .attr("width", width)
        .attr("height", height)
        .append("g")
        .attr("transform", "translate(" + width/2
              + "," + height/2 + ")")
        .selectAll("text")
        .data(words)
        ...
}
```

Lab 1: d3.js를 사용한 wordcloud

- Step5 (cont.): Wordcloud 그리기

```
// data.map(function(d) {...}) 에서 size 결정
.style("font-size", function(d) {
    return d.size + "px"
})
.style("fill", function(d, i) {
    return fill(d.size)
})
```

R로 작성한 텍스트 분석기

- 분석에 사용할 텍스트 → 형태소 분석기를 이용하여 품사별로 분류한 후 빈도수 계산해야 함
- Python, Ruby, R 등으로 분석 가능
  - Python: KoNLPy
    - http://konlpy.org/en/v0.4.4/
    - 꼬꼬마 (http://kkma.snu.ac.kr), Twitter 형태소 분석기 (https://github.com/twitter/twitter-korean-text) 등 사용 가능
  - Ruby: twkorean
    - https://github.com/jun85664396/twkorean-ruby
    - Twitter 형태소 분석기 기반
  - R: KoNLP

R 형태소 분석기 KoNLP

- KoNLP
  - http://cran.r-project.org/web/packages/KoNLP/KoNLP.pdf
  - https://github.com/haven-jeon/KoNLP/wiki/KoNLP-examples
- install.packages("KoNLP")
  install.packages("rJava")
  install.packages("hash")
  install.packages("tau")
  install.packages("Sejong")
- library("KoNLP")

R 형태소 분석기 KoNLP

- 간단한 명령어
  - is.hangul / is.ascii → TRUE/FALSE
  - extractNoun("한글 형태소 분석기로 분석한 한글문장")
    => [1] "한글" "형태소" "분석" "기" "분석" "한" “한글문장"
  - sapply("한글 형태소 분석기로 분석한 한글문장", extractNoun)

text_analyzer.r

- Read text from a file

```
f <- file("data/sample-data.txt", blocking=F)
```

- Read each line and create a list

```
lines <- readLines(f)
```

- Extract noun (using KoNLP)

```
nouns <- sapply(lines, extractNoun)
```

- Finally close file

```
close(f)
```

text_analyzer.r

- Data cleanup

```
nouns <- nouns[-which(nouns=="")]
```

  - 명사 리스트 중, 빈 문자열 (“”)을 삭제

```
wcount <- table(unlist(nouns))
```

  - nouns 변수에 저장된 단어들을 table 함수를 이용하여 빈도수를 계산한다. table 함수는 서로 다른 길이를 가진 리스트에는 적용할 수 없기 때문에 리스트 형태로 저장된 변수를 unlist 함수를 사용하여 벡터 형태로 변환.

text_analyzer.r

- 데이터 저장

```
# convert to data frame and assign header
df <- as.data.frame(wcount)
colnames(df) <- c("text","frq")

# save data
write.table(df, file="data/sample.csv",
sep=",", col.names=NA)
```

Lab 2: R을 사용한 wordcloud

- Wordcloud
  - install.packages("wordcloud")
  - library(wordcloud)
  - library(RColorBrewer) #color palette

Lab 2: R을 사용한 wordcloud

- Color palette

```
pal <- brewer.pal(8,"Dark2")
wcount <- table(unlist(nouns))
```

- 한글폰트 지정 (Macintosh 에서만 테스트)

```
par(family="Apple SD Gothic Neo")
```

Lab 2: R을 사용한 wordcloud

- Wordcloud 생성
  - ```
    wordcloud(names(wcount),
    freq=wcount,
    scale=c(8,.2),
    min.freq=2,
    max.words=Inf,
    rot.per=.1,
    colors=pal)
    ```


    ```

Questions...?