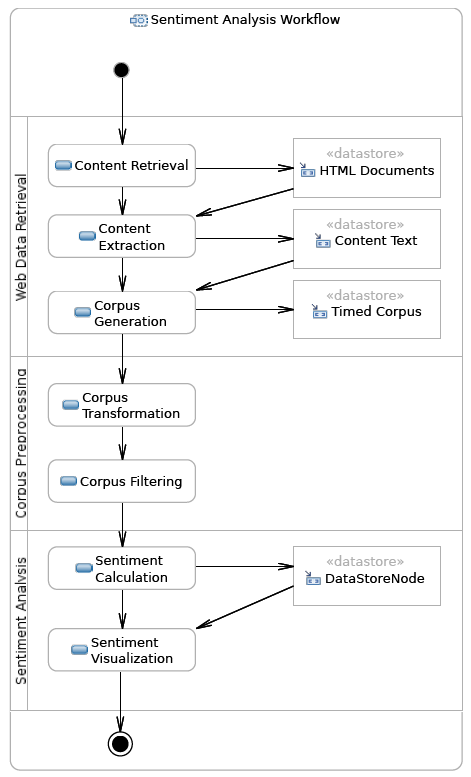
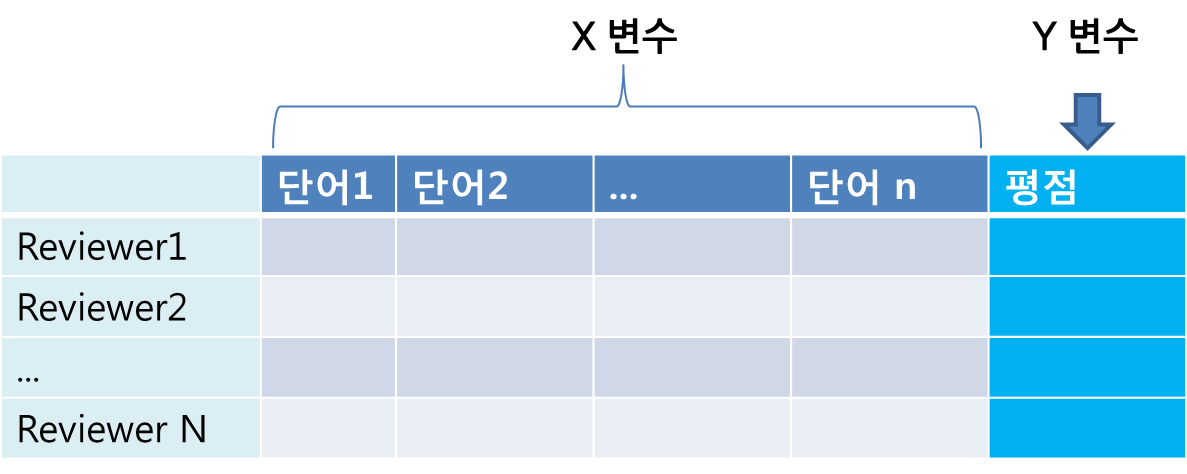
# 감정분석

<http://statmath.wu.ac.at/courses/SNLP/Presentations/DA-Sentiment.pdf>









## <1> [R] 패키지 & 함수

### 라이브러리 설치

|  |
| --- |
| install.packages(c("xts", "TTR"))  install.packages(c('tm', 'glmnet', 'devtools', 'caret'), repos='http://cran.nexr.com')  library(devtools)  install\_github("mannau/tm.plugin.sentiment") |

|  |  |  |
| --- | --- | --- |
| **함수** | **설명** | **예** |
| glmnet::glmnet() | 회귀분석  ∙ family  = “binomial” #Y가 이진값  = “Gaussian” # Y가 연속형 | glmnet(X, Y, family = "binomial", lambda = 0) |
| glmnet::cv.glmnet() | 회귀분석  ∙ alpha  = 1 : 라쏘  = 0 : 릿지  0 < alpha < 1 : 엘라스틱 넷   * 라쏘와 릿지의 비율로 표현. alpha =0.7로 하게 되면, 라쏘에 가중이 가는것 | ## 라쏘  cv.glmnet(X, Y, family = "binomial", alpha = 1)  cv.glmnet(X, Y, family = "binomial", alpha = 1, nfolds = 4, type.measure = "class") # k-fold값을 정해줌.  ## 릿지  cv.glmnet(X, Y, family = "binomial", alpha = 0, nfolds = 4, type.measure="class")  ## 엘라스틱 넷  cv.glmnet(X, Y, family = "binomial", alpha = .5, nfolds = 4, type.measure="class") |
| tm.plugin.sentiment ::polarity() | 감정 점수 계산  ∙ 긍정단어  ∙ 부정단어 | polarity(dtm, names(pos.lm), names(neg.lm)) |
| caret:: confusionMatrix() | 평가 | confusionMatrix(senti.lm.b.test, mobile.test$Sentiment) |

### [연습문제] 상품평 - 단어사전과 감정예측

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115 | library(caret)  library(glmnet)  commet.org = read.csv("현대홈쇼핑\_퍼팩트파운데이터(2564).csv", stringsAsFactors=FALSE)  1. 긍부정 사전 만들기  # tran-data/test-data 분리 set.seed(1234) train.idx = caret::createDataPartition(y = commet.org$star, p = 0.8, times = 1, list = F)  train.data <- commet.org[train.idx,] # 포함 test.data <- commet.org[-train.idx,] # 제거  texts <- unlist(train.data$comment) cps <- VCorpus(VectorSource(texts)) dtm <- DocumentTermMatrix(cps,control=list(removePunctuation = T ,  tokenize=ko.words,   wordLengths=c(2,5),   removeNumbers=T,  weighting = weightTfIdf))  x = as.matrix(dtm) y = ifelse(train.data$star <= 3, 0, 1) # 평점이 3점이하라면 부정으로 봄. # elastic net으로 학습 res.elastic = cv.glmnet(x, y, family = "binomial", alpha = .5, nfolds = 4, type.measure="class") coef.elastic = coef(res.elastic, s = "lambda.min")[,1]  # 회귀계수에 따른 긍.부정 분리 pos.elastic = coef.elastic[coef.elastic >= 0] # =0 포함여부는 데이터 보고 확인할 것  neg.elastic = coef.elastic[coef.elastic < 0]  head(sort(pos.elastic, decreasing = TRUE))  ## (Intercept) 메가 결재 리카락 번거롭긴 넘좋아  ## 3.527584 2.081867 1.831894 1.448976 1.407791 1.399788  print("[부정사전 확인]")  ## [1] "[부정사전 확인]"  head(sort(neg.elastic, decreasing = TRUE))  ## 요가 였나 실험 볼에실험 아니지않  ## -2.378973e-05 -1.059134e-04 -1.666770e-04 -1.396701e-03 -1.416550e-03  ## 편해보  ## -2.220505e-03  # 파일저장 - rownames()에 단어가 있으므로 함께 저장  write.csv(pos.elastic, "dic\_pos.csv", row.names = TRUE)  write.csv(neg.elastic, "dic\_neg.csv", row.names = TRUE)  # 감정점수 확인하기 senti.elastic <- tm.plugin.sentiment::polarity(dtm, names(pos.elastic), names(neg.elastic)) #-1~1 점수를 갖는다. senti.elastic.b <- ifelse(senti.elastic > 0, 1, 0) caret::confusionMatrix(senti.elastic.b, y) # 85%  ## Confusion Matrix and Statistics ##  ## Reference ## Prediction 0 1 ## 0 350 377 ## 1 195 3369 ##  ## Accuracy : 0.8667  ## 95% CI : (0.8562, 0.8767) ## No Information Rate : 0.873  ## P-Value [Acc > NIR] : 0.8957  ##  ## Kappa : 0.4739  ## Mcnemar's Test P-Value : 3.79e-14  ##  ## Sensitivity : 0.64220  ## Specificity : 0.89936  ## Pos Pred Value : 0.48143  ## Neg Pred Value : 0.94529  ## Prevalence : 0.12701  ## Detection Rate : 0.08157  ## Detection Prevalence : 0.16942  ## Balanced Accuracy : 0.77078  ##  ## 'Positive' Class : 0  ##  2. 감정점수 예측  # 감정사전 불러오기  dic\_pos.org = read.csv("dic\_pos.csv", stringsAsFactors = FALSE) dic\_neg.org = read.csv("dic\_neg.csv", stringsAsFactors = FALSE) dic\_pos = unlist(dic\_pos.org[, 1]) dic\_neg = unlist(dic\_neg.org[, 1])  # test data 준비  texts <- unlist(test.data$comment)  cps <- VCorpus(VectorSource(texts))   dtm.test <- DocumentTermMatrix(cps,control=list(removePunctuation = T ,  tokenize=ko.words,   wordLengths=c(2,5),   removeNumbers=T,  weighting = weightTfIdf,  dictionary = c(dic\_pos, dic\_neg)))  senti.elastic.test <- tm.plugin.sentiment::polarity(dtm.test, names(pos.elastic), names(neg.elastic)) senti.elastic.b.test <- ifelse(senti.elastic.test > 0, 1, 0) caret::confusionMatrix(senti.elastic.b.test, ifelse(test.data$star <= 3, 0, 1))  ## Confusion Matrix and Statistics ##  ## Reference ## Prediction 0 1 ## 0 54 87 ## 1 80 850 ##  ## Accuracy : 0.8441  ## 95% CI : (0.8209, 0.8653) ## No Information Rate : 0.8749  ## P-Value [Acc > NIR] : 0.9987  ##  ## Kappa : 0.3033  ## Mcnemar's Test P-Value : 0.6424  ##  ## Sensitivity : 0.40299  ## Specificity : 0.90715  ## Pos Pred Value : 0.38298  ## Neg Pred Value : 0.91398  ## Prevalence : 0.12512  ## Detection Rate : 0.05042  ## Detection Prevalence : 0.13165  ## Balanced Accuracy : 0.65507  ##  ## 'Positive' Class : 0  ## |

### ADP 7회 유사 문제 만들기

1. 다음 파일을 읽으세요.

2. 긍정단어와 부정단어를 단어사전에 추가하시오.

- 긍정 단어 40개

어머니, 선물, 스틱, 감사, 고민, 만족, 머리결, 휴대용, 후기, 효율성,

횟수, 활용, 확실, 화장실, 화면, 차홍원장님, 홍보, 호기심, 형태, 헤어케어

향, 향기, 행복, 핸드백, 품질, 필수품, 편리함, 편리, 탁월, 칭찬,

좋은제품, 몇번, 가격, 가격대, 가는머리, 힘이없어서, 힘탓인지, 감쪽, 감탄, 강추

- 부정 단어 40개

신뢰감, 비용, 모델, 짜증, 개뿔, 바램, 최악, 헤어제품, 체험, 밀착력,

가격대, 고려, 대기업, 어깨, 긴머리, 불쾌, 그냥저냥, 애물단지, 고가, 고객,

고대, 기대, 고장, 곤란, 광고, 허위, 그냥저냥, 기름, 긴머리, 가격대,

간지, 갈라짐, 감고, 각보, 그렇게까진, 긴단발이라, 길이, 내손, 대체

- 불용어 사전 (stopwords.csv)

3. 해당 상품에 대해서 감성 점수를 파악하시오.

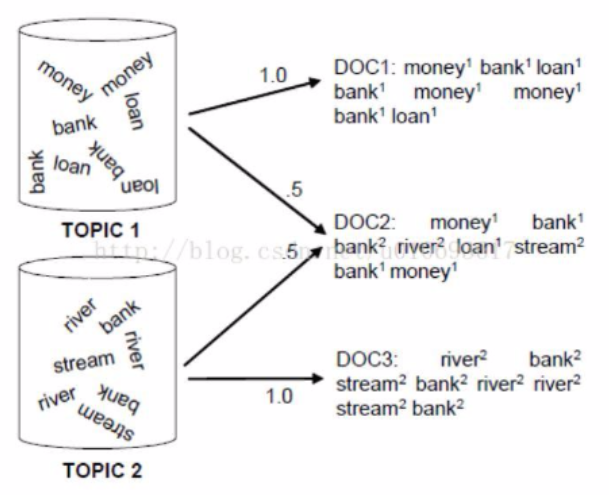
4. 2016년 데이터 중 월별로 호감도의 변동성을 그래프로 나타내라.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96 | ##############  #  # library  #  ##############  library(reshape)  library(data.table)  library(dplyr)  library(tm)  library(KoNLP)  options(mc.cores=1)  library(rJava)  #install.packages(c("xts", "TTR"))  #install.packages(c('tm', 'glmnet', 'devtools', 'caret'), repos='http://cran.nexr.com')  #library(glmnet)  library(devtools)  install\_github("mannau/tm.plugin.sentiment")  ##############  # function  ##############  ko.words <- function(doc){  d <- as.character(doc)  noun <- extractNoun(d, autoSpacing = T)  noun = unlist(noun)  noun = gsub('[[:punct:]]', '', noun) # 문장부호 제거  noun = gsub('[[:cntrl:]]', '', noun) # 특수문자 제거  noun<-gsub("\\d+","",noun) #숫자 제거  #noun<-gsub(" ","",noun) # 공백 제거  noun<-gsub("\\.","",noun) # 마침표 제거  noun<-gsub("\\^","",noun) # 꺽쇠들어간 글자 제거  noun <- gsub("[a-zA-Z]", "", noun) # 알파벳제거  noun = Filter(function(x){nchar(x)>=2 & nchar(x) <=5}, noun)    return (noun)  }  ##############  # file read & data cleaning  ##############  comment.org = fread("comment.csv", stringsAsFactors = FALSE)  comment = na.omit(comment.org)  ko.stopwords = read.csv("stopwords.csv", stringsAsFactors = FALSE)$x  #read.csv("pos.csv", stringsAsFactors = FALSE)$X  dic\_pos = c("어머니","선물","스틱","감사","고민","만족","머리결","휴대용","후기","효율성",  "횟수","활용","확실","화장실","화면","차홍원장님","홍보","호기심","형태","헤어케어",  "향","향기","행복","핸드백","품질","필수품","편리함","편리","탁월","칭찬",  "좋은제품","몇번","가격","가격대","가는머리","힘이없어서","힘탓인지","감쪽","감탄","강추")  #read.csv("neg.csv", stringsAsFactors = FALSE)$X#  dic\_neg = c("신뢰감","비용","모델","짜증","개뿔","바램","최악","헤어제품","체험","밀착력",  "가격대","고려","대기업","어깨","긴머리","불쾌","그냥저냥","애물단지","고가","고객"  ,"고대","기대","고장","곤란","광고","허위","그냥저냥","기름","긴머리","가격대",  "간지","갈라짐","감고","각보","그렇게까진","긴단발이라","길이","내손","대체","떡질까봐")  temp = comment$comment  texts <- unlist(temp) # list -> vector로 만듬.  cps <- VCorpus(VectorSource(texts))  ##############  # table  ##############  dtm <- DocumentTermMatrix(cps,control=list(removePunctuation = T ,  tokenize=ko.words,  wordLengths=c(2,5),  removeNumbers=T,  stopwords=ko.stopwords,  weighting = weightTfIdf,  dictionary = c(dic\_pos, dic\_neg))) # dictionary값을 주는 것이 중요  ##############  # 감정점수 산출  ##############  # 감정점수 확인하기 tm.plugin.sentiment::polarity()  senti.score <- tm.plugin.sentiment::polarity(dtm, dic\_pos, dic\_neg)  range(senti.score) # -1~1 점수를 갖는다.  senti.score.b <- ifelse(senti.score > 0, 1, 0)  ## not used : 정확도 확인용  #y <- ifelse(comment$star > 3, 1, 0) # 평점이 3초과면 긍정으로 판단  #caret::confusionMatrix(senti.score.b, y) # 84%  ##############  # 2016년 월별 긍부정 추이 - 그래프  ##############  comment$predict.senti.score = senti.score.b  date = as.POSIXlt(comment$date, format="%Y-%m-%d")  comment$year = date$year + 1900  comment$mon = date$mon+1  comment$day = date$mday  trend = comment %>% filter(year==2016) %>%  group\_by(mon) %>% summarize(score\_sum = sum(predict.senti.score, na.rm=T))  ggplot(trend, aes(x=mon, y=score\_sum)) + geom\_point(size=3) + geom\_line() # 9월부터 11월까지 감소 추세를 보이고 있음. |

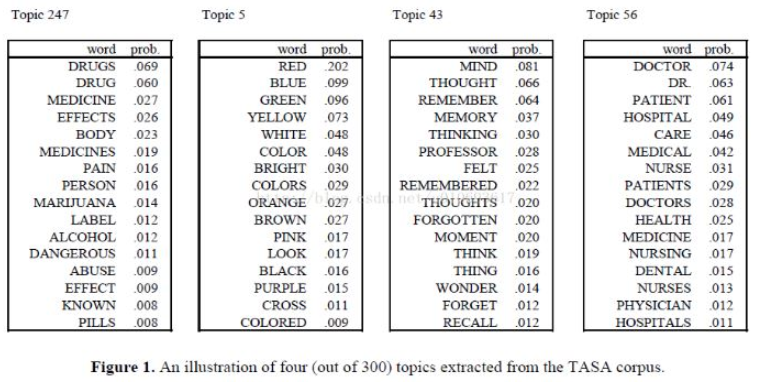
# 토픽분석(Topic model)



단어가(word)가 모이면 토픽(topic)이 되고, 토픽이 모이면 문서(document)가 된다.



‘bank’라는 단어를 생각해보자. 이 단어는 두가지 의미 (1 은행, 2. 둑, 제방)를 가지고 있다. 이 단어가 독립적으로 존재하고 있으면 정확한 의미를 알기 어렵다. 그런데 ‘bank’가 ‘money’와 ‘loan’과 같은 단어들과 함께 있다면, 이 ‘bank’는 은행이라는 뜻이라는 것을 알 수 있다. 마찬가지로 ‘bank’가 ‘river’나 ‘stream’과 같은 단어들과 함께 있다면 이때는 둑이나 제방을 의미할 확률이 높아진다.



이는 LDA 분석 결과이다. PCA처럼 PC(토픽)은 연구자가 직접 보고 판단해야 한다. 또한 토픽의 개수(k)는 k-means clustering에서 처럼 연구자가 정해주어야 한다. 아래 예시를 보면, 토픽 247은 약에 대한 내용, 토픽 5는 색깔, 토픽 43은 생각이나 느낌, 토픽 56은 병원에 대한 내용이라는 것을 **(사람이 직접 보고)**알 수 있는 것이다.



## <2> LDA(Latent Dirichlet Allocation) 알고리즘

※ 주의! LDA는 2가지 종류가 있음

LDA : 판별분석

LDA : 잠재 디리클레 할당 – ‘잠재’는 직접적으로 관찰되지 않은 변수를 의미함.

우리는 “잠재 디리클레 할당”을 의미하는 LDA를 학습할 것이다.

LDA는 k, G, alpha, eta값을 어떻게 설정하느냐에 따라 결과가 크게 달라질 수 있으며, 자신이 사용하는 데이터에 최적화된 설정값들을 찾아내는 것이 가장 중요한 부분이라고 할 수 있다.

## <3> [R] 패키지 & 함수

### 라이브러리 설치

|  |
| --- |
| install.packages(c('tm', 'lsa', 'topicmodels', 'lda', 'h2o'), repos='http://cran.nexr.com') |

|  |  |  |
| --- | --- | --- |
| **함수** | **설명** | **예** |
| topicmodels ::dtm2ldaformat() | dtm을 lda에서 쓰는 형식으로 변경  ※ topicmodels 패키지에서 사용할 수 있도록 Document Term Matrix -> lda type으로 변경해준다. | ldaform = dtm2ldaformat(dtm, omit\_empty = T) |
| lda:: lda.collapsed.gibbs.sampler() | LDA분석  • K = # 토픽의 개수 (결과를 보고 늘리거나, 줄이면 된다)  • vocab = # 어떤 단어들이 있는지  • num.iterations = # 계속 반복해서 계산을 해줘야 결과가 좋아진다. 몇번을 반복할지 설정해준다. 많이 하면 결과가 좋아지지만, 시간이 늘어날 것이다. (결과를 보고 늘리거나, 줄이면 된다)  • burnin = # 앞부분의 계산결과는 부정확할 수 있다. 그래서 앞부분의 결과를 해당 개수만큼 지우라는 의미인데, 5000번 중에 앞의 1000번을 지워라. (결과를 보고 늘리거나, 줄이면 된다)  • alpha = # 중요! 토픽이 얼마나 골고루 섞여 있느냐를 의미함. 한 문서 내에서 30개의 토픽이 얼마나 골고루 넣어줄지를 결정. 1보다 작은값으로 설정. 얼마나 작은 값으로 해야 하느냐는 답이 없음.1은 모든 경우의 수가 똑같다는 의미이다. 1보다 커지게 되면 모든 문서가 토픽의 배합비율이 비슷하다. 30개의 토픽이 골고루 들어가 있다는 의미. (결과를 보고 늘리거나, 줄이면 된다)  • eta = # 중요! 한 토픽에서 단어들이 얼마나 골고루 있느냐를 의미함. 1보다 작은 값을 주는 것이 좋음 (결과를 보고 늘리거나, 줄이면 된다) | result.lda = lda.collapsed.gibbs.sampler(documents = ldaform$documents, K = 30, vocab = ldaform$vocab, num.iterations = 5000, burnin = 1000, alpha = 0.01, eta = 0.01  ) |

### [연습문제] 상품평 토픽분석

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | library(lda)  library(topicmodels)  commet.org = read.csv("현대홈쇼핑\_퍼팩트파운데이터(2564).csv", stringsAsFactors=FALSE) texts <- unlist(commet.org$comment) cps <- VCorpus(VectorSource(texts)) dtm <- DocumentTermMatrix(cps,control=list(removePunctuation = T ,  tokenize=ko.words,   wordLengths=c(2,5),   removeNumbers=T,  weighting = weightTfIdf))  ldaform = dtm2ldaformat(dtm, omit\_empty = T) result.lda = lda.collapsed.gibbs.sampler(documents = ldaform$documents,  K = 5, # 5개의 토픽   vocab = ldaform$vocab,   num.iterations = 5000,   burnin = 1000,   alpha = 0.01,   eta = 0.01)    top.topic.words(result.lda$topics)  ## [,1] [,2] [,3] [,4] [,5]  ## [1,] "기대" "잘받았어" "발림성" "정말" "사용"  ## [2,] "생각" "주문" "커버력" "화장" "구성"  ## [3,] "자연" "완전" "가격" "커버력" "좋습니"  ## [4,] "상품" "지속력" "배송" "화사" "선물"  ## [5,] "발립" "얼굴" "커버" "디자인" "방송"  ## [6,] "밀착력" "추천" "밀착감" "딸아이" "언니"  ## [7,] "조아" "생각" "피부" "얼굴" "커버력"  ## [8,] "잘쓰고" "건조" "감사" "강추" "만족"  ## [9,] "감사" "팩트" "만족" "밀착력" "ㅎㅎㅎㅎㅎ" ## [10,] "받았습니" "배송" "품질" "상품" "하기"  ## [11,] "오늘" "시간" "쓰겠습니" "독특" "제품"  ## [12,] "써보진" "색상" "잘쓸께" "맘에들" "엄마"  ## [13,] "잘발라지네" "퍼프" "구매했습니" "잘맞았으면" "재구매"  ## [14,] "커버력" "피부" "화사" "아주" "처음"  ## [15,] "편리" "괜찬고" "개봉" "환해" "발림성"  ## [16,] "립스틱" "다른제품과" "겠어" "별루" "쿠션"  ## [17,] "배송" "감사" "구석" "쿠션" "구성도"  ## [18,] "품질" "곱게잘먹어" "모공" "도전" "입자"  ## [19,] "밀착" "노랑" "걱정" "발리는거" "좋은거"  ## [20,] "너무" "얇게잘" "발림" "붉은기" "기대" |

# 단어 상관분석

“co : 같이, occur : 나타나다”는 의미이다.

왜 중요할까? 단어들이 어떤 단어들과 함께 사용되는지가 의미가 있다. ex) 은행 (1. 돈맡기는 은행, 2. 먹는 은행) 은행이란 뜻은 어떻게 알까? 주변에 사용되는 단어를 보면 알 수 있다.

## <1> Co-occurrence Matrix

Co-occurrence Matrix 를 만들면, 어떤 단어들과 함께 사용되는지 알 수 있게 된다.

예를 들어 다음과 같은 문장이 있다고 하자.

|  |
| --- |
| "오늘은 영화를"  "영화에는 팝콘이" |

영화는 오늘, 팝콘하고 관련이 있지만, 오늘하고 팝콘은 관련이 없다.

만드는 순서는 다음과 같다.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | texts = c("오늘은 영화를", "영화에는 팝콘이")  texts  cps <- VCorpus(VectorSource(texts))  tdm = TermDocumentMatrix(cps, control=list(removePunctuation = T ,  tokenize=ko.words,  wordLengths=c(2,5),  removeNumbers=T,  weighting = **weightBin**)) # 해당단어 사용 유/무  tdm.matrix=as.matrix(tdm)  tdm.matrix    tdm.matrix %\*% t(tdm.matrix) # t()는 전치 행렬이고, %\*%가 행렬연산이다.    # (해석) 영화와 영화는 2번 같이 쓰였고, 오늘과 오늘은 1번 같이 쓰였고, 오늘과 팝콘은 같이 사용되지 않았다. |

### [연습문제] 상품평 상관분석totodo : 06. 빅데이터\01. R프로그래밍\02. 실습\R\_기초\55. 텍스트마이닝\ 현대홈쇼핑\_퍼팩트파운데이터(2564).rmd

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74 | library(reshape)  library(dplyr)  library(ggplot2)  library(data.table)  library(tm)  library(KoNLP)  library(qgraph)  options(mc.cores=1)  commet.org = read.csv("현대홈쇼핑\_퍼팩트파운데이터(2564).csv", stringsAsFactors=FALSE) texts <- unlist(commet.org$comment) cps <- VCorpus(VectorSource(texts))  ############################## # # 상관분석(Co-occurrence Matrix 만들기) #  ############################## tdm <- TermDocumentMatrix(cps,control=list(removePunctuation = T ,  tokenize=ko.words,   wordLengths=c(2,5),   removeNumbers=T,  weighting=weightBin)) tdm.matrix = as.matrix(tdm)  word.count <- rowSums(tdm.matrix)  word.order <- order(word.count, decreasing=T)  freq.words <- tdm.matrix[word.order[1:20], ] co.matrix <- freq.words %\*% t(freq.words)  #### 시각화 qgraph(co.matrix,  labels=rownames(co.matrix),  diag=F,  layout='spring',  edge.color='blue',  vsize=log(diag(co.matrix))\*2)    ## 주어진 단어와 상관계수가 높은 단어를 찾음 # corlimit : 상관계수의 하한 head(findAssocs(tdm, "퍼프", corlimit=0.13))  ## $퍼프 ## 삼각 불편 삼각형 세모 구석 교체  ## 0.28 0.21 0.20 0.15 0.14 0.13  head(findAssocs(tdm, "색상", corlimit=0.13))  ## $색상 ## 그것 부인 센타  ## 0.14 0.14 0.14  head(findAssocs(tdm, "시간", corlimit=0.13))  ## $시간 ## 바를땐 절약 펴바른 화장 기구 설명서 작은 지나 컴팩트  ## 0.18 0.18 0.18 0.16 0.14 0.14 0.13 0.13 0.13 |

# 텍스트마이닝 총 정리

