

# TITULO DO TRABALHO

Hugo Veríssimo - 124348 - hugoverissimo@ua.pt

**Abstract** – abstrato em ingles

**Resumo** – abstrato em pt resumo

## I. INTRODUÇÃO

A análise de texto é uma área de estudo fundamental, com diversas aplicações tais como análise de sentimentos ou de opiniões, personalização da experiência do utilizador, recomendação de conteúdo, entre outras [1]. Uma das tarefas centrais nesta área é a identificação da frequência de palavras em grandes volumes de texto, tal como livros, bases de dados ou redes sociais, de modo extrair informações relevantes sobre o conteúdo e estrutura dos textos em análise.

Contudo, a identificação precisa da frequência de palavras em textos de larga escala apresenta desafios significativos, especialmente em termos de memória. Métodos de contagem precisa, que mantêm o registo exato da contagem de cada palavra, revelam-se ineficientes devido ao elevado consumo de memória. Há assim a necessidade do estudo de métodos mais eficientes e escaláveis, principalmente em situações em que os dados estão em constante fluxo, como em *streams* de dados. Neste contexto, algoritmos de contagem aproximada e identificação de itens frequentes têm vindo a ganhar destaque, uma vez que permitem a identificação de palavras mais frequentes de forma eficiente e com uma margem de erro controlada [2].

Este relatório visa explorar três abordagens para este problema: contadores exatos, contadores aproximados e identificação de itens frequentes em *streams* de dados. Para cada uma destas abordagens, será apresentado um algoritmo e ....

## II. METODOLOGIA DA ANÁLISE

Para realizar a análise de frequência de palavras, foram selecionados três livros, a partir livraria online *Project Gutenberg* [3], nomeadamente: *Pinocchio: The Tale of a Puppet* (inglês), *Le avventure di Pinocchio: Storia di un burattino* (italiano) e *Pinocchion seikkailut: Kertomus marioneteista* (finlandês). Estes livros foram selecionados por serem traduções do mesmo livro original, conhecido em português como *As Aventuras de Pinóquio*, de Carlo Collodi. A escolha destes livros permite a comparação da frequência de palavras em diferentes idiomas, bem como a análise de semelhanças e diferenças entre as traduções.

Numa primeira fase, os ficheiros de texto descarregados a partir do *Project Gutenberg* foram processados removendo informações irrelevantes, como metadados e licenças, palavras insignificantes e sinais de pon-

uação. Para além disso todas as palavras foram convertidas para minúsculas e lematizadas. Estas transformações são fundamentais, de modo a simplificar o texto e concentrar a análise nas palavras mais relevantes, garantindo uma avaliação mais precisa e eficiente da frequência de termos. É importante referir que estas transformações foram realizadas com recurso à biblioteca *spaCy*, através do *Python*.

implementar algoritmos, analise dos dados, correr ns quantas vezes, .....

## III. CONTADORES EXATOS

Quanto aos contadores exatos, tal como o nome indica, este tipo de técnica é exate, resultando numa contagem precisa da frequência de palavras, no contexto em causa. O algoritmo apresentado de seguida, designado por *Contador Exato*, é um exemplo de um contador exato, que percorre o texto processado e regista a frequência de cada palavra num dicionário. Este algoritmo é eficiente em termos de precisão, uma vez que mantém um registo exato da contagem de cada palavra, no entanto, revela-se ineficiente em termos de memória, especialmente em situações em que o volume de texto é elevado.

---

### Algoritmo 1 Contador Exato

---

**Entrada:** texto processado (T)

**Saída:** dicionário onde as palavras são as chaves e os valores são as suas frequências (D)

---

```

1: D ← empty dictionary
2: words ← list of words from T
3: for each word in words do
4:     if word ∉ D then
5:         D[word] ← 0
6:     end if
7:     D[word] ← D[word] + 1
8: end for
9: return D

```

---

Atendendo à complexidade espacial, no pior caso, onde todas as palavras que constituem o texto T são distintas, a mesma é dada por  $O(|\text{words}|)$ , onde  $|\text{words}|$  representa o número de palavras no texto. Isto acontece pelo facto do dicionário D conter uma entrada para cada palavra distinta no texto. NAO SEI SE ESTA CERTO, NS SE TENHO DE CONTABILIZAR O TAMANHO DE WORDS AO EM INVEZ DE SER SO O D ou REFERIR QUE O QUE IMPORTA É O TAMANHO DO DICIONARIO? TALVEZ FALAR DOS DOIS e dps no fim dizer q o words deve ser ignorado pq é o texto e o que importa é o dicionario? idk

comparison of the memory (complexity ?) of the algorithms  
.....

IV. CONTAGEM 2

Contadores Aproximados  
lalalla  
DIZER QUE É 1/16 ANTES DO PSEUDOCODIGO

<b>Algoritmo 2</b> Contador Aproximado
<b>Entrada:</b> texto processado (T)
<b>Saída:</b> dicionário onde as palavras são as chaves e os valores são as suas frequências estimadas (D)
1: D ← empty dictionary
2: words ← list of words from T
3: for each word in words do
4:     r ← Uniform(0, 1)
5:     if $r < \frac{1}{16}$ then
6:         if word ∉ D then
7:             D[word] ← 0
8:         end if
9:         D[word] ← D[word] + 1
10:     end if
11: end for
12: for each word in D do ▷ Estimate the total count
13:     D[word] ← D[word] × 16
14: end for
15: return D

V. CONTAGEM 3

lalallala  
<http://dimacs.rutgers.edu/~graham/pubs/papers/freqcacm.pdf>  
fonte do PSEUDOCODIGO

<b>Algoritmo 3</b> Contador <i>Space-Saving</i>
<b>Entrada:</b>
- texto processado (T)
- número máximo de itens a manter (k)
<b>Saída:</b> dicionário com a estimativa das k palavras mais frequentes e respetivas frequências (D)
1: D ← empty dictionary
2: words ← list of words from T
3: for each word in words do
4:     if word ∈ D then
5:         D[word] ← D[word] + 1
6:     else if  D  < k then
7:         D[word] ← 1
8:     else
9:         j ← arg min <sub>j ∈ D</sub> D[j]
10:         D[word] ← D[j] + 1
11:         D ← D \ {j}
12:     end if
13: end for
14: return D

VI. RESULTADOS

TABELA I: CAPTION CAPTION CAPTION

Algoritmo	Complexidade
a	$O(m)$
b	$O(m)$
c	$O(m^2 \times n)$

VII. CONCLUSÃO

conclusaooooo

BIBLIOGRAFIA

[1] Amazon Web Services, “What is text analysis?”, 2024, <https://aws.amazon.com/what-is/text-analysis/>. Accessed: 2024-12-11.

[2] Hongyan Liu, Ying Lu, Jiawei Han, e Jun He, “Error-adaptive and time-aware maintenance of frequency counts over data streams”, 2006.

[3] Project Gutenberg Literary Archive Foundation, “Project gutenberg”, <https://www.gutenberg.org>. Accessed: 2024-12-11.