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* (em inglês, EN), *Le avventure di Pinocchio: Storia di un burattino* (em italiano, IT) e *Pinocchion seikkailut: Kertomus marioneteista* (em finlandês, FI). 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 pontuação. 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 algortimos, 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 $O(|\texttt{words}|)$, onde $|\texttt{words}|$ 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 INVES 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
.....
através da aplicacao do algoritmo de contagem exata, foi possível identificar as 10 palavras mais frequentes em cada um dos livros analisados. A Tabela I apresenta as palavras mais frequentes em cada idioma, juntamente com o número de ocorrências de cada palavra.

TABELA I: CAPTION top10 palavras mais frequentes em cada idioma

EN		IT		FI	
Palavra	#	Palavra	#	Palavra	#
pinocchio	457	pinocchio	460	pinocchio	443
say	282	il	386	sanoa	258
little	238	dire	282	saada	143
puppet	209	si	251	alkaa	134
come	141	burattino	225	tehdä	134
boy	140	volere	167	marionetti	131
like	133	vedere	152	poika	81
good	131	andare	134	huutaa	81
poor	127	povero	134	nähdä	80
go	116	ragazzo	126	kysyä	77

como seria de esperar, a palavra "pinocchio"é a mais frequente em todos os idiomas, uma vez que se trata do nome do protagonista do livro. Para além disso, é possível observar algumas semelhanças entre os idiomas, nomeadamente a presença de palavras que têm o mesmo siginificado em diferentes idiomas, como "puppet"e "burattino"(marioneta em italiano), ...
para além disso, esta analise tambem permite a analise da quantidade de palavras distintas Fig. 2 em cada idioma, bem como a distribuição da frequência de palavras Fig. 1, o que pode ser útil para a comparação de diferentes traduções de um mesmo livro, por exemplo.

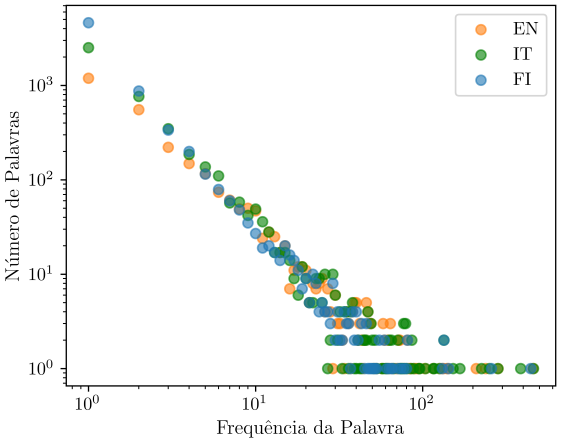

Fig. 1: distribuicao da frequência de palavras em cada idioma

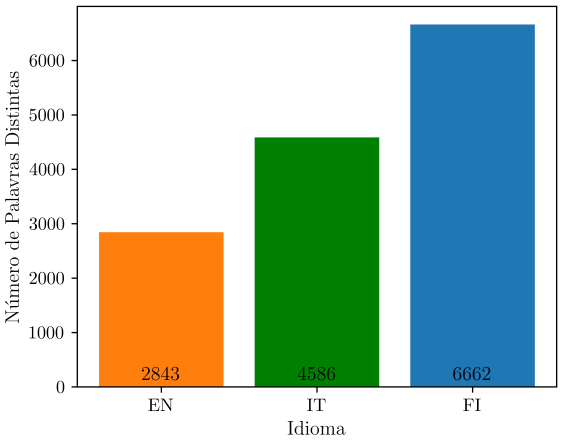

Fig. 2: numero de palavras distintas em cada idioma

a diferenca pode ter a ver com o desempenho do spacy a lematizar as palavras em diferentes idiomas, ou com a propria complexidade da lingua

IV. CONTADORES APROXIMADOS

Os contadores aproximados, também conhecidos por contadores probabilísticos, inventados por Robert Morris, são algoritmos especializados em realizar estimativas eficientes de contagens, utilizando quantidades reduzidas de memória em comparação com métodos tradicionais, através de técnicas baseadas em probabilidade. Estes contadores são particularmente úteis em situações em que a precisão exata da contagem não é crítica, permitindo uma redução significativa no uso de memória, sem comprometer a qualidade da análise [4].

Um exemplo de um contador aproximado é o algoritmo de contagem aproximada apresentado de seguida, designado por *Contador Aproximado*. Neste exemplo, o algoritmo utiliza uma probabilidade de contagem fixa de 1/16. Para cada evento, neste caso, cada palavra do texto, é gerado um número aleatório entre 0 e 1, e a palavra é contada se o número gerado for menor que a probabilidade de contagem. Isto permite que a contagem seja realizada de forma aproximada, com uma margem de erro controlada, e um menor uso de memória.

Algoritmo 2 Contador Aproximado

Entrada: texto processado (`T`)
Saída: dicionário onde as palavras são as chaves e os valores são as suas frequências estimadas (`D`)

```
D ← empty dictionary
words ← list of words from T
for each word in words do
    r ← Uniform(0, 1)
    if r < 1/16 then
        if word ∉ D then
            D[word] ← 0
        end if
        D[word] ← D[word] + 1
    end if
end for
for each word in D do  ▷ Estimate the total count
    D[word] ← D[word] ×16
end for
return D
```

é importante referir q existem outros tipos, diferentes fixed probability, etc.

as contagem sao smp multiplas de 16, pq a probabilidade é 1/16, e dps é multiplicado por 16 para obter a contagem aproximada

complexidade espacial tal tal bla bla

nas tabelas seguintes, por ser um processo aleatorio, apos tere sudo corrido 20 vezes, foi calculada a média, valor max e min de cada palavra, escolhidos as 10 mais frequentes tendo em conta a media, e comparado com a contagem exata. nas tabelas seguintes pode-se observar esses resultados, para cada um dos idiomas analisados. também se repara que as palavras a preto mantiiveram a sua posucao em relacao ao top10 exato, as a laranja estao descoladas mas fazem parte, e as a vermelho não fazem parte do top10 real

TABELA II: top10 dos aproximados sendo o top tendo em conta a media, e media arredondada ao inteiro, e idioma ING

Palavra	$\#_{\text{médio}}$	$\#_{\text{max}}$	$\#_{\text{min}}$	$\#_{\text{real}}$
pinocchio	439	544	336	457
say	270	400	96	282
little	227	320	144	238
puppet	223	336	128	209
come	148	240	64	141
boy	140	176	80	140
go	132	240	48	116
poor	129	240	48	127
like	128	192	32	133
good	126	224	64	131

TABELA III: top10 dos aproximados sendo o top tendo em conta a media, e media arredondada ao inteiro, e idioma italiano

Palavra	$\#_{\text{médio}}$	$\#_{\text{max}}$	$\#_{\text{min}}$	$\#_{\text{real}}$
pinocchio	439	544	240	460
il	379	512	224	386
dire	263	448	160	282
si	252	320	128	251
burattino	223	320	96	225
volere	169	256	96	167
povero	139	240	32	134
bello	133	240	80	116
vedere	131	192	32	152
andare	127	192	80	134

TABELA IV: top10 dos aproximados sendo o top tendo em conta a media, e media arredondada ao inteiro, e idioma finlandes

Palavra	$\#_{\text{médio}}$	$\#_{\text{max}}$	$\#_{\text{min}}$	$\#_{\text{real}}$
pinocchio	445	672	320	443
sanoa	266	352	160	258
saada	160	256	48	143
tehdä	136	224	48	134
marionetti	132	224	80	131
alkaa	104	160	64	134
geppetto	87	160	32	71
huutaa	86	128	32	81
pää	80	128	32	65
olla	76	144	16	61

nota se mais erros quanto menor é a contagem, pq ha maior densidade de palavras, como se viu na contagem exata figura tal

V. Contadores Space-Saving

Pelo facto de muitos processos de geração de dados poderem ser modelados como fluxos de dados, que produzem enormes quantidades de informações simples isoladamente, mas que, em conjunto, formam um todo complexo, torna-se interessante a utilização de métodos que respondam rapidamente a cada nova informação e utilizem recursos muito pequenos em comparação com o volume total de dados. Neste contexto, o algoritmo de contagem *Space-Saving* é uma solução eficiente para a identificação de itens frequentes em *streams* de dados, permitindo acompanhar contagens frequentes de forma eficiente, mesmo sob restrições de memória [5].

Este algoritmo é exposto no pseudocódigo seguinte:

Algoritmo 3 Contador *Space-Saving* [5]

Entrada:
- texto processado (`T`)
- número máximo de itens a manter (`k`)

Saída: dicionário com a estimativa das `k` palavras mais frequentes e respetivas frequências estimadas (`D`)

```
D ← empty dictionary
words ← list of words from T
for each word in words do
    if word ∈ D then
        D[word] ← D[word] + 1
    else if |D| < k then
        D[word] ← 1
    else
        j ← arg min_{j∈D} D[j]
        D[word] ← D[j] + 1
        D ← D \{j}
    end if
end for
return D
```

dizer limitacoes
apresentar resultados
....

VI. Resultados

ou comparacao dos metodos ou assim

VII. Conclusão

conclusaoooo

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.

[4] Robert Morris, "Counting large numbers of events in small registers", *Commun. ACM*, vol. 21, no. 10, pp. 840–842, 1978.

[5] Graham Cormode e Marios Hadjieleftheriou, "Finding the frequent items in streams of data", *Commun. ACM*, vol. 52, no. 10, pp. 97–105, 2009.