Pesquisa Literária com R: Análise Quantitativa de Dados Textuais, Quanteda tomando como exemplo o *Livro do Desassossego*

Literary research using R language: Quantitative Analysis of Textual Data, Quanteda, taking the book *Livro do Desassossego* as an example

Diego Giménez
Andressa Gomide

Diego Giménez
Universidade de Coimbra, Centro de Literatura Portuguesa
ORCID: 0000-0002-1229-3969

Andressa Gomide
Universidade de Coimbra, Centro de Estudos de Linguística Geral e Aplicada
ORCID: 0000-0002-1481-4748

https://doi.org/10.14195/1647-8622_22_7

PESQUISA LITERÁRIA COM R: ANÁLISE QUANTITATIVA DE DADOS TEXTUAIS, QUANTEDA TOMANDO COMO EXEMPLO O *LIVRO DO DESASSOSSEGO*

O presente artigo pretende oferecer uma metodologia de pesquisa com o pacote Quanteda, que utiliza a linguagem R, aplicada à análise da obra de Fernando Pessoa. Quanteda (Quantitative Analysis of Textual Data) é um pacote de R para a manipulação e estudo de dados textuais. O programa objetiva aplicar processamento de linguagem natural a textos. Por sua vez, R é uma linguagem de programação para computação estatística suportada pelo R Core Team e R Foundation for Statistical Computing. A ferramenta, assim, permite o estudo textual quantitativo de corpus e oferece ferramentas de visualização que representam as análises. Desde *topic modeling* até redes semânticas ou análises de coocorrências, as ferramentas possibilitam estudos e representações detalhados de estruturas textuais.

Palavras-chave: quanteda; r; Fernando Pessoa; textual data; leitura distante.

LITERARY RESEARCH USING R LANGUAGE: QUANTITATIVE ANALYSIS OF TEXTUAL DATA, QUANTEDA, TAKING THE BOOK *LIVRO DO DESASSOSSEGO* AS AN EXAMPLE

This article aims to provide a research methodology with the Quanteda package, which uses the R language, applied to the analysis of the work of Fernando Pessoa. Quanteda (Quantitative Analysis of Textual Data) is an R package for the manipulation and analysis of textual data. The program was created by R users who needed to apply natural language processing to texts. R is a programming language for statistical computing supported by the R Core Team and the R Foundation for Statistical Computing. The tool, therefore, allows the quantitative textual analysis of the corpus and offers visualization tools that represent the corpus analyses. From topic modeling to semantic networks or analysis of co-occurrences, the tools enable detailed studies and representations of textual structures.

Keywords: quanteda; r; Fernando Pessoa; textual data; distant reading.

RECHERCHE LITTÉRAIRE AVEC R : ANALYSE QUANTITATIVE DE DONNÉES TEXTUELLES, QUANTEDA PRENANT LE *LIVRE DE L'INQUIÉTUDE* COMME EXEMPLE

Cet article vise à proposer une méthodologie de recherche avec le module Quanteda, qui utilise le langage R, appliqué à l'analyse de l'œuvre de Fernando Pessoa. Quanteda (Analyse quantitative des données textuelles) est un module de R pour la manipulation et l'étude des données textuelles. Le programme vise à appliquer le traitement du langage naturel aux textes. De son côté, R est un langage de programmation pour le calcul statistique soutenu par la R Core Team et la R Foundation for Statistical Computing. L'outil permet donc une étude textuelle quantitative d'un corpus et propose des outils de visualisation qui représentent les analyses. Du *topic modeling* aux réseaux sémantiques ou à l'analyse des cooccurrences, les outils permettent des études et des représentations détaillées des structures textuelles.

Mots-clés : quanteda; r; Fernando Pessoa; données textuelles; lecture à distance.

Introdução

Desde que se popularizou o termo "Humanidades Digitais" em 2004 (Alves, 2016, p. 91), muito se têm desenvolvido as ferramentas e as metodologias utilizadas no estudo de corpus dentro das Humanidades. No presente texto, pretendemos especificar e descrever a análise de uma obra literária com o pacote Quanteda (Benoit et al., 2018), que utiliza a linguagem R de programação. Praticamente não há bibliografia em português que descreva análises textuais de obras literárias que usem a linguagem em questão. Desse modo, o objeto a examinar é o *Livro do Desassossego*[1], de Fernando Pessoa. Não é intenção deste artigo descrever a história de edição do *Livro*, nem analisar a estrutura narrativa da obra composta por fragmentos ou entrar em interpretações heteronímicas. Basta assinalar, de forma sumária, que o *Livro do Desassossego* está estruturado, ou desestruturado, por uma série de impressões em forma de trechos, "sem nexo nem desejo de nexo" (Pessoa, 2012, p. 58), articuladas por um narrador à procura de sentido na urbe moderna lisboeta do Portugal de princípios do século XX. Não queremos, assim, interpretar a obra pessoana, embora os resultados da análise textual possam aportar alguns dados literariamente relevantes. O texto visa descrever um modelo e metodologia de pesquisa que possa servir de ilustração para qualquer investigação similar mediante o pacote Quanteda. Dessa forma, o artigo aborda, em primeiro lugar, uma contextualização teórica sobre a interpretação de dados quantitativos; em um segundo momento, apresenta as ferramentas e metodologias utilizadas; posteriormente, expõe visualizações e análises de resultados; por fim, levanta as considerações finais.

As seções 2 e 3 introduzem e explicam parte dos códigos empregados nas análises. Os códigos estão expostos em uma caixa cinza e os resultados obtidos com a execução dos códigos são indicados com dois # que os precedem. Os códigos e dados utilizados neste trabalho estão disponíveis em um repositório online e aberto[2] que pode ser replicado mediante qualquer interface gráfica para R. O código permite a adaptação a qualquer corpus.

1. Interpretação de dados

Uma das dificuldades inerentes à pesquisa digital é a falta de descrição das ferramentas e dos métodos aplicados na análise computacional. O debate teórico que suscita a especificação metodológica é expresso por Johanna Drucker em *Visualization and Interpretation* (2020): "The interpretative dimensions of the activity that shaped the data are rendered invisible, not so much concealed as simply missing from view, absent without a trace" (p. 11). A teórica afirma que o trabalho com dados quantitativos não está isento de interpretação e modelação dos dados. As visualizações e os resultados restantes dos diferentes processos computacionais são modelados, e é preciso evidenciar esses processos

[1] A criação do corpus para a análise foi feita a partir da edição de Jacinto do Prado Coelho que consta no Arquivo LdoD, editado por Manuel Portela (2017). No ponto 2.3 há uma descrição da criação do corpus a partir do arquivo digital.

[2] https://github.com/andressarg/analise_lit_quanteda.

para tornar mais claros os resultados e para poder repetir as análises, caso se queira conferir a metodologia. Adiante, no exemplo do tópico 3.2, veremos a seleção das palavras irrelevantes que não queremos que sejam contadas na análise e cuja configuração condiciona o produto do computo. Também, a descrição do funcionamento dos algoritmos, os quais, muitas vezes, por serem técnicos demais, podem resultar obscuros.

Uma vez especificado o percurso, podemos considerar os dados, que não devem ser analisados como factos incontestáveis, mas como construções. Da mesma forma, os dados por si só não conferem sentido à obra. É preciso interpretar os resultados nos marcos estéticos, teóricos ou históricos correspondentes. Jurgen Renn (2020), em *The Evolution of Knowledge*, afirma:

> This prospect, combined with the capacities of the information technologies, also includes new possibilities of dealing with the challenges of quantitative analyses and large data sets in the social sciences and the humanities. Network analysis has become an important research instrument in this context, but it has to be treated with some care because of the risk of blurring conceptual distinctions or losing the intellectual depth of other traditions and approaches in the humanities (p. 303).

O apontamento de Renn, que incide em não perdermos a profundidade das análises tradicionais, chama a atenção para o debate entre *close* e *distant reading*.[3] Como aponta também Ted Underwood, em *Distant Horizons* (2019), ao alertar para os riscos do *distant reading*, ao início do século vinte acreditava-se que os métodos quantitativos pudessem ser introduzidos nas humanidades de forma tranquila. Seriam construídas ferramentas e todos iriam usá-las. Os académicos não precisariam entender os detalhes das ferramentas, etc. Mas a realidade demonstrou ser outra. O convívio entre as diferentes metodologias de pesquisa tem sido lento e não isento de conflitos:

> The reason, I think, is that new methods have turned out to be more consequential than was widely believed a decade ago. Search engines can be encapsulated and treated as tools. But statistical models are not well envisioned as tools: they offer new methods of representing and interpreting the world. Scholars cannot adopt a new mode of interpretation without fully understanding the reasoning it implies (Underwood, 2019, p. 145).

Os modelos quantitativos de análise são novas formas de representar o mundo, proposição que vem demonstrada pela quantidade de investigações em todas as áreas do conhecimento que estão a aplicar métodos quantitativos de análise e representação. Tanto Undewood como Drucker pedem que sejam especificados os caminhos com os

[3] No artigo "Leitura distante em português: resumo do Primeiro Encontro" (2020) vários autores refletem sobre o Primeiro Encontro sobre Leitura Distante em Português que aconteceu na Universidade de Oslo a 27 e 28 de outubro de 2019. No texto é definida a leitura distante: "A leitura distante (em inglês, distant reading) é uma área interdisciplinar específica e em crescente evolução que combina os domínios dos Estudos Literários, da Linguística Computacional e da Informática Aplicada na análise de grandes coleções de textos, que, pela sua natureza, compreende dados de volume significativo. Os primeiros trabalhos desenvolvidos nesta área preocuparam-se com textos literários (Moretti, 1999, 2005), mas os seus usos não se restringem a este tipo de fontes" (Santo et al., 2020, p. 280).

quais modelamos e pesquisamos a informação. Os dados extraídos são construídos. A seleção do corpus, sua preparação para a análise, a própria análise e a limpeza dos resultados modelam aquilo que se está a estudar e a visualizar. A análise quantitativa, vista sob este aspeto, não descobre significado, mas o cria. Daí que a intenção do artigo seja especificar a metodologia e os passos a seguir com a linguagem R. Neste contexto podemos, assim, afirmar com Galloway e Thacker (2007), que "today to write theory is to write code" (p. 100)[4], na medida em que é preciso descrever o processo mediante o qual damos forma aos dados.

2. Ferramentas, Método

2.1. Instalação

Quanteda (Quantitative Analysis of Textual Data) é um pacote de R para a manipulação e análise de dados textuais. O programa, em código aberto, foi desenvolvido por Kenneth Benoit, Kohei Watanabe e outros colaboradores. Seu desenvolvimento inicial foi apoiado pela concessão do Conselho Europeu de Pesquisa ERC-2011-StG 283794-QUANTESS. Quanteda foi criado para usuários de R que precisam aplicar processamento de linguagem natural a textos. Por sua vez, R é uma linguagem de programação para computação estatística suportada pelo R Core Team e R Foundation for Statistical Computing. Criado pelos estatísticos Ross Ihaka e Robert Gentleman, o R é usado entre mineradores de dados e estatísticos para análise de dados e desenvolvimento de software estatístico. Os usuários podem criar pacotes ou bibliotecas (como é o caso do Quanteda) que contém dados, códigos e documentação que auxiliam na replicabilidade de estudos e evitam que códigos já criados anteriormente sejam desnecessariamente "reinventados". O ambiente de software R oficial é um ambiente de software livre de código aberto dentro do pacote GNU, disponível sob a GNU General Public License. Ele é escrito principalmente em C, Fortran e R (parcialmente auto-hospedado). A instalação do R varia de acordo com o sistema operacional (ex.: Windows, Mac, Linux) bem como suas diferentes versões. Há várias fontes onde se pode obter instruções atualizadas de como instalar o R[5]. O Comprehensive R Archive Network (CRAN)[6], a rede oficial de distribuição do R, oferece instruções confiáveis para tal, porém, talvez não tão detalhada como em outras fontes. Uma outra sugestão é instalar uma interface gráfica do utilizador, do inglês Graphical User Interface (GUI). As GUIs facilitam consideravelmente a interação do usuário com o computador. O RStudio[7] é a GUI mais utilizada para R, e, assim como o R, é gratuita e possui o código aberto.

---

[4] Galloway e Thacker (2007) falam em *The Exploit* sobre escrever teoria em um contexto diferente, ao falar das redes e dos protocolos de controle e os contraprotocolos em uma leitura deleuziana das redes. Para os autores, a vida enquanto resistência aos protocolos de controle se abre nas linhas de fuga, ou furos, dos protocolos computacionais.

[5] https://didatica.tech/como-instalar-a-linguagem-r-e-o-rstudio/

[6] https://cran.r-project.org/

[7] https://www.rstudio.com/

2.2. Configuração: preparando o ambiente

Ao reutilizar códigos, é uma boa prática estar atento à versão instalada tanto do R quanto das bibliotecas utilizadas. Não é necessário que as versões sejam as mesmas daquelas utilizadas durante a criação dos códigos, entretanto, em alguns casos, pode não haver compatibilidade entre versões diferentes e algumas funções ou pacotes podem ter sido descontinuados. Este artigo foi escrito utilizando a versão 4.2.0 do R.

Para nossa análise, utilizaremos alguns pacotes já existentes. Estes pacotes nada mais são que extensões para o R que normalmente contém dados ou códigos. Para utilizá-los, precisamos instalá-los no computador, caso ainda não tenha sido feito, e carregá-lo ao R. Uma vantagem de carregar apenas os pacotes necessários (em vez de todos os pacotes instalados) é evitar processamento computacional desnecessário. O código abaixo cria uma lista dos pacotes utilizados na presente análise e os carrega, instalando os que ainda não estavam presentes.

```
# Listamos os pacotes que precisamos
packages = c("quanteda", "quanteda.textmodels", "quanteda.textstats",
"quanteda.textplots",
             "newsmap", # para classificar documentos, com base em
"seed words"
             "readtext", # para ler diferentes formatos de texto
             "spacyr", # para etiquetação de classes gramaticais,
reconhecimento de entidades reconhecidas, analisador de
dependência (o python deve estar instalado)
             "ggplot2", # para exibir gráfico simples de frequências
             "seededlda" # para modelagem de tópicos
             )

# Instalamos (se necessário) e carregamos os pacotes
package.check <-lapply(
  packages,
  FUN = function(x) {
    if (!require(x, character.only = TRUE)) {
      install.packages(x, dependencies = TRUE)
      require(x, character.only = TRUE)
    }
  }
)
```

Os códigos abaixo foram implementados na versão 3.2.1 do Quanteda. Utilizar uma versão diferente dessa pode resultar em erros ou resultados indesejados. Para verificar qual é a versão dos pacotes, empregamos a função `packageVersion`. Para verificar a versão do R, utilizamos `R.version.string`.

```
packageVersion("quanteda")
## [1] '3.2.1'
R.version.string
## [1] "R version 4.2.0 (2022-04-22 ucrt)"
```

Por fim, precisamos estabelecer qual será nosso diretório de trabalho. Este será o local onde os resultados serão salvos. Para identificar qual é o diretório de trabalho selecionado, utilizamos `getwd()`. Esta função retorna o caminho absoluto, i.e., o endereço completo, do diretório. Para definirmos o novo local de trabalho, utilizamos a função `setwd()`. Arquivos salvos nesse diretório podem ser lidos apenas com a indicação do nome do arquivo. Isto porque podemos utilizar o caminho relativo, ou seja, o endereço onde o arquivo está salvo a partir do diretório em que estamos trabalhando.

2.3. Dados

Uma vez instalados os pacotes necessários, pode-se proceder à análise do corpus. Para isso, precisamos carregar o corpus no R. Se estamos trabalhando com dados armazenados localmente, isto é, disponíveis no computador onde as análises serão realizadas, basta utilizar a função `readtext()`, indicando o local (relativo ou absoluto) do arquivo desejado.

Aqui mostramos o exemplo do *Livro do Desassossego*. Para tal efeito, a partir da edição de Jacinto do Prado Coelho (1982) disponível no *Arquivo LdoD* (https://ldod.uc.pt/), fizemos um levantamento de todo o corpus textual que compõe o *Livro do Desassossego* e colámos fragmento por fragmento em um bloco de notas com todos os fragmentos da edição de 1982. Uma vez pronto o ficheiro com o corpus, apagamos qualquer tipo de informação para-textual e editorial (como notas dos editores) que pudessem interferir na pesquisa automática do software.

Um corpus pode ser estruturado de forma que todos os textos que o compõem sejam parte de uma mesma unidade, sem ocorrência de delimitação de fronteiras entre um texto e outro. Porém, em muitos casos, há a necessidade de estipular os limites entre as unidades que formam um corpus. Por exemplo, se estivermos a analisar todas as obras de Camilo Castelo Branco, poderíamos criar um único ficheiro contendo todas as obras ou poderíamos criar um ficheiro para cada obra (livro) do autor. A necessidade de estipular os limites entre as unidades que formam um corpus depende de cada projeto. Para investigações em que essas delimitações não sejam necessárias, basta armazenar todo o corpus em apenas um único arquivo simples e importá-lo ao ambiente do R utilizando a função *readtext*.

Embora não seja necessário, é uma boa prática explicitar o tipo de codificação dos caracteres do texto. Esta codificação, ou *encoding*, é um processo que permite a representação dos caracteres (alfabeto, pontuação, símbolos) de forma compreensível para humanos e eficiente para armazenamento e processamento computacional. Existem vários tipos de *encoding*, sendo ASCII e Unicode os mais comuns (Moran & Cysouw, 2018).

```
# ler um corpus salvo em um arquivo único
LdoD <-readtext("pessoaldodmerged.txt", encoding = "LATIN1")
# ler um corpus salvo em arquivos separados
LdoD_multi <-readtext(paste0(path_data, "/ldo/fragmentos/*.txt"))
```

Caso o texto importado ainda possua elementos indesejados, ou ruídos, como cabeçalhos ou números de páginas, é possível identificá-los e removê-los de forma sistemática, utilizando as chamadas expressões regulares, ou regex. No caso específico do LdoD, utilizamos expressões para remover as datas e os títulos (L.doD.) contidos no rodapé.

2.4. Investigações com o Quanteda

Depois que os arquivos estão carregados no sistema, precisamos criar um objeto "corpus", i.e., o formato necessário para que o Quanteda possa processar e gerar informações sobre o(s) texto(s). Para isso, basta aplicar a função corpus. Automaticamente, o texto é segmentado em *tokens* e frases. *Tokens* correspondem a todas as ocorrências (incluindo as repetições) de palavras, e outros itens como pontuação, números e símbolos. Quando investigamos o corpus com a função *summary*, temos a contagem das frases, *tokens* e dos *types* (o número de *tokens* distintos em um corpus).

```
LdoD_corpus <-corpus(LdoD)
summary(LdoD_corpus)
## Corpus consisting of 1 document, showing 1 document:
##
##                  Text Types Tokens Sentences
##  pessoaldodmerged.txt 17648 161682      7453
```

O corpus correspondente ao *Livro do Desassossego* analisado consta de 7453 frases, 161682 *tokens* (palavras e outros itens) e 17648 palavras e pontuação únicas. Caso seja necessário, podemos alterar a estrutura do nosso corpus. No corpus do LdoD acima, temos apenas um texto. Com *corpus_reshape* podemos criar um novo corpus em que cada frase seja considerada um texto.

```
ndoc(LdoD_corpus)
## [1] 1
LdoD_sent <-corpus_reshape(LdoD_corpus, to = "sentences")
ndoc(LdoD_sent)
## [1] 7453
```

Os exemplos acima nos mostram que um corpus é um conjunto de textos com informações sobre cada texto (metadados), do qual pode-se extrair facilmente a contagem de *tokens*, *types* e frases para cada texto. Porém, para realizar análises quantitativas no corpus, precisamos quebrar os textos em *tokens* (*tokenização*). É possível também filtrá-los, removendo elementos como pontuação, símbolos, números, urls e separadores.

```
# todos os tokens
toks <-tokens(LdoD_corpus)
toks_Sent <-tokens(LdoD_sent)

# remover pontuação
toks_nopunct <-tokens(LdoD_corpus, remove_punct = TRUE)

# remover números
toks_nonumbr <-tokens(LdoD_corpus, remove_numbers = TRUE)

# remover separadores (Unicode "Separator" [Z] and "Control" [C]
categories)
toks_nosept <-tokens(LdoD_corpus, remove_separators = TRUE)

# remover vários ao mesmo tempo
toks_simples <-tokens(LdoD_corpus, remove_numbers = TRUE, remove_
symbols = TRUE, remove_punct = TRUE)
```

É possível também remover *tokens* indesejados. Quanteda oferece uma lista de "*stopwords*" para diferentes línguas. *Stopwords*, ou palavras vazias em português, são palavras a serem removidas quando se processa textos para análises computacionais. Não existe uma lista padrão, mas geralmente as *stopwords* são as palavras mais frequentemente utilizadas em uma língua, como preposições e artigos. O bloco abaixo elimina as palavras incluídas na lista *stopwords* para português e inclui outras palavras que se repetem no corpus em questão.

```
toks_nostop <-tokens_select(toks, pattern = stopwords("pt"),
selection = "remove")
toks_nostop_alias <-tokens_remove(toks, pattern = stopwords("pt"))

toks_rm <-tokens_select(toks, pattern = c("é", "l.dod", "porqu",
"ha", "ond", "tudo", "toda", "porque", "onde", "mim", "todo", "tão",
"ter", "grand", "ell", "sobr"), selection = "remove", padding =
TRUE) # padding elima as palavras, mas preserva a contagem
```

A seleção de *tokens* indesejados é já um processo de interpretação sobre o corpus da obra que pode afetar o resultado da análise textual. A seção 3.2 ilustrará essa diferença.

Diferentemente dos ruídos (1.3) previamente eliminados com o auxílio de expressões regulares, a remoção acima é aplicada apenas às listas de *tokens*. O corpus permanece inalterado. Isso é importante pois palavras vazias podem ser irrelevantes ou afetar negativamente o processamento qualitativo do texto. Porém, ao analisar qualitativamente o texto, é importante que elementos como conectivos e pontuação estejam presentes no texto para que o mesmo seja compreensível.

Após a *tokenização*, o próximo passo é criar uma tabela com a frequência de cada *token* por cada texto, ou nos termos do Quanteda, um *document-feature-matrix* (dfm).

A dfm é um pré-requisito para várias outras funções no quanteda, como é o caso da topfeatures, que retorna os *tokens* mais frequentes e um corpus.

```
dfm_pessoa <-dfm(toks_rm)
dfm_stem <-dfm_wordstem(dfm_pessoa)
topfeatures(dfm_pessoa, 20)
##    vida      ser     alma     nada   sempre    nunca    todos
outros      sei    sonho    mundo    assim
## 763      534      412      334      331      314      290
264      261      250      234      220
## sobre    ainda qualquer    outro    outra   grande      dia
coisa
## 215      210      200      198      193      184      184
175
topfeatures(dfm_stem, 20)
## vida   ser outro  alma sonho coisa  nada sempr nunca grand  todo
outra   sei   dia mundo cousa   ell assim
## 795   534   462   462   400   345   336   331   314   291   290
283   264   262   245   243   235   220
## sobr ainda
## 215   210
```

Depois de gerar a lista de *tokens*, podemos então explorar o corpus. Uma das técnicas mais simples e utilizadas para investigação de corpus é através das linhas de concordâncias, ou *concordance lines*, ou *keywords in context* (kwic). As linhas de concordância mostram fragmentos do corpus onde há ocorrência do(s) termo(s) buscados. O número de palavras no contexto, pode ser estipulado pelo usuário, sendo 5 *tokens* a esquerda e 5 a direita o padrão.

Há várias opções para buscas. Elas podem ser feitas por palavras ou por fragmentos, sequências, combinações das mesmas. O código abaixo mostra todas as ocorrências de palavras que iniciam com "feli".

```
# palavras que iniciam com feli
kwic(toks, pattern =  "feli*")
# buscas por mais de um termo
kwic(toks, pattern = c("feli*", "alegr*"))
# buscas por sequência de mais de um token
kwic(toks, pattern = phrase("me fal*"))
```

Desde o início da *tokenização* e da contagem de *tokens* mais frequentes é possível perceber como esta primeira análise apresenta significantes relevantes dentro da obra. Pode surpreender que a palavra "vida" seja a mais utilizada no *Livro do Desassossego*, mas deve ser colocada no contexto também das outras palavras relevantes e das coocorrências do próprio termo. Este tipo de análise, efetuado sobre um grande corpus textual, por exemplo de obras pertencentes a um mesmo género e/ou século, pode oferecer

campos semânticos que ajudem, quer no estabelecimento de taxonomias, quer na interpretação de pensamento entendido como relação entre esses termos.

2.4.1. N-gramas

Listas de frequência de palavras podem ser úteis para identificar elementos comuns a um texto. Porém, em muitos casos, é importante também saber em qual contexto estas palavras estão. Identificar quais palavras coocorrem frequentemente em um corpus podem nos revelar ainda mais informações sobre o texto. Por exemplo, saber que o par "estou triste" ocorre frequentemente no corpus nos diz mais sobre o corpus do que a frequência da palavra "triste" sozinha. A sequência "estou triste" é um exemplo de que chamamos de *n-grams*, ou neste caso específico, bigramas. *N-gramas* são sequências de duas ou mais palavras que ocorrem em um texto. Para gerar listas de *n-grams*, partimos de uma lista de *tokens* e delimitamos o número mínimo e máximo de *tokens* em cada *n-grama*.

```
# criar uma lista de 2-grama, 3-grama e 4-grama
toks_ngram <-tokens_ngrams(toks_simples, n = 2:4)
# visualizar apenas os 30 mais frequentes
head(toks_ngram[[1]], 30)
##  [1] "Na_casa"               "casa_de"               "de_Saude"
##  [4] "Saude_de"              "de_Cascaes"            "Cascaes_
Inclue"
##  [7] "Inclue_Introdução"     "Introdução_entrevista" "entrevista_
com"
## [10] "com_António"           "António_Mora"          "Mora_
Alberto"
## [13] "Alberto_Caeiro"        "Caeiro_Ricardo"        "Ricardo_
Reis"
## [16] "Reis_Prolegómenos"     "Prolegómenos_de"       "de_Antonio"
## [19] "Antonio_Mora"          "Mora_Fragmentos"       "Fragmentos_
Vida"
## [22] "Vida_e"                "e_obras"               "obras_do"
## [25] "do_engenheiro"         "engenheiro_Alvaro"     "Alvaro_de"
## [28] "de_Campos"             "Campos_Livro"          "Livro_do"
```

N-gramas, no entanto, englobam apenas sequências ininterruptas de palavras. Uma outra função, muito comumente utilizada na linguística de corpus, é a identificação de colocações. Colocações acontecem quando há uma "coocorrência de duas (ou mais) palavras em uma frequência maior do que seria de se esperar caso a coocorrência fosse aleatória." (Tagnin, 2004). Diferentemente das *n-gramas* calculadas acima, as palavras que formam uma colocação (colocado e base do colocado) não são necessariamente sequências ininterruptas, sendo a distância entre o colocado e seu base estipulado durante o seu cálculo. A distância máxima (ou horizonte) é frequentemente definida com o valor

cinco. Isso significa que o colocado pode ocorrer cinco palavras à esquerda ou cinco palavras à direita de sua base.

A identificação de colocações em um corpus pode ser feita através de diferentes medidas de associação. As medidas de associação são fórmulas matemáticas que interpretam os dados de frequência de coocorrência. Para cada par de palavras extraídas de um corpus, elas calculam um valor de associação que indica o quão associadas (estatisticamente) essas duas palavras são. Muitas medidas de associação são baseadas em testes estatísticos de hipótese, enquanto outras são combinações puramente heurísticas das frequências (como colocados ou não). A função abaixo retorna uma lista de possíveis colocados para o termo desejado.

```
#
print_collocations("sentir")
Freq-terms Freq   MI-terms       MI  Dice-Terms       Dice    LL-Terms       LL
1       sentir  133     sentir 4.010745       sentir 1.00000000
pensar 76.36341
```

No caso da obra de Fernando Pessoa, estudamos as colocações entre os campos semânticos de "pensar" e "sentir". Foi interessante ver a relação entre os termos, dois significantes já valorados em interpretações clássicas da obra, como a de José Gil (2020). O algoritmo mostra a função de verosimilhança (LL-Terms), segundo a qual a palavra que teria mais probabilidade de sair junto de "sentir" seria "pensar". Ferramentas deste tipo podem ter diversas utilidades: o estudo de coocorrências, o estudo de significantes associados a personagens, etc.

2.4.2. Dicionário

Uma outra forma de extrair informações de um texto é com a criação de "dicionários". A função *dictionary* no Quanteda nos permite agrupar tokens por categorias. Esta categorização pode então ser utilizada para buscas no corpus. Por exemplo, podemos criar as categorias "alegria" e "tristeza" contendo palavras relacionadas a esses sentimentos respetivamente. Com o dicionário criado, podemos identificar a distribuição desses termos em um corpus.

```
Dict <-dictionary(list(alegria = c("alegr*", "allegr*", "feli*",
"content*"),
                       tristeza = c("trist*", "infeli*")))

dict_toks <-tokens_lookup(toks, dictionary = dict)
print(dict_toks)
## Tokens consisting of 1 document.
## pessoaldodmerged.txt :
##  [1] "tristeza" "tristeza" "tristeza" "tristeza" "tristeza" "alegria"
##  [7] "alegria"  "tristeza" "alegria"  "alegria"  "alegria"  "alegria"
```

```
## [ … and 389 more ]
dfm(dict_toks)
## Document-feature matrix of: 1 document, 2 features (0.00% sparse)
and 0 docvars.
##                        features
## docs                    alegria tristeza
##   pessoaldodmerged.txt     240      161
```

Em relação com o ponto anterior, o dicionário pode ajudar na criação de campos semânticos, que permitiriam, por exemplo, facilitar a agrupação de textos do corpus mediante núcleos temáticos.

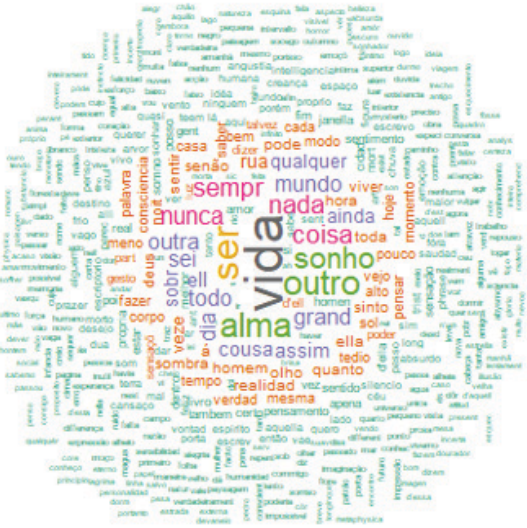
3. Visualização e análise dos dados

3.1. Nuvem de palavras e gráfico de frequência

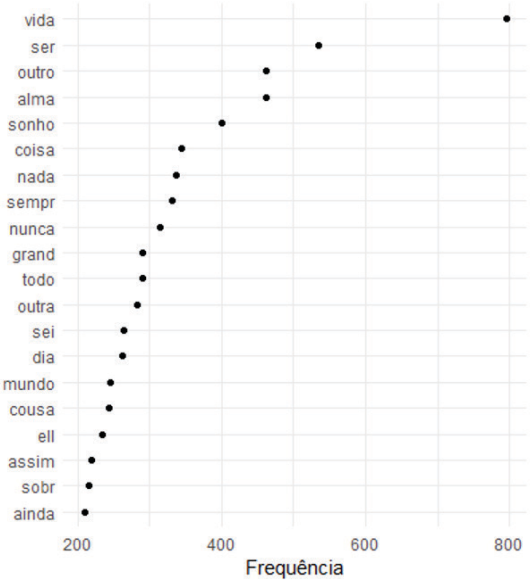
Em 1.4 criamos uma dfm com a frequência dos *tokens*. Para absorver estas frequências de uma forma mais rápida, podemos gerar visualizações com estas frequências. A nuvem de palavras é um gráfico que permite a rápida visualização dos termos mais frequentes.

```
Set.seed(100) #para reprodução dos resultados
textplot_wordcloud(dfm_nostop, min_count = 6, random_order = FALSE,
rotation = .25, color = RcolorBrewer::brewer.pal(8, "Dark2"))
```

Outra solução é utilizar a biblioteca *ggplot* e representar em um gráfico com o número de ocorrências das palavras mais frequentes.



```
Dfm_nostop %>%
  textstat_frequency(n = 20) %>%
  ggplot(aes(x = reorder(feature, frequency), y = frequency)) +
  geom_point() +
  coord_flip() +
  labs(x = NULL, y = "Frequência") +
  theme_minimal()
```



A visualização baseia-se na tradução metafórica entre um conjunto de dados quantitativos e um conjunto de elementos gráficos. Os elementos gráficos estabelecem entre si um sistema de relações cujo objetivo é abstrair e simplificar um sistema de relações quantitativas. Neste caso concreto, as visualizações pretendem mostrar a relação de termos frequentes de forma gráfica.

3.2. Topic modeling (LDA)

Uma outra função frequentemente utilizada na PLN é a modelagem de tópicos, ou *topic modeling* (TM). A modelagem de tópicos aplica um modelo estatístico que procura "entender" a estrutura do corpus e identificar e agrupar palavras que de alguma forma se relacionam entre si. O TM utiliza uma técnica semi ou não supervisionada para identificação desses tópicos. Ou seja, o programa aprende a reconhecer padrões nos dados sem haver a necessidade de anotá-los previamente.

Os códigos abaixo demonstram a aplicação do modelo Latent Dirichlet Allocation (LDA) utilizando dois pacotes distintos: *topicmodel* (Grün, Hornik, 2011) e *seededlda* (Lu et al., 2010). O *seededlda* calculou a modelagem de tópicos tirando sotpwords. Com o topicmodel, além de tirar os *stopwords*, excluímos da análise alguns tokens indesejado:

(“elle”, “ella”, “ha”, “todos”, “?”, “tudo”, “porque”, “mim”, “ter”, “sempre”, “onde”,”l.dod”, “assim”, “sobre”, “todo”, “toda”, “?”, “t?o”, “mesma”,”sen?o”, “todas”, “grande”, “cada”, “ainda”, “qualquer”, “grandes”,”vezes”, “quanto”,”talvez”, “outra”, “outro”, “outros”).

```
# pacote seededlda
> lda_sl <-textmodel_lda(dfm_filtered, k = 8)
> terms(lda_sl, 10)
      topic1       topic2       topic3         topic4  topic5
topic6       topic7      topic8
 [1,] “rapariga”   “trez”       “póde”         “vida”  “symbolo”
“gato”       “ent”       “surg”
 [2,] “proxima”    “viagen”     “egualment”    “ser”   “futil”
“leito”      “conseguir” “deserto”
 [3,] “febr”       “canto”      “distant”      “outro” “desprezo”
“relevo”     “confuso”   “senhora”
 [4,] “ia”         “commum”     “guarda-livro” “alma”  “precisa”
“muita”      “curioso”   “regra”
 [5,] “grandeza”   “cinza”      “soffro”       “sonho” “negação”
“indefinida” “supposto”  “além”
 [6,] “desillusão” “dolorosa”   “soam”         “coisa” “mendigo”
“tinta”      “egua”      “ambo”
 [7,] “haver”      “fome”       “obscuro”      “nada”  “2”
“christo”    “doe-m”     “minuto”
 [8,] “encosta”    “brusco”     “sonhando”     “sempr” “porventura”
“inverno”    “abandono”  “perfum”
 [9,] “braço”      “isolamento” “guarda”       “nunca” “bon”
“continua”   “montanha”  “dess”
[10,] “bocado”     “elemento”   “deu”          “todo”  “acima”
“aliá”       “grand”     “vivendo”


# pacote topicmodels
> lda_tp <-LDA(convert(dfm_filtered, to = “topicmodels”), k = 8)
> get_terms(lda_tp, 10)
      Topic 1 Topic 2 Topic 3 Topic 4 Topic 5    Topic 6 Topic 7 Topic 8
 [1,] “vida”  “vida”  “vida”  “ser”   “alma”     “vida”  “outro” “vida”
 [2,] “alma”  “alma”  “ser”   “outro” “ser”      “alma”  “coisa” “ser”
 [3,] “sonho” “todo”  “sonho” “vida”  “sei”      “sempr” “vida”  “outro”
 [4,] “coisa” “nada”  “nada”  “coisa” “nada”     “assim” “nada”  “outra”
 [5,] “outro” “sei”   “nunca” “sempr” “vida”     “á”     “nunca” “sonho”
 [6,] “outra” “ser”   “sempr” “dia”   “dia”      “todo”  “sonho” “grand”
 [7,] “ser”   “tempo” “grand” “mundo” “qualquer” “vejo”  “ainda” “sobr”
 [8,] “sempr” “outro” “dia”   “grand” “outra”    “rua”   “todo”  “nunca”
 [9,] “cousa” “toda”  “todo”  “alma”  “ainda”    “cousa” “alma”  “noit”
[10,] “grand” “saber” “outra” “cousa” “cousa”    “ser”   “assim” “mundo”
```

Este exemplo é significativo de como a preparação de corpus e o algoritmo de análise podem condicionar o resultado. Como se aprecia no computo anterior, os resultados das análises diferem. No primeiro caso (# pacote seededlda), se tiraram as *stopwords* a partir da lista que Quanteda tem para esses termos em português. No segundo caso (# pacote topicmodels), aos *tokens* indesejados que o programa retira por defeito, se lhe acrescentaram outras palavras, manualmente escolhidas. A primeira análise foi feita sem nenhum tipo de interpretação ou leitura do corpus. A segunda análise foi feita a partir de uma interpretação que considerou irrelevante, para a contagem, uma série de palavras do corpus pessoano. Daí a necessidade de especificar como são modelados os dados.

O TM é útil, como dito anteriormente, para descrever a estrutura de um corpus e as relações entre termos. Pode-se ver uma aplicação prática do TM em uma edição virtual do *Arquivo LdoD* cuja elaboração consulta-se em Manuel Portela (2022), *Literary Simulation and Digital Humanities*, onde o professor da Faculdade de Letras da Universidade de Coimbra descreve como criou uma edição virtual do *Livro do Desassossego* mediante a modelagem de tópicos do Mallet (Machine Learning for LanguagE Toolkit), utilizado para gerar 30 categorias depois de realizar 1500 interações com o corpus (p. 124).

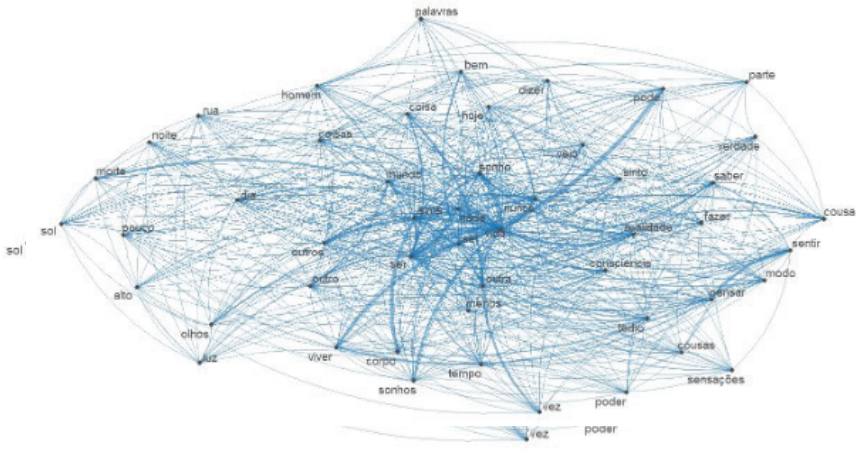
3.3. Semantic Network

O *Feature co-occurrence matrix* (FCM) é similar ao dfm, mas considerando as coocorrências, e apresenta um gráfico com as redes semânticas. Podemos ver o código e o gráfico aqui:

```
fcm_nostop <-fcm(dfm_nostop) #criar fcm a partir de dfm
feat <-names(topfeatures(fcm_nostop, 50)) # listar as top features
fcm_select <-fcm_select(fcm_nostop, pattern = feat, se“ecti”n =
“keep”) #selecionar

size <-log(colSums(dfm_select(dfm_nostop, feat, se“ecti”n = “keep”)))

textplot_network(fcm_select, min_freq = 0.8, vertex_size = size /
max(size) * 3)
```



As redes semânticas em um nível elementar, segundo a teoria de grafos, são entendidas como a ligação entre uma série finita de pontos (*nodes* or *vertex*) mediante uma série finita de líneas (*edges* or *links*). Desde as últimas duas décadas o conceito de rede tem vindo paulatinamente a ocupar um espaço metafórico que pretende dar conta da constelação de relações que há entre os nodes. Citamos anteriormente duas obras que abordam as redes: Galloway & Thacker (2007) e Renn (2020). Na gráfica sobre o *Livro do Desassossego*, vemos a relação entre os termos mais utilizados na obra. Estes tipos de visualizações ajudam a ilustrar tanto o universo temático e estilístico de um autor como a natureza combinatória da linguagem.

4. Considerações finais

O presente artigo mostra as ferramentas de análise com a linguagem R de programação, para computação estatística, que aplica processamentos de linguagem natural a textos. Uma das motivações para escrever o artigo foi explicitar o procedimento e a metodologia para que estas possam ser replicadas em qualquer corpus textual. Ao tentar usar a ferramenta Quanteda no corpus do *Livro do Desassossego*, deparamos com a falta de bibliografia sobre análises de corpus literário que usassem a ferramenta em questão.

Consideramos importante especificar o processo de construção do corpus e de representação de dados nas investigações e publicações a partir de análises textuais com ferramentas digitais e com a manipulação de dados quantitativos. Os resultados variam de acordo com o modelo escolhido. Portanto, especificar o processo de preparação do corpus torna mais transparente a construção quer do corpus, quer da representação, quer da análise, os quais, em última instância, criam significado. Nos tópicos dois e três, apresentamos a ferramenta Quanteda e um método de investigação, que pressupõem termos previamente um corpus para a análise. Podemos resumir a metodologia nos seguintes passos: (a) criar o corpus; (b) *tokenizar* e limpar o corpus mediante o código específico; (c) realizar as análises quantitativas e de representação; (d) ler os resultados, (e) interpretar os resultados; e (f) caso se optar por publicar a investigação, especificar as ferramentas e códigos utilizados na análise.

Deste modo, a metodologia aplicada à análise do *Livro do Desassossego* com o pacote Quanteda, tal e como vimos no artigo, se resume da seguinte forma: (a) a criação do corpus passou pela obtenção da obra com base na edição de Jacinto do Prado Coelho a partir do *Arquivo LdoD* (2017)[8]. O texto foi colocado em um ficheiro .txt para poder efetuar a análise com o programa; (b) se converteu o corpus em tokens e se tiraram as *stopwords* e *tokens* indesejados para a análise; (c) realizaram-se as análises computacionais[9]; (d) leram-se os resultados; (e) interpretaram-se os dados.

---

[8] Acreditamos importante descrever como se cria o corpus. Por exemplo, no caso do *Livro do Desassossego*, optar por uma edição ou outra pode modificar o resultado da análise, dado que as edições variam no número de trechos e na ortografia que usam. Qualquer tipo de especificidade do corpus, deve ser descrita se possível.

[9] Lembramos que o código utilizado na análise e que pode ser replicado no RStudio. O código está acessível para consulta em https://github.com/andressarg/analise_lit_quanteda.

A escolha das *stopwords* (b) no nosso exemplo de *topic modeling* demonstra quão importante é especificar a criação e modelação do corpus. Ter obtido resultados diferentes a partir do mesmo corpus ao utilizar distintas seleções de *tokens* indesejados ilustra o argumento de Andrew Piper em *Enumerations* (2018) quando afirma que "The data sets that I use here are thus not to be confused with some stable macrocosm, a larger whole to which they unproblematically gesture. They too are constructions" (p. 10).

Saber que as palavras mais utilizadas no *Livro do Desassossego*, por exemplo, são "vida", "ser", "sonho", entre outras, ou que a modelagem de tópicos apresenta uma determinada série de conceitos, ou que existe uma relação direta entre os termos de "pensar" e "sentir", comprovada por meio dos algoritmos de análise, não é suficiente. Os processos computacionais e os resultados que oferecem são construções, não verdades absolutas. A análise com R é uma ferramenta que acreditamos ajudar a ler uma grande quantidade de dados, localizar estruturas textuais e representar tais resultados. Mas os dados são meros dados, e é preciso que sejam colocados a par dos marcos teóricos, estéticos, históricos ou filosóficos que os contextualizem.

Referências

Alves, D. (2016). As Humanidades Digitais como uma comunidade de práticas dentro do formalismo académico: dos exemplos internacionais ao caso português. *Ler História [Online]*. 69 | https://doi.org/10.4000/lerhistoria.2496.

Benoit et al. (2018). quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software, 3*(30), 774, https://doi.org/10.21105/joss.00774

Drucker, J. (2020). *Visualization and Interpretation*. The MIT Press.

Galloway A. R., & Thacker, E. (2007). *The exploit*. University of Minnesota Press.

Gil, J. (2020). *Fernando Pessoa ou a Metafísica das Sensações*. N.1. Edições.

Grün, B., & Hornik, K. (2011). Topicmodels: An R Package for Fitting Topic Models. *Journal of Statistical Software*, *40*(13), 1-30. doi:10.18637/jss.v040.i13.

Lu, B., Ott, M., Cardie, C., & Tsou, B. K. (2011). Multi-aspect sentiment analysis with topic models. *2011 IEEE 11th international conference on data mining workshops* (pp. 81-88). IEEE.

Moran, S., & Cysouw, M. (2018). *The Unicode cookbook for linguists: Managing writing systems using orthography profiles*. (Translation and Multilingual Natural Language Processing 10). Language Science Press. DOI: 10.5281/zenodo.1296780

Moretti, F. (2005). *Graphs, Maps, Trees: Abstract Models for Literary History*. Verso Books.

Pessoa, F. (2012). *Livro do Desassossego*. Ed. Zenith, R. Assírio & Alvim.

Piper, A. (2018). *Enumerations*: Data and Literary Study. The University of Chicago Press.

Portela, M., & Silva, A. R. (2017). *Arquivo LdoD: Arquivo Digital Colaborativo do Livro do Desassossego*. Centro de Literatura Portuguesa da Universidade de Coimbra. URL: https://ldod.uc.pt/.

Portela, M. (2022) Literary Simulation and the Digital Humanities: Reading, Editing, Writing. Bloomsbury.

Renn, J. (2020). *The Evolution of Knowledge*: Rethinking Science for the Anthropocene. Princeton University Press.

Santos, D., Alves, D., Amaro, R., Araújo Branco, I., Fialho, O., Freitas, C., Higuchi, S., Langfeldt, M., Marques Lopes, J., Luís dos Santos, A., Pires, E., Ramos, B., Sanches, D., Schumacher Fuão, R., Silva Pereira, P., & Terra, P. (2020). Leitura distante em português: resumo do Primeiro Encontro. MATLIT: Materialidades Da Literatura, *8*(1), 279-298. https://doi.org/10.14195/2182-8830_8-1_16.

Tagnin, S. (2004). "Corpora: o que são e para quê servem". On line: http:// www.fflch.usp.br/dlm/comet/

Underwood, T. (2019). *Distant Horizons*: Digital Evidence and Literary Change. The University of Chicago Press.

(Página deixada propositadamente em branco)