Divulgação do R para linguistas[1]

Júlia Vidigal Zara - Universidade Federal de Minas Gerais

RESUMO: Este artigo objetiva divulgar o uso do software R para estudos linguísticos, sendo um ponto de partida para linguistas que ainda não conhecem o programa. Como uma introdução geral ao R, as suas vantagens em relação a outros programas disponíveis para a análise de corpus são inicialmente destacadas. Em seguida, informações básicas para a instalação e iniciação do uso do programa são apresentadas. Finalmente, um exemplo prático de como o programa pode ser utilizado em análises de corpora é discutido. Neste exemplo, é mostrado como carregar no R um arquivo externo para a manipulação de seus dados, resultando na geração de uma lista de frequência de palavras. Espera-se, com este trabalho, despertar o interesse de estudantes e pesquisadores para conhecerem mais sobre o R.
PALAVRAS CHAVE: Estudos Linguísticos, Análise de Corpus, Software R.

INTRODUÇÃO

Este artigo objetiva divulgar o uso do software R para estudos linguísticos, sendo um ponto de partida para linguistas que ainda não conhecem o programa. Em específico, será mostrada uma aplicação do uso do R para estudos baseados em corpora. Um corpus corresponde a um conjunto estruturado de textos escritos ou orais, eletronicamente armazenados e processados, que foram coletados de acordo com critérios específicos (conteúdo, gênero, tipologia, registro etc.) para fins de análise linguística (McEnery & Wilson, 2001). A análise de corpus traz uma série de vantagens para os estudos linguísticos (Biber et al., 1998; Kennedy, 1998; McEnery & Wilson, 2001; McEnery et al., 2006), dentre elas a possibilidade de se pesquisar, recuperar, organizar e realizar cálculos com um número de dados linguísticos antes inimaginável. Além disso, através da observação do uso real da língua, a análise de corpus apresenta-se como uma abordagem mais sistemática e objetiva para a análise da linguagem em comparação a métodos introspectivos.

Apesar de existirem diversos programas disponíveis para a análise de corpora, o R é um software que tem ganhado cada vez mais adeptos e colaboradores (Peternelli & Mello, 2011). Além de ser um software livre, ou seja, gratuito, o R é uma ferramenta muito versátil, permitindo, em um só ambiente, a manipulação de dados, a realização de cálculos e a geração de gráficos (Gries, 2009). O R permite também maior controle por parte do pesquisador sobre a análise que está sendo feita. Através de scripts, isto é, de conjuntos de comandos que "dizem" ao R "o que fazer", pode-se, por exemplo, definir como uma busca específica por dados é realizada em um arquivo de texto com características específicas.

Na seção abaixo são apresentadas informações básicas para a instalação e iniciação de uso do R. Em seguida é mostrado como usar o programa para gerar uma lista de frequência de palavras de um arquivo de corpus etiquetado.

[1] X EVIDOSOL e VII CILTEC-Online - junho/2013 - http://evidosol.textolivre.org

1. UTILIZANDO O R EM ESTUDOS LINGUÍSTICOS

O R pode ser obtido na página do projeto: http://www.r-project.org. Para a sua instalação, clica-se na opção "Download, Packages CRAN" no *menu* à esquerda da tela. Em seguida, escolhe-se o local de disponibilização do programa (CRAN Mirrors) (exemplo: Universidade Federal do Paraná). O próximo passo é baixar e instalar o programa. Em "Download and Install R", deve-se clicar primeiramente no link correspondente ao sistema operacional do computador no qual se quer instalar o software (exemplo: Download R for Windows) e, em seguida, no link "base". Por fim, escolhe-se a versão do arquivo que se quer executar (exemplo: Download R 2.15.3 for Windows) e segue-se a rotina de instalação.

Ao abrir o programa, visualiza-se uma janela na qual o símbolo ">" indica que o R está pronto para receber comandos. Um comando consiste em uma função (instrução sobre o que fazer) seguida de argumentos, que especificam a que a função será aplicada e, em alguns casos, como será aplicada. Os argumentos devem vir entre parênteses e, se houver mais de um, devem ser separados por vírgulas. Por exemplo, no comando "tolower(x)", a função "tolower" diz ao R para converter para letras minúsculas todas as palavras contidas em seu argumento x, que representa uma estrutura de dados pré-existente. O tipo mais simples de estrutura de dados no R é chamada de "vetor", e, por limitações de espaço, é a única que será comentada neste artigo.

Para criar vetores no R, dá-se um nome para o vetor e atribui-se a ele um valor correspondente a uma sequência de números ou de caracteres de palavras. A atribuição de valores pode ser feita de duas formas: através do sinal "=" ou do sinal "<-". Por exemplo, para criar um vetor x com o valor "Eu sou uma Pesquisadora", escreve-se no R: x<-"Eu sou uma Pesquisadora" ou x="Eu sou uma Pesquisadora". Executando-se a função "tolower", supracitada, no vetor x, o seguinte resultado é obtido: "eu sou uma pesquisadora".

Além das funções do R que fazem parte de sua configuração padrão, conjuntos de funções chamados de "pacotes" podem ser adicionados à "biblioteca" (library) do R de acordo com os interesses de cada pesquisador. Informações acerca das funções de um pacote são obtidas através da função library(help=" "), colocando-se o nome do pacote entre aspas. Para instalar e baixar um pacote, as funções "install.packages()" e "library()" são utilizadas. Os argumentos destas funções são os pacotes que se quer utilizar. Com os pacotes já instalados, pode-se executar a função help() para se obter ajuda sobre uma função específica. A execução do comando help(toupper), por exemplo, abre uma janela com informações sobre a função "toupper".

Como mencionado acima, o R é uma ferramenta muito versátil que possibilita diversas formas de manipulação de dados e de cálculos estatísticos. Uma vez que este artigo é apenas uma introdução ao assunto e devido à limitação de espaço, irei me restringir aqui a uma operação de interesse comum entre linguistas de corpus: a geração de listas de frequência de palavras de arquivos de corpora. Para ilustrar como isto pode ser feito através do R, utilizarei como exemplo um arquivo do componente britânico do International Corpus of English (ICE-GB), um corpus etiquetado com informações sobre classes gramaticais, estrutura sintática e traços morfológicos. A figura abaixo mostra parte do arquivo:

```
 NPHD,N(prop,sing) {J. N. L. Myres}
VB,VP(montr,pres)
 MVB,V(montr,pres) {begins}
OD,NP
```

```
DT,DTP
DTCE,PRON(poss,sing) {his}
```

Figura 1. Parte de um arquivo do corpus ICE-GB.

As etapas da geração de uma lista de frequência de palavras de um arquivo como este no R são: (I) carregar o arquivo no R; (II) converter todos os caracteres para letra minúscula; (III) eliminar as informações não desejadas para a lista de frequência; (IV) gerar a lista de frequência. Os comandos necessários para executar estas operações são detalhados a seguir. Todas as funções citadas estão no pacote "base" do R. O símbolo # é utilizado para comentar os comandos executados no R. Tudo que segue este símbolo é ignorado pelo programa.

Através da função "setwd", define-se o diretório de trabalho com o qual se quer trabalhar.

```
> setwd("C:/Users/Júlia/Desktop/Essays") #definir diretório de trabalho
```

O arquivo de corpus a ser analisado (em formato .txt) é então carregado no R através da função "scan". Os argumentos desta função são: o arquivo externo ao R ("file=choose.files()" abre uma janela para se selecionar este arquivo), o tipo de dado contido neste arquivo (exemplo: char = caractere) e o caractere que delimita a separação entre os elementos do arquivo (exemplo: \n = nova linha). Os resultados dos comandos serão sempre salvos em novos vetores (exemplo: criação do vetor "arquivo").

```
> arquivo<-scan(file=choose.files(), what="char",sep="\n") #carregar arquivo de corpus no R
```

Geralmente não é interessante diferenciar palavras que começam com letras maiúsculas de palavras que começam com letras minúsculas em listas de frequência e, por isto, converte-se todo o arquivo para letras minúsculas através da função "tolower", e salva-se o resultado em um novo vetor ("minuscula"):

```
> minuscula<-tolower(arquivo)
```

Eliminar informações do arquivo de corpus que não se deseja em uma lista de frequência é outro passo importante. Para eliminar a pontuação, uma das possibilidades é criar um vetor com os sinais de pontuação (vetor "pontuação") e, em seguida, criar um novo vetor, que chamo abaixo de "palavras", com os elementos do vetor que já existia anteriormente ("minúscula") menos os elementos do vetor com os sinais de pontuação:

```
> pontuação<-grep("punc,punc(.+)\\{.+\\}", minuscula, perl=TRUE, value=TRUE)

> palavras<-minuscula[!(minuscula %in% pontuação)]
```

No primeiro comando acima, a função "grep" diz ao R para buscar os sinais de pontuação, representados pela expressão regular "punc,punc(.+)\\{.+\\}", no vetor "minuscula". Uma expressão regular corresponde a um padrão que o programa irá buscar em uma estrutura de dados. O argumento pert=TRUE indica que a expressão regular utilizada é compatível e deve ser interpretada segundo a linguagem de programação Perl. Nesta linguagem, alguns caracteres possuem sentidos não literais. No exemplo dado, o ponto "." significa "qualquer caractere exceto uma nova linha", e o caractere "+" significa "uma ou mais ocorrências da expressão regular anterior". Quando queremos que um caractere com significado não literal na linguagem Perl seja interpretado de forma literal pelo programa,

precisamos utilizar o caractere \\ antes dele. Assim, em "punc,punc(.+)\\{.+\\}", o programa interpretará os caracteres "{" e "}", antecedidos por \\, literalmente. O último argumento da função "grep", o "value", especifica se os elementos encontrados devem ser retornados nos resultados ou se apenas as suas posições no vetor devem ser informadas.

No segundo comando acima, cria-se um vetor ("palavras") que irá conter os elementos do vetor "minuscula" que não (negação representada pelo sinal "!") sejam os elementos de "minuscula" que também se fazem presentes (%in%) no vetor pontuação. Assim, tem-se agora um novo vetor, "palavras", livre dos sinais de pontuação.

Antes da geração da lista de frequência deve-se selecionar apenas as linhas do arquivo que possuem palavras e, em seguida, excluir as etiquetas que acompanham as palavras. Já que cada palavra aparece no corpus entre { }, usamos a função "grep" para buscarmos apenas as que possuam o caractere" "{" no vetor "palavras":

```
> apenas.palavras<-grep("\\{", palavras, perl=TRUE, value=TRUE)
```

Agora é necessário eliminar tudo que aparece antes e depois das palavras na linha em que eles ocorrem, o que pode ser feito através da função "gsub", utilizada para realizar substituições. O primeiro argumento desta função é o padrão que será substituído pelo padrão especificado no segundo argumento. Estes dois argumentos devem ser colocados entre aspas. O terceiro argumento define o vetor no qual a substituição deve ser feita. No exemplo abaixo, o vetor "limpar" foi criado para armazenar o resultado da substituição (gsub) do caractere "{" e de tudo o que ocorre na mesma linha antes deste caractere ("^.*?\\{") por nenhum caractere ("") no vetor já existente "apenas.palavras".

```
> limpar<-gsub("^.*?\\{", "", apenas.palavras, perl=TRUE)
```

Um comando semelhante é utilizado para eliminar o caractere "}" que segue as palavras nas linhas do arquivo de corpus. Um novo vetor, "limpar.2", é criado para armazenar o resultado deste comando:

```
> limpar.2<-gsub("\\}", "", limpar, perl=TRUE)
```

A lista de frequência pode finalmente ser gerada através do comando "table", cujo argumento é o vetor contendo as palavras que devem compor a lista de frequência:

```
> frequencia<-table(limpar.2)
```

Para organizar as palavras da lista de frequência em ordem decrescente de frequência, a função "sort" é utilizada com o argumento "decreasing" como "verdadeiro" (TRUE):

```
> lista.frequencia<-sort(frequencia, decreasing=T)
```

As frequências geradas a partir dos comandos acima são os valores do vetor "lista.frequencia", e os nomes dos valores são as palavras. É necessário, então, combinar os valores e os nomes antes de se imprimir a lista de frequência em um arquivo externo ao R. Para isto, utiliza-se a função "paste" e define-se qual caractere separará as palavras de suas frequências (sep="\t" significa que o separador é um tab).

```
> sorted.table<-paste(names(lista.frequencia), lista.frequencia, sep="\t")
```

Finalmente, através da função “cat”, o conteúdo de “sorted.table” é salvo em um arquivo externo ao R.

```
> cat(sorted.table, file="lista de frequencia.txt",what="char",sep="\n")
```

Os argumentos de “cat” são: o vetor que se quer imprimir, o arquivo no qual se quer imprimir, o tipo de elemento presente no vetor e o caractere que irá separar cada elemento.

CONCLUSÃO

Este artigo foi uma breve introdução ao programa R para linguistas. Foram destacadas as vantagens do R para estudos linguísticos, e noções básicas para a instalação e uso inicial do programa foram fornecidas. Um exemplo prático de aplicação do R para estudos linguísticos foi então ilustrado através da geração de uma lista de frequência de palavras de um arquivo de corpus etiquetado.

Escrever scripts do R pode demandar trabalho inicialmente. Porém, depois de pronto, um mesmo script, como o discutido acima, pode ser utilizado inúmeras vezes pelo mesmo pesquisador em pesquisas diferentes ou por outros pesquisadores. Além disso, com o R o linguista tem a flexibilidade de informar ao programa como análises específicas em grandes bancos de dados com características particulares devem ser feitas.

Aprender a utilizar o R pode ser comparado a aprender uma segunda língua: os conhecimentos adquiridos abrem portas para um novo mundo de possibilidades. Espero que este artigo tenha instigado a curiosidade de pesquisadores para conhecerem mais sobre o R.

REFERÊNCIAS

BIBER, Douglas; CONRAD, Susan; REPPEN, Randi. Corpus linguistics: investigating language structure and use. New York: Cambridge University Press, 1998.

GRIES, Stefan Thomas. Quantitative Corpus Linguistics with R: A Practical Introduction. New York: Routledge, 2009.

KENNEDY, Graeme. An introduction to corpus linguistics. New York: Longman, 1998.

MCENERY, Tony; WILSON, Andrew. Corpus Linguistics: An Introduction. Edimburgo: Edinburgh University Press, 2001.

MCENERY, Anthony; XIAO, Richard; TONO, Yukio. Corpus-based language studies: an advanced resource book**.** New York: Routledge, 2006. 386 p

PETERNELLI, Luiz Alexandre; MELLO, Márcio Pupin de. Conhecendo o R: uma visão estatística. Série Didática. Minas Gerais: Editora UFV, 2011.

INTERNATIONAL CORPUS OF ENGLISH – British Component. URL <http://www.ucl.ac.uk/english-usage/projects/ice-gb/>. Acessado em 02/03/2013.

The R Project for Statistical Computing. URL http://www.r-project.org/. Acessado em 02/03/2013.