**[Day-02-Lecture-01]**

## 관련 논문

> **최운호, 김동건 (2009), "군집분석 기법을 이용한 텍스트의 계통 분석-수궁가 '고고천변' 대목을 대상으로", [인문논총 62] pp. 203~229, 서울대학교 인문학연구원.**
>
> **최운호, 김동건 (2012), "「춘향가」 서두 단락의 어휘 사용 유사도를 이용한 판본 계통 분류 연구", [한국정보기술학회논문지 10(4)] pp. 111~117, 한국정보기술학회.**

**[폴더] sprod > sprod_001**

- **spr000_002_옛글자.hwp (pdf)**
  - 춘향가 서두단락이 있는 이 파일을 읽어봅시다.
  - 이 파일의 서두 단락에 사용된 어휘들을 비교하기 위해서는 다음과 같은 문제를 해결해야 합니다.
    - 표기법 정규화
    - 형태분석 > 어휘형태 추출
    - 어휘형태의 사용 양상에 따른 비교/거리 척도(distance metric) 결정

- **spr001_001.txt**
  - 판본별로 형태 분석을 모아놓은 결과이다.
- **spr001_002.txt**
  - 어휘 형태만 추출
- **spr001_003.R**
  - Jaccard Similarity 계산 샘플

```
#
# spr001_003.R
#

rm(list=ls())
CBSH = c('강산정기/N', '군산만학부형문/N', '나/V', '남원부/N', '녹주/N',
'동/N', '산수정기/N', '생겨나/V', '생기/V', '생하/V', '서/N', '서시/N', '수
려/N', '쌍각산/N', '약야/N', '어리/V', '왕소군/N', '저라산/N', '적/N', '적성
강/N', '전라좌도/N', '절대가인/N', '종출/N', '지리산/N', '춘향/N', '타/V')
CJJB = c('간성지장/N', '계계승승/N', '금고옥촉/N', '기운/N', '남/V', '덕/N',
'버금/N', '산하/N', '성자성손/N', '숙종대왕/N', '시절/N', '없/V', '요순/N',
'용양호위/N', '우탕/N', '의관문물/N', '있/V', '조정/N', '존비/N', '좌우보
필/N', '주석지신/N', '즉위/N', '피/V', '향곡/N', '흐르/V')
CJKS = c('강산정기/N', '군산만학부형문/N', '금강활이아미수/N', '나/V', '남원
부/N', '녹주/N', '동/N', '산수정기/N', '생기/V', '생장/N', '서/N', '서시/N',
'설도/N', '수려/N', '쌍각산/N', '약야계/N', '어리/V', '왕소군/N', '저라산/N',
'적성강/N', '절대가인/N', '제/N', '종출/N', '지리산/N', '춘향/N', '타/V', '태
어나/V', '호남좌도/N', '환출/N')
```

```r
CJSH = c('곳곳이/B', '관왕묘/N', '나/V', '남녀/N', '남북강성/N', '남원부/N', '당당하/V', '대방국/N', '동/N', '만고충신/N', '모시/V', '북통운암/N', '산수정기/N', '서/N', '수/N', '승지/N', '아니/B', '어리/V', '옛날/N', '일색/N', '있/V', '적성강/N', '지리산/N', '충렬/N', '하/V', '호남좌도/N')
CKSH = c('강산정기/N', '군산만학부형문/N', '금강활이아미수/N', '나/V', '남원부/N', '녹주/N', '동/N', '산수정기/N', '생기/V', '생장/N', '서/N', '서시/N', '설도/N', '수려/N', '쌍각산/N', '약야계/N', '어리/V', '왕소군/N', '저라산/N', '적성강/N', '절대가인/N', '제/N', '종출/N', '지리산/N', '춘향/N', '타/V', '태어나/V', '호남좌도/N', '환출/N')
CKYR = c('숙종대왕/N', '즉위/N', '초/N')
CKYS = c('강산정기/N', '군산만학부형문/N', '금강활이아미수/N', '나/V', '남원부/N', '동/N', '문군/N', '산수정기/N', '생겨나/V', '생기/V', '서/N', '설도/N', '셍겨나/N', '어리/V', '영웅열사/N', '왕소군/N', '우리나라/N', '적성강/N', '절대가인/N', '제/N', '지리산/N', '춘향/N', '타/V', '호남좌도/N', '환생/N')
CLSU = c('고요직설/N', '나/V', '때/N', '법/N', '숙종/N', '시절/N', '아동국/N', '여상/N', '요순시절/N', '은주/N', '이윤/N', '있/V', '자고로/B', '진실로/B', '충신/N', '현성지국/N', '홍모우순풍/N')
CPBS = c('간성지장/N', '계계승승/N', '금고옥촉/N', '기운/N', '남/V', '덕화/N', '산하/N', '성자성손/N', '숙종대왕/N', '시절/N', '없/V', '요순/N', '용왕호위/N', '있/V', '조정/N', '존비/N', '즉위/N', '피/V', '향곡/N', '흐르/V')

# ##########################
# Sample 01: sim 0 == dist 1
# ##########################

intersect(CBSH, CJJB)
union(CBSH, CJJB)
length(intersect(CBSH, CJJB))
length(union(CBSH, CJJB))
jac_sim = length(intersect(CBSH, CJJB)) / length(union(CBSH, CJJB))
jac_sim


# ##########################
# Sample 02: sim 1 == dist 0
# ##########################
intersect(CJKS, CKSH)
length(intersect(CJKS, CKSH))
union(CJKS, CKSH)
length(union(CJKS, CKSH))
jac_sim = length(intersect(CJKS, CKSH)) / length(union(CJKS, CKSH))

# ##########################
# Sample 03: sim = 0.5428571
# ##########################
intersect(CJKS, CKYS)
length(intersect(CJKS, CKYS))
union(CJKS, CKYS)
length(union(CJKS, CKYS))
jac_sim = length(intersect(CJKS, CKYS)) / length(union(CJKS, CKYS))
jac_sim
jac_dist = 1 - jac_sim
jac_dist
```

**[폴더] sprod > sprod_002**

- spr002_001.txt
  - header가 있는 Similarity Score 파일
- spr002_002.txt
  - header를 삭제한 Similarity Score 파
- spr002_003.py
  - okss_004_007.py 를 변경해서 만든 파일.
  - Jac. Sim.는 유사도 척도이기 때문에 이 값을 distance로 변경하는 코드로 수정한다.
    - [Line 45]에서 Sim.를 Dist.로 변경하는 코드로 수정.

```
#!python
#
#!-*-coding=utf-8-*-
#
# okss_004_007.py > spr002_003.py
#

from tqdm import tqdm

if __name__ == "__main__":

    with open("spr002_002.txt", "r", encoding="utf-8") as f_in:
        m_lines = [l.strip() for l in f_in.readlines()]

        dic_book_code = {}
        # 작품 ID 구하기
        print("Preprocessing book codes.")
        for line in tqdm(m_lines):

            cur_elts = line.split("\t")
            if cur_elts[0] in dic_book_code.keys():
                dic_book_code[cur_elts[0]] += 1
            else:
                dic_book_code[cur_elts[0]] = 1

            if cur_elts[1] in dic_book_code.keys():
                dic_book_code[cur_elts[1]] += 1
            else:
                dic_book_code[cur_elts[1]] = 1

        tbl_size = len(dic_book_code.keys())
        tbl_dist = [ [0] * tbl_size for _ in range(tbl_size)]
        list_keys = list(dic_book_code.keys())

        for idx_i in range(len(list_keys)):
            dic_book_code[list_keys[idx_i]] = idx_i

```

```
        # Table Index 안에 값 채워 넣기.
        print("\nFilling the distance values into the matrix.")
        for line in tqdm(m_lines):
            cur_elts = line.split("\t")
            elt_01 = cur_elts[0]
            elt_02 = cur_elts[1]
            elt_dist = 1 - float(cur_elts[2])
            idx_01 = dic_book_code[elt_01]
            idx_02 = dic_book_code[elt_02]
            tbl_dist[idx_01][idx_02] = elt_dist
            tbl_dist[idx_02][idx_01] = elt_dist

        # Table 출력하기
        print("\nPrinting the table to the output file.")

        with open("spr002_003.txt", "w", encoding="utf-8") as f_out:
            # Header 출력
            str_header = '\t'.join(list_keys)
            print("", str_header, sep="\t", file=f_out)
            for idx_i in tqdm(range(len(tbl_dist))):
                list_line = list(map(str, tbl_dist[idx_i]))
                str_key = list_keys[idx_i]
                str_line = "\t".join(list_line)
                print(str_key, str_line, sep="\t", file=f_out)

```

---

R Code (MDS, HClust)

```
#
#

#install.packages("vegan", "amap", "scatterplot3d")

require(ape)
require(MASS)
require(graphics)
require(vegan)
require(amap)
require(scatterplot3d)

options(digits=22)

setwd('d:/current_work/kwonks_drill/sprod/sprod_002')

kdkjac = read.table('spr002_003.txt')
kdkjac.d = as.dist(kdkjac)

kdkjac.mds1 = cmdscale(kdkjac.d, k=1)
kdkjac.mds2 = cmdscale(kdkjac.d, k=2)

```

```r
kdkjac.mds3 = cmdscale(kdkjac.d, k=3)
kdkjac.mds4 = cmdscale(kdkjac.d, k=4)
kdkjac.mds5 = cmdscale(kdkjac.d, k=5)

plot(kdkjac.mds2[, 1], kdkjac.mds2[, 2], "p")

######################
# HC (Cophenetic Cor.)
######################

kdk_hc.average = hclust(kdkjac.d, method="average")

#####################################
# Hierarchical Clustering (average method)
#####################################
X11(72, 48)
plot(kdk_hc.average, hang=-1, cex=.8, main="Hierarchical Clustering (average)\nJaccard Similarity (Distance) Index",  sub="", xlab="Work IDs", ylab="Distance (0~1)")


######################
# Clustering 16 clusters
######################

X11(72, 48)
plot(kdk_hc.average, hang=-1, cex=.8, main="Cut into 16 clusters",  sub="", xlab="Work IDs", ylab="Distance (0~1)")
rect.hclust(kdk_hc.average, k=16, border="red")

#########################
#
# Unrooted Tree
#
#########################

kdk_hc.tree = as.phylo.hclust(kdk_hc.average)
X11(64, 64)
plot(kdk_hc.tree, type="u", cex=0.8, font=3)

########################
# Left-Justified Clade
########################
kdk_hc.tree = as.phylo.hclust(kdk_hc.average)
X11(64, 64)
plot(kdk_hc.tree, type="c", use.edge.length=FALSE, direction="l", cex=.8)
```