Maurício Yoshida Izumi

Doutorando em Ciência Política (DCP/USP).

http://lattes.cnpq.br/2546701843557096

Meus exercícios

exec

Proposta de trabalho final

Plano A - função para calcular a similaridade entre dois textos

FUNÇÃO: text.similarity(text.a, text.b)

DESCRIÇÃO: A utilização de análises quantitativas de textos tem crescido no meio acadêmico. Por exemplo, é comum querermos saber quão próximos são os discursos de políticos, textos literários e artigos de jornais ou revistas. Portanto, esta função busca facilitar a utilização de textos como dados quantitativos. O principal objetivo dela será calcular o índice de similaridade de Sigelman e Buell (2004) entre dois textos em língua portuguesa. O índice de Sigelman e Buell (2004) é definido como:

sb = 100 - sum(|PAi - PBi|) / 2,

onde PA e PB são as porcentagens do total de atenção dada a cada termo i pelos textos A e B, respectivamente. O índice varia de 0 a 100 e quanto maior o seu valor, maior a similaridade entre os textos. Para fazer essa comparação a função irá excluir as preposições e outros caracteres não informativos (números, pontuações...) pois, caso contrário, iremos superestimar a similaridade entre os documentos.

INPUT: Os dados de entrada da função são dois vetores de tipo 'character' contendo os textos em língua portuguesa a serem comparados. Assim, o formato da função será:

text.similarity(text.a, text.b),

onde text.a será o vetor com um texto e text.b será o vetor com o outro texto.

OUTPUT: A função exibirá o número total de palavras em cada um dos vetores de entrada, o número total de palavras excluindo aquelas pouco informativas e o número total de termos únicos em cada texto (isto é, os termos utilizados para se calcular o índice de similaridade). Exibirá e irá guardar o valor do índice de Sigelman e Buell (2004).

Oi Mauricio...gostei bastante da sua ideia! Algumas

dúvidas/sugestões: 1)os textos precisam ser em português mesmo? não dá pra englobar outras línguas, ao menos inglês?; 2) medir a similaridade dos textos através de qto o termo "i" aparece nos textos é interessante, mas será que a ordem dos termos nos textos tbm não interessa? Eu digo, os termos podem ser os mesmos, fazendo com que os textos tenham alto índice de similaridade, mas os termos podem estar em ordem e colocação totalmente diferentes, fazendo com que os textos digam coisas totalmente dissimilares, não?; 3) Você pode colocar a referência que vc citou?; 4) A pessoa pode ter mais de dois textos que ela queira investigar a similaridade...ela não poderia fornecer por exemplo 3 textos e a sua função avaliar par a par e devolver a similaridade entre eles? ; 5)Você também pode fazer com que sua função retorne alguns dos termos que tiveram valores mais alto nos dois textos? ... Eu vi q vc já trabalhou com isso antes, então acho que esta função será divertida pra vc fazer! —-*Thais Lopes*

Olá Thais! Muito obrigado pelos comentários. Vou comentar cada um deles aqui. 1) Eu inclui na função a possibilidade de usar textos em inglês e em espanhol (além de português). Eu havia pensado apenas no português devido ao desafio de excluir os pronomes; 2) Como você observou, é possível que a ordem em que os termos aparecem seja importante. Mas também é possível que textos que utilizem termos muito diferentes (ou em ordens diferentes) sejam muito semelhantes. Além disso, em casos extremos, é possível que textos exatamente iguais tenham conotações totalmente diferentes. Por exemplo, se alguém usar um tom de ironia em sua fala, ele estará apresentando uma visão totalmente diversa, mesmo que o texto seja o mesmo. Ou seja, técnicas de análises quantitativas de textos tem algumas limitações. Mas se o pressuposto de que textos que utilizam conjuntos semelhantes de palavras possuem conteúdos mais similares seja válido. A técnica que estou utilizando seria um boa aproximação para verificar a similaridade entre os textos; 3) Essa é a referência: Sigelman, L. and Buell, E. H. (2004). Avoidance or engagement? issue convergence in us presidential campaigns, 1960–2000. American Journal of Political Science, 48(4):650–661; 4) eu deixei a função apenas com a possibilidade de se comparar dois textos pois, caso contrário, a função irá retornar muitos valores que serão de difícil interpretação para o usuário; 5) eu inclui isso na função. Agora ela retorna o índice de similaridade e uma tabela de frequências.

—-*Maurício Izumi*

Página de ajuda (Help):

```
text.similarity                package:unknown                R
Documentation

    Cálculo de similaridade entre dois textos

Description:

    A função text.similarity calcula o índice de similaridade desenvolvido
por Sigelman e Buell (2004) entre dois textos.


Usage:

     text.similarity(text.a, text.b, language = "portuguese")

Arguments:

    text.a: é um vetor de tipo character com um texto.
    text.b: é um vetor de tipo character com outro texto.
    language: indica o idioma dos textos a serem comparados. Temos 3 opções:
'portuguese' para textos em língua portuguesa;

'english' para  textos em língua inglesa; e 'spanish' para textos em lingua
espanhola.

Details:


Value:

     A função retorna uma lista de dois elementos:

  similar: contém o valor do índice de similaridade.

  freq: é um dataframe onde a coluna 'termos' possui os termos utilizados no
cálculo do índice; 'n.termos.a' a frequência de cada termo

no primeiro vetor; 'n.termos.b' a frequência de cada termo no segundo vetor;
'pa' a porcentagem de cada termo no primeiro vetor; e 'pb'

a porcentagem de cada termo no segundo vetor.


Warning:
```

```
Note:



Author(s):

     Maurício Izumi (mauricio.izumi@usp.br)

References:

     Sigelman, L. and Buell, E. H. (2004). Avoidance or engagement? issue
convergence in us presidential campaigns, 1960–2000. American

Journal of Political Science, 48(4):650–661.

See Also:


Examples:

## Exemplo com textos em português (trechos de discursos do Lula)

texto.a <- c("Meus companheiros e minhas companheiras, Excelentíssimos
senhores chefes de Estado presentes nesta solenidade,

Trabalhadores e trabalhadoras do meu Brasil, Meu querido companheiro José
Alencar, meu vice-presidente da República, Minha companheira

querida, Dona Mariza, esposa do José Alencar, Minha querida esposa Marisa
que, juntos, já partilhamos muitas derrotas e, por isso, hoje,

estamos realizando um sonho que não é só meu, mas um sonho do povo deste
país, que queria mudança.")
texto.b <- c("Sou profundamente grato à compreensão da dona Marisa Letícia
que, nesses quatro anos, esteve junto comigo, nos bons e nos

maus momentos. E, certamente, José Alencar e eu somos gratos também à dona
Mariza, a esposa do José Alencar, porque certamente nos

momentos difíceis ela era o ombro, o consolo e a consciência política para
nos afirmar: Continuem lutando...")

similar.lula <- text.similarity(texto.a, texto.b, language = "portuguese")
similar.lula

## Exemplo com textos em inglês (trechos de discursos do Obama)

texto.a <- c("I said then and believe now that Saddam Hussein was a ruthless
dictator who craved weapons of mass destruction but posed
```

```
no imminent threat to the United States")
texto.b <- c("I said that Saddam Hussein was a ruthless man, but that he
posed no imminent and direct threat to the United States. I

said that a war in Iraq would take our focus away from our efforts to defeat
al-Qaeda.")

similar.obama <- text.similarity(texto.a, texto.b, language = "english")
similar.obama


## Exemplo com  textos em espanhol (trechos de "El amor en los tiempos del
cólera" de Gabriel García Márquez)

texto.a <- c("Alcanzó a reconocerla en el tumulto a través de las lágrimas
del dolor irrepetible de morirse sin ella, y la miró por

última vez para siempre jamás con los ojos más luminosos, más tristes y más
agradecidos que ella no le vio nunca en medio siglo de vida

en común, y alcanzó a decirle con el último aliento: - Sólo Dios sabe cuánto
te quise.")
texto.b <- c("Florentino Ariza había pensado llevarle los setenta folios que
entonces podía recitar de memoria de tanto leerlos, pero

luego se decidió por media esquela sobria y explícita en la que sólo
prometió esencial: su fidelidad a toda prueba y su amor para

siempre.")

similar.ggm <- text.similarity(texto.a, texto.b, language = "spanish")
similar.ggm
```

Código da função

```
##
## Função text.similarity
##

        ## cria a função
text.similarity <- function(text.a, text.b, language = "portuguese"){
            ##
            ## 1. esta parte do código apenas verifica se
            ## os parâmetros informados estão corretos
            ##

        ## cria uma condição que verifica se os vetores são de tipo
character
    if(is.character(text.a) == F | is.character(text.b) == F){
        ## se não for de tipo character, envia uma mensagem de erro
```

```
        stop("Um dos seus vetores não é do tipo character!\n")
    }

        ## cria uma condição que verifica se o usuário escolheu um idioma
válido
    if(language != "portuguese" & language != "english" & language !=
"spanish"){
        ## se o usuário não escolheu um idioma válido, envia uma mesagem de
erro
        stop("Você não escolheu um idioma válido.\n As suas opções
são:\n'portuguese', 'english' ou 'spanish'\n")
    }

            ##
            ## 2a. esta parte do código começa a 'limpar' os dados
            ##

        ## insere um espaço no início e no final do vetor.
        ## Esta etapa é importante mais para frente.
        ## Se não tiver esse espaço, os pronomes no início ou no final
        ## do vetor não serão excluídos

        ## insere os espaços no primeiro vetor
    temp.a <- paste(" ", text.a, " ", sep = "")
        ## insere os espaços no segundo vetor
    temp.b <- paste(" ", text.b, " ", sep = "")

        ## exclui os números do primeiro vetor
    temp.a <- gsub("[[:digit:]]", " ", temp.a)
        ## exclui os números do segundo vetor
    temp.b <- gsub("[[:digit:]]", " ", temp.b)

        # exclui pontuação e caracteres especiais do primeiro vetor
    temp.a <- gsub("[[:punct:]]", " ", temp.a)
        # exclui pontuação e caracteres especiais do segundo vetor
    temp.b <- gsub("[[:punct:]]", " ", temp.b)

        # transforma os caracteres do primeiro vetor para minúsculo
    temp.a <- tolower(temp.a)
        # transforma os caracteres do segundo vetor para minúsculo
    temp.b <- tolower(temp.b)

                ##
                ## * conta o número de palavras que cada vetor possui
                ##

            ## separa as palavras do primeiro vetor
        count.a <- strsplit(temp.a, " ")
            ## separa as palavras do segundo vetor
        count.b <- strsplit(temp.b, " ")
```

```
        ## exclui campos vazios do primeiro vetor
    count.a <- count.a[[1]][count.a[[1]] != ""]
        ## exclui campos vazios do segundo vetor
    count.b <- count.b[[1]][count.b[[1]] != ""]

        ## conta o número de palavras no primeiro vetor
    count.a <- length(count.a)
        ## conta o número de palavras no segundo vetor
    count.b <- length(count.b)

            ##
            ## 2b. Cria a lista de preposições que serão excluídos
            ## A lista de preposições foi extraída de
http://www.intext.com.br/
            ##

        ## cria uma condição que verifica se o idioma escolhido foi
'portuguese'
    if(language == "portuguese"){
        ## cria uma lista com as preposições da língua portuguesa
        stopwords <- c(" a "," à "," agora "," ainda "," alguém "," algum
"," alguma "," algumas "," alguns "," ampla ",
        ## continuação da lista...
            " amplas "," amplo "," amplos "," ante "," antes "," ao "," aos
"," após "," aquela "," aquelas ",
        ## continuação da lista...
            " aquele "," aqueles "," aquilo "," as "," até "," através ","
cada "," coisa "," coisas "," com ",
        ## continuação da lista...
            " como "," contra "," contudo "," da "," daquele "," daqueles
"," das "," de "," dela "," delas ",
        ## continuação da lista...
            " dele "," deles "," depois "," dessa "," dessas "," desse ","
desses "," desta "," destas ",
        ## continuação da lista...
            " deste "," deste "," destes "," deve "," devem "," devendo ","
dever "," deverá "," deverão ",
        ## continuação da lista...
            " deveria "," deveriam "," devia "," deviam "," disse "," disso
"," disto "," dito "," diz ",
        ## continuação da lista...
            " dizem "," do "," dos "," e "," é "," e' "," ela "," elas ","
ele "," eles "," em "," enquanto ",
        ## continuação da lista...
            " entre "," era "," essa "," essas "," esse "," esses "," esta
"," está "," estamos "," estão ",
        ## continuação da lista...
            " estas "," estava "," estavam "," estávamos "," este "," estes
"," estou "," eu "," fazendo ",
        ## continuação da lista...
            " fazer "," feita "," feitas "," feito "," feitos "," foi ","
```

```
for ","foram "," fosse "," fossem "," grande ",
        ## continuação da lista...
            " grandes "," há "," isso "," isto "," já "," la "," la "," lá
"," lhe "," lhes "," lo "," mas ",
        ## continuação da lista...
            " me "," mesma "," mesmas "," mesmo "," mesmos "," meu "," meus
"," minha "," minhas "," muita ",
        ## continuação da lista...
            " muitas "," muito "," muitos "," na "," não "," nas "," nem ","
nenhum "," nessa "," nessas ",
        ## continuação da lista...
            " nesta "," nestas "," ninguém "," no "," nos "," nós "," nossa
"," nossas "," nosso "," nossos ",
        ## continuação da lista...
            " num "," numa "," nunca "," o "," os "," ou "," outra ","
outras "," outro "," outros "," para ",
        ## continuação da lista...
            " pela "," pelas "," pelo "," pelos "," pequena "," pequenas ","
pequeno "," pequenos "," per ",
        ## continuação da lista...
            " perante "," pode "," pôde "," podendo "," poder "," poderia
"," poderiam "," podia "," podiam ",
        ## continuação da lista...
            " pois "," por "," porém "," porque "," posso "," pouca ","
poucas "," pouco "," poucos ",
        ## continuação da lista...
            " primeiro "," primeiros "," própria "," próprias "," próprio
"," próprios "," quais "," qual ",
        ## continuação da lista...
            " quando "," quanto "," quantos "," que "," quem "," são "," se
"," seja "," sejam "," sem ",
        ## continuação da lista...
            " sempre "," sendo "," será"," serão "," seu "," seus "," si ","
sido "," só "," sob "," sobre ",
        ## continuação da lista...
            " sua "," suas "," talvez "," também "," tampouco "," te "," tem
"," tendo "," tenha "," ter ",
        ## continuação da lista...
            " teu "," teus "," ti "," tido "," tinha "," tinham "," toda ","
todas "," todavia "," todo ",
        ## continuação da lista...
            " todos "," tu "," tua "," tuas "," tudo "," última "," últimas
"," último "," últimos "," um ",
        ## continuação da lista...
            " uma "," umas "," uns "," vendo "," ver "," vez "," vindo ","
vir "," vos "," vós ")
    }

        ## cria uma condição que verifica se o idioma escolhido foi
'english'
```

```
    if(language == "english"){
        ## cria uma lista com as preposições da língua inglesa
        stopwords <- c(" a ", " about "," above "," according "," across "," 
actually "," after "," again "," against "," all "," almost "," along ",
        ## continuação da lista...
            " already "," also "," although "," always "," among "," an "," 
and "," another "," any "," anything "," are "," aren ",
        ## continuação da lista...
            " as "," at "," away "," back "," back "," be "," because "," 
been "," before "," behind "," being "," below "," besides ",
        ## continuação da lista...
            " better "," between "," beyond "," both "," but "," by "," can 
"," certain "," could "," do "," does "," during "," each ",
        ## continuação da lista...
            " else "," enough "," even "," ever "," few "," for "," from "," 
further "," get "," going "," got "," great "," has "," have ",
        ## continuação da lista...
            " he "," her "," here "," high "," his "," how "," however "," i 
"," if "," in "," instead "," into "," is "," it "," its "," itself ",
        ## continuação da lista...
            " just "," later "," least "," less "," less "," let "," little 
"," many "," may "," maybe "," me "," might "," more "," most ",
        ## continuação da lista...
            " much "," must "," neither "," never "," new "," no "," non "," 
nor "," not "," nothing "," of "," off "," often "," old "," on ",
        ## continuação da lista...
            " once "," one "," only "," or "," other "," our "," out "," 
over "," perhaps "," put "," rather "," really "," set "," several ",
        ## continuação da lista...
            " she "," should "," since "," snot "," snt "," so "," some "," 
something "," sometimes "," soon "," still "," such "," t "," than ",
        ## continuação da lista...
            " that "," the "," their "," them "," then "," there "," 
therefore "," these "," they "," thing "," this "," those "," though ",
        ## continuação da lista...
            " three "," through "," till "," to "," together "," too "," 
toward "," towards "," two "," under "," up "," upon "," us "," very ",
        ## continuação da lista...
            " very "," was "," were "," what "," when "," where "," whether 
"," which "," while "," whole "," whose "," will "," with "," within ",
        ## continuação da lista...
            " without "," would "," yet "," you "," your ")
    }

        ## cria uma condição que verifica se o idioma escolhido foi 
'spanish'
    if(language == "spanish"){
        ## cria uma lista com as preposições da língua espanhola
        stopwords <- c(" a "," acá "," adelante "," además "," ahí "," ahora 
"," al "," alguna "," algunas "," alguno "," algunos "," as "," así "," aún 
",
```

```
        ## continuação da lista...
            " bastante "," bien "," cada "," como "," con "," cosa "," cosas
"," cual "," cuál "," cuales "," cuáles "," cuando "," da ",
        ## continuação da lista...
            " de "," debe "," debemos "," del "," dentro "," desde ","
después "," donde "," dos "," durante "," e "," el "," ellos "," en ",
        ## continuação da lista...
            " entonces "," entre "," era "," eran "," es "," esa "," esas
"," ese "," eso "," esos "," esta "," está "," estaba "," estaban ",
        ## continuação da lista...
            " estamos "," están "," estar "," estas "," este "," esto ","
estos "," frente "," fue "," fuera "," fueron "," ha "," haber ",
        ## continuação da lista...
            " había "," hace "," hacen "," hacer "," hacia "," haciendo ","
han "," hasta "," hay "," hecho "," hoy "," la "," las "," le ",
        ## continuação da lista...
            " les "," lo "," los "," me "," mientras "," misma "," mismas
"," mismo "," mismos "," mucha "," muchas "," mucho "," muchos ",
        ## continuação da lista...
            " muy "," ni "," no "," nos "," nuestra "," nuestras "," nuestro
"," nuestros "," o "," ó "," os "," otra "," otras "," otro ",
        ## continuação da lista...
            " otros "," para "," pero "," podemos "," podría "," por ","
porque "," puede "," pueden "," pues "," que "," qué "," se "," sea ",
        ## continuação da lista...
            " según "," segundo "," ser "," sería "," si "," sido "," siendo
"," sobre "," solamente "," son "," su "," tal "," también "," tan ",
        ## continuação da lista...
            " tanto "," tenemos "," tener "," tenían "," tiene "," tienen
"," toda "," todas "," todavía "," todo "," todos "," través "," un ",
        ## continuação da lista...
            " una "," uno "," va "," vamos "," van "," veces "," vemos ","
ver "," vez "," y "," ya ")
    }

            ##
            ## 2c. continua a limpar os dados
            ## Exclui as preposições do vetor
            ##

        ## inicia o ciclo para excluir as preposições
    for(i in 1:length(stopwords)){
        ## exclui os pronomes do primeiro vetor
        temp.a <- gsub(stopwords[i], " ", temp.a)
        ## exclui os pronomes do segundo vetor
        temp.b <- gsub(stopwords[i], " ", temp.b)
    }

            ##
            ## 3. separa as palavras em diferentes vetores.
```

```
            ## É importante notar que o vetor se transforma em uma lista
            ##

        ## separa as palavras do primeiro vetor
    temp.a <- strsplit(temp.a, " ")
        ## separa as palavras do segundo vetor
    temp.b <- strsplit(temp.b, " ")

        ## exclui campos vazios do primeiro vetor
    temp.a <- temp.a[[1]][temp.a[[1]] != ""]
        ## exclui campos vazios do segundo vetor
    temp.b <- temp.b[[1]][temp.b[[1]] != ""]

            ##
            ## * conta o número de palavras que cada vetor possui após
excluir os preposições
            ##

            ## conta o número de palavras no primeiro vetor
        count.a2 <- length(temp.a)
            ## conta o número de palavras no segundo vetor
        count.b2 <- length(temp.b)

        ##
        ## 4. conta o número de vezes que cada palavras aparece por vetor
        ##

        ## cria um vetor com as palavras utilizadas (termos únicos)
    termos.unicos <- unique(c(temp.a, temp.b))

        ## declara a variável relativa ao primeiro vetor
    n.termos.a <- NULL
        ## declara a variável relativa ao segundo vetor
    n.termos.b <- NULL
        ## declara a variável das palavras
    termos <- NULL
        ## inicia o indexador com zero
    k <- 0
        ## inicia o ciclo para contar o número de vezes que cada
        ## palavra aparece em cada vetor
    for(i in termos.unicos){
        ## atualiza o indexador
        k <- k + 1
        ## conta o número de vezes que cada palavra aparece no primeiro
vetor
        n.termos.a[k] <- sum(temp.a == i)
        ## conta o número de vezes que cada palavra aparece no segundo vetor
        n.termos.b[k] <- sum(temp.b == i)
        ## indica a palavra
        termos[k] <- i
    }
```

```
        ## cria a tabela com as contagens
    tab <- data.frame(termos, n.termos.a, n.termos.b)

        ##
        ## 5. calcula a porcentagem de cada palavra em cada vetor
        ##

        ## calcula a porcentagem de cada palavra no primeiro vetor
    tab$pa <- tab$n.termos.a / sum(tab$n.termos.a) * 100
        ## calcula a porcentagem de cada palavra no segundo vetor
    tab$pb <- tab$n.termos.b / sum(tab$n.termos.b) * 100

        ##
        ## 6. calcula o índice de similaridade
        ##

        ## declara a variável que irá guardar o índice
    dif.abs <- NULL
        ## inicia o ciclo para calcular o índice
    for(i in 1:nrow(tab)){
        ## calcula o módulo das diferenças entre pa e pb
        dif.abs[i] <- abs(tab$pa[i] - tab$pb[i])
    }

        ## calcula o índice de similaridade
    similaridade <- 100 - sum(dif.abs)/2

        ##
        ## 7. Mensagens a serem exibidas
        ##

        ## indica o número de palavras que possui o primeiro vetor
        ## (excluindo os números, pontuação e caracteres especiais)
    cat("\tO primeiro vetor possui ", count.a, "palavras.\n")
        ## indica o número de palavras que possui o segundo vetor
        ## (excluindo os números, pontuação e caracteres especiais)
    cat("\tO segundo vetor possui ", count.b, "palavras.\n\n")

        ## indica o número de palavras que possui o primeiro vetor
        ## (excluindo as preposições)
    cat("\tExcluindo as preposições, o primeiro vetor possui ", count.a2,
"palavras.\n")
        ## indica o número de palavras que possui o segundo vetor
        ## (excluindo as preposições)
    cat("\tExcluindo as preposições, o segundo vetor possui ", count.b2,
"palavras.\n\n")

        ## indica o número de termos únicos que o primeiro vetor possui
    cat("\tO primeiro vetor possui ",length(unique(temp.a)), " termos
únicos.\n")
```

```
        ## indica o número de termos únicos que o segundo vetor possui
    cat("\tO segundo vetor possui ",length(unique(temp.b)), " termos
únicos.\n\n\t\t***\n")

        ## indica o índice de similaridade entre os dois textos
    cat("\t\tA similidade entre os textos foi de ", similaridade, "%.\n")

        ##
        ## 8. Essa é a lista de valores a serem guardados pela função
        ##

    ## cria uma lista com o índice de similaridade e com a frequência das
palavras
    final <- list(similaridade, tab)
    ## renomeia os elementos da lista
    names(final) <- c("similar", "freq")
    ## lista de resultados
    final
}
```

Help da função

Função

Plano B - função para fazer um gráfico com os coeficientes de regressão linear

FUNÇÃO: plot.lm(model, conf.int = .95)

DESCRIÇÃO: No meio acadêmico, a utilização de gráficos ao invés de tabelas tem sido cada vez maior. O objetivo dessa função será fazer um gráfico dos coeficientes (juntamente com os seus intervalos de confiança) de um modelo de regressão linear.

INPUT: A entrada da função será o resultado de um modelo de regressão linear. Por exemplo, se o modelo for:

model ← lm(y ~ x + z),

a entrada será 'model'. Além disso, o usuário terá a opção de escolher o nível de significância (o default será .95).

OUTPUT: A função exibirá um gráfico (dotplot) com com coeficientes do modelo e o seu intervalo de confiança.

> Essa parece uma ideia simples e talvez não mto empolgante como a do Plano A.
>
> —-*Thais Lopes*