Most Frequent Letters – Estratégias de Contagem

Guilherme Antunes

***Resumo* – O problema da contagem de letras num texto é relativamente intuitivo de resolver. Neste relatório são testadas 3 estratégias diferentes para realizar esse processo: um contador exato, um contador aproximado e o algoritmo de Metwally et al. destinado objetivamente à poupança de memória. Estas abordagens foram testadas para diferentes livros, em diferentes idiomas, com diferentes parâmetros de forma a realizar uma análise sobre as circunstâncias em que é mais vantajoso utilizar cada abordagem.**

***Abstract* – The problem of counting letters in a text is relatively intuitive to solve. In this report, three different strategies are tested to perform this process: an exact counter, an approximate counter, and the Metwally et al. algorithm, objectively designed for memory savings. These approaches were tested for different books in different languages with various parameters in order to conduct an analysis on the circumstances in which it is most advantageous to use each approach.**

Keywords – Most Frequent Letters, Fixed Probability Counter, Space-Saving Count, Metwally et al

I. INTRODUÇÃO

A determinação das letras mais frequentes num texto é uma tarefa relativamente simples de executar; em termos práticos consiste apenas em contabilizar as ocorrências de cada letra no texto que estamos a avaliar. Para frases singulares e simples essa contabilização é feita facilmente por um indivíduo sem necessidade de recorrer a nada mais que um papel e caneta, no entanto, para textos maiores e inclusive livros, coleções, bibliotecas inteiras, etc, esta tarefa pode revelar-se bastante cansativa e demorada para o cérebro humano, levando esse cansaço a uma maior preponderância a falhas e erros. Por isso é útil desenvolver algoritmos que sejam capazes de executar esta tarefa rápida e eficazmente.

A abordagem mais óbvia seria percorrer todas as letras e ir somando à medida que cada uma ocorre, como numa contabilização de votos. Apesar desta ser a única maneira de determinar com 100% de certeza a ordem das ocorrências de cada letra pode revelar-se muito dispendiosa em termos de memória pelo conjunto de operações de atribuição e atualização que este processo envolve em termos computacionais.

A fim de tentar resolver esta questão foram desenvolvidas duas abordagens de contabilização para além da que já foi descrita, uma de probabilidade fixa e uma especificamente dirigida à poupança de memória. Estes algoritmos foram desenvolvidos em Python 3.12 e o seu código encontra-se no script run.py que segue em anexo a este documento.

II. PRÉ-PROCESSAMENTO DE TEXTO

Para colocar à prova as 3 abordagens desenvolvidas foi requisitada a utilização de obras literárias em diferentes idiomas como objetos de avaliação, neste caso específico foram utilizadas obras do Projeto Gutenberg.

Foram selecionadas as obras “La venganza de Don Mendo” de Pedro Muñoz Seca, em castelhano, “O Mysterio da Estrada de Cintra” de Eça de Queirós e Ramalho Ortigão, em português, e “The Tragedy of Romeo and Juliet” de William Shakespeare, em inglês. Uma vez que o objetivo é comparar os resultados entre abordagens e não a variação de letras entre idiomas cada obra apenas será utilizada no seu idioma de escrita.

De forma a tentar uniformizar cada texto foram removidos os cabeçalhos e rodapés referentes ao Projeto Gutenberg, foram removidos todos os sinais de pontuação e stopwords, sendo que para cada idioma a lista de stopwords é diferente. Todas as letras foram ainda colocadas em maiúsculas e foram removidos acentos e outras marcas textuais que podem acompanhar uma letra, ficando com uma lista final de 26 letras de A a Z. Para finalizar foram removidos todos os espaços em branco e parágrafos, tornando cada livro numa sequência ininterrupta de letras maiúsculas com apenas 1 linha.

Cada conjunto de stopwords pode ser encontrado na pasta “stopwords” do projeto.

```
def process_line(line: str):

    line = unidecode(line)
    words = line.upper().split()
    words = [word for word in words if word not in stopwords]
    line = " ".join(words)
    line = re.sub(r"[^a-bA-Z]+", ", line)
    return line.strip()
```

III. Contador Exato

Como o nome indica o contador exato contabiliza e guarda num dicionário o número de ocorrências que cada letra faz no texto e é o método mais eficaz para obter a frequência exata de cada letra. O seu simples funcionamento consiste em percorrer cada letra do livro uma a uma e verificar se esta já tem uma entrada no dicionário, em caso afirmativo o valor associado à chave é incrementado em 1, em caso negativo uma nova chave é adicionada com um valor associado de 1. Em termos de implementação, foi utilizado o Counter da biblioteca collections:

```
def exact_counter(file, k):
    return Counter(read_file(file)).most_common(k)
```

A classe *Counter* funciona essencialmente como foi descrito no funcionamento do contador exato, a função *read_file* é uma função desenvolvida para retornar todo o conteúdo do livro pré-processado e a função *most_common* retorna os *k* elementos mais frequentes.

A constante inserção em memória leva-nos a especular que esta abordagem possa ser demasiado dispendiosa para quantidades de texto maiores consumindo assim muita memória virtual. Sendo que esta "especulação" está correta podemos imaginar que não será sempre a melhor solução utilizá-la. Obviamente que com os resultados gerados por este algoritmo poderemos efetuar qualquer cálculo relacionado com a frequência das letras no texto, mas tendo em conta os contras que esta abordagem pode ter vale a pena avaliar para que tipo de cálculos é que será obrigatória a utilização desta estratégia. Por exemplo, precisaremos dos resultados exatos providenciados pelo contador exato para calcular a precisão dos métodos alternativos ou para saber exatamente quantas letras tem o livro ou quantas vezes é que certa letra aparece.

Para mostrar os resultados obtidos pelo algoritmo para os livros pré processados foram calculadas as frequências absolutas de cada letra em cada livro, estando o livro "La venganza de Don Mendo" identificado como "Spanish", o livro "O Mysterio da Estrada de Cintra" como "Portuguese" e o livro "The Tragedy of Romeo and Juliet" como "English"

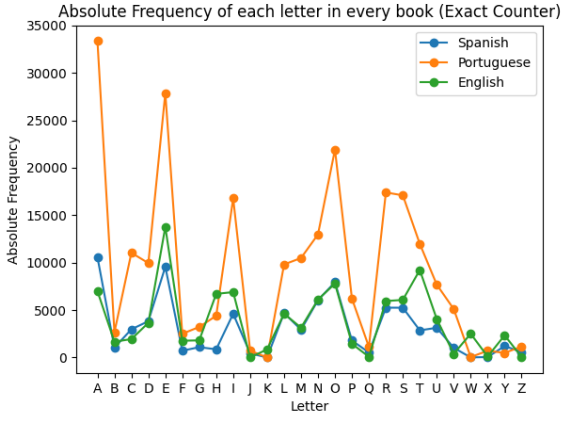

Figura 1 - Frequência absoluta de cada letra em cada livro

Pelo gráfico da figura 1 podemos retirar, por exemplo, que para o livro em português as letras mais usadas são o "A", o "E", o "I" e o "O", vogais, mas que a vogal "U" é muito pouco utilizada quando comparado a outras letras como o "R" e o "S". Também podemos ver que as letras "K" e "W" têm muito poucas utilizações nos idiomas espanhol e português, o que pode ser explicado pela comum associação das letras "K", "W" e "Y" a letras "inglesas". No entanto o "Y" tem alguma utilização no espanhol, explicado pelo pequeno conjunto de palavras castelhanas que utilizam essa letra, e ainda um emprego residual no português, que neste caso pode dever-se à idade que o livro tem estando escrito num português mais arcaico que não excluía o "Y" do seu alfabeto.

Os resultados obtidos pelo contador exato permitem-nos obter o número total de letras em cada livro. Isto é particularmente útil no cálculo da frequência relativa de cada letra, que é uma métrica mais indicada para comparar a utilização de cada letra em idiomas diferentes, livros diferentes, de tamanhos diferentes, utilizando o seu "peso" no livro final e não o seu valor absoluto.

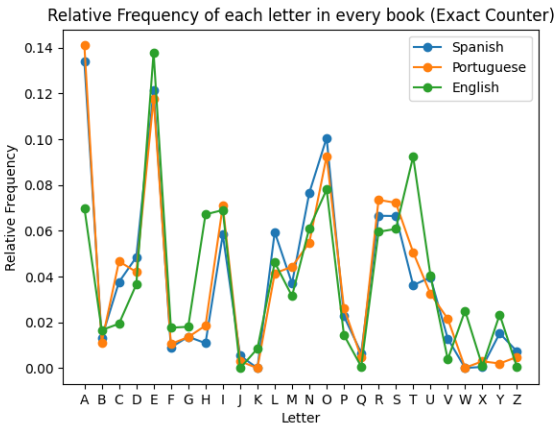

Figura 2 - Frequência relativa de cada letra em cada livro

Observando a figura 2, comparando-a com a 1, podemos ver que a utilização da letra "A" entre o espanhol e o português é bastante idêntica e que entre o espanhol e o inglês afinal é bastante mais diferente do que aquela que aparentava ser na figura 1. Para além do "A" é possível ver diferenças significativas nos pares idioma-letra em letras como o "C", o "H", o "T", o "W" e o "Y". Estas diferenças são novamente justificadas pela forma como cada idioma privilegia a utilização de certas letras em detrimento de outras.

IV. Contador Aproximado

Um contador aproximado sacrifica a exatidão dos resultados em prol da diminuição da memória ocupada. Sendo que para este relatório foi experimentado um contador aproximado de probabilidade fixa de $\frac{1}{16}$, a cada letra do texto será gerado um número aleatório entre 0 e 1, caso esse número seja inferior a $\frac{1}{16}$ então a letra é adicionada à estrutura de dados responsável pela

contabilização da frequência das letras. A utilização deste método para decidir se uma letra conta ou não irá reduzir em muito o número de letras contabilizado, não providenciando, portanto, resultados úteis para estudos que precisem dos valores totais de cada letra. Este contador tem vantagens, como por exemplo, permite determinar a ordem aproximada das frequências das letras com menos recursos, mas também desvantagens, como por exemplo, para letras com frequências muito próximas a ordem poderá não estar correta uma vez que a pseudo aleatoriedade do algoritmo de decisão pode tender a guardar muito mais ocorrências de uma letra do que de outra simplesmente por mero acaso.

```
def approximate_counter(file, k):
    counter = Counter()
    stream = read_file(file)
    for letter in stream:
        if random.random() < PROBABILITY:
            counter[letter] += 1

    return counter.most_common(k)
```

Sendo as funções *read_file* e *most_common* as mesmas utilizadas no contador exato, neste algoritmo as ocorrências das letras são guardadas uma a uma na estrutura *Counter* mediante a probabilidade de $\frac{1}{16}$ de cada ocorrência ser contabilizada, $\frac{1}{16}$ que é o valor definido para a constante *PROBABILITY*.

Como referido a maior vantagem deste algoritmo é a poupança de memória, por exemplo, para uma amostra de tamanho considerável espera-se que apenas cerca de $\frac{1}{16}$ das ocorrências seja armazenado, ao invés da sua totalidade. Isto irá resultar numa diminuição de cerca de 4 bits do número total de ocorrências armazenadas.

Apesar desta vantagem é preciso verificar se aquilo que este contador abdica não interfere na ordem final das ocorrências de cada letra, e para isso os seus resultados foram comparados com os resultados do contador exato.

Spanish		Portuguese		English	
Exact	Approx.	Exact	Approx.	Exact	Approx.
A	A	A	A	E	E
E	E	E	E	T	T
O	O	O	O	O	O
N	N	R	R	A	A
R	S	S	I	I	I
S	R	I	S	H	H
L	L	N	N	N	R
I	D	T	T	S	S
D	I	C	C	R	N
U	U	M	D	L	L

Tabela 1 – 10 letras mais frequentes de cada livro para os contadores exato e aproximado

Os resultados da tabela 1 resultam de uma execução singular do contador exato e do contador aproximado. É possível visualizar que o contador aproximado errou (a sombreado) na posição de 4 letras para o livro em castelhano, 3 letras para o livro em português e 2 letras para o livro em inglês. No entanto, como se trata de um algoritmo com aleatoriedade a análise puramente baseada numa execução não está a ser corretamente avaliada. Por isso o algoritmo foi executado, 1, 10 e 100 vezes para cada livro para fins de comparação e análise de métricas utéis.

Book	La venganza de Don Mendo
Counter	Approximate 1
Precision	0.72
Max Abs Error	9948.0
Min Abs Error	7.0
Avg Abs Error	2962.6
Max Rel Error	1.0
Min Rel Error	0.9047619047619048
Avg Rel Error	0.9407259682931653

Tabela 2 – Métricas do Contador Aproximado do livro "La venganza de Don Mendo" para 1 execução

Book	La venganza de Don Mendo
Counter	Approximate 10
Precision	0.84
Max Abs Error	9897.0
Min Abs Error	7.0
Avg Abs Error	2956.88
Max Rel Error	0.9428571428571428
Min Rel Error	0.9315424610051993
Avg Rel Error	0.9377892741765649

Tabela 3 – Métricas do Contador Aproximado do livro "La venganza de Don Mendo" para 10 execuções

Book	La venganza de Don Mendo
Counter	Approximate 100
Precision	0.92
Max Abs Error	9911.0
Min Abs Error	7.0
Avg Abs Error	2958.24
Max Rel Error	0.9428571428571428
Min Rel Error	0.9357365684575389
Avg Rel Error	0.9376431068008573

Tabela 4 – Métricas do Contador Aproximado do livro "La venganza de Don Mendo" para 100 execuções

Book	O Mysterio da Estrada de Cintra
Counter	Approximate 1
Precision	0.76
Max Abs Error	31283.0
Min Abs Error	44.0
Avg Abs Error	8868.68
Max Rel Error	0.9777777777777777
Min Rel Error	0.9218967921896792
Avg Rel Error	0.9388921239328123

Tabela 5 – Métricas do Contador Aproximado do livro "O Mysterio da Estrada de Cintra" para 1 execução

Book	O Mysterio da Estrada de Cintra
Counter	Approximate 10
Precision	0.92
Max Abs Error	31259.0
Min Abs Error	42.0
Avg Abs Error	8870.76
Max Rel Error	0.9411642411642411
Min Rel Error	0.9333333333333333
Avg Rel Error	0.9371361698068954

Tabela 6 – Métricas do Contador Aproximado do livro "O Mysterio da Estrada de Cintra" para 10 execuções

Book	O Mysterio da Estrada de Cintra
Counter	Approximate 100
Precision	0.92
Max Abs Error	31268.0
Min Abs Error	43.0
Avg Abs Error	8871.96
Max Rel Error	0.9411111111111111
Min Rel Error	0.9360658578856152
Avg Rel Error	0.9375333437173557

Tabela 7 – Métricas do Contador Aproximado do livro "O Mysterio da Estrada de Cintra" para 100 execuções

Book	The Tragedy of Romeo and Juliet
Counter	Approximate 1
Precision	0.6
Max Abs Error	12895.0
Min Abs Error	42.0
Avg Abs Error	3744.64
Max Rel Error	0.9420899854862119
Min Rel Error	0.9021739130434783
Avg Rel Error	0.9357726155739745

Tabela 8 – Métricas do Contador Aproximado do livro "The Tragedy of Romeo and Juliet" para 1 execução

Book	The Tragedy of Romeo and Juliet
Counter	Approximate 10
Precision	1.0
Max Abs Error	12896.0
Min Abs Error	43.0
Avg Abs Error	3743.32
Max Rel Error	0.9445652173913044
Min Rel Error	0.9323529411764706
Avg Rel Error	0.9378273930686231

Tabela 9 – Métricas do Contador Aproximado do livro "The Tragedy of Romeo and Juliet" para 10 execuções

Book	The Tragedy of Romeo and Juliet
Counter	Approximate 100
Precision	1.0
Max Abs Error	12904.0
Min Abs Error	43.0
Avg Abs Error	3743.0
Max Rel Error	0.9404411764705882
Min Rel Error	0.936304347826087
Avg Rel Error	0.9377048963889955

Tabela 10 – Métricas do Contador Aproximado do livro "The Tragedy of Romeo and Juliet" para 100 execuções

As tabelas anteriores permitem-nos observar o aumento da precisão com o aumento do número de vezes que o contador é executado, inclusivamente essa precisão é de 100% para o livro em inglês a partir das 10 execuções. Os valores de erro elevados devem-se à inevitabilidade de a seleção probabilística reduzir o número de ocorrências armazenadas, no entanto é possível observar que o erro relativo médio em cada livro se aproxima de 0.9375, que corresponde às $\frac{15}{16}$ ocorrências que são ignoradas, à medida que a execução do contador é efetuada mais vezes. Outro dado curioso é o erro máximo relativo observado no livro em castelhano, "La venganza de Don Mendo", que para uma única execução regista um valor de 1.0, o que significa que existiu uma letra que efetivamente está no livro, mas que não registou uma única contabilização pelo contador aproximado. Esta situação enaltece a importância de executar o algoritmo várias vezes e de trabalhar com os valores médios de todas elas.

Quanto às ocorrências médias registadas pelo contador aproximado executado 100 vezes para cada letra as mesmas podem ser consultadas na tabela 11 que se segue.

	Spanish		Portuguese		English	
Letter	Exact	Approx.	Exact	Approx.	Exact	Approx.
A	10568	657.9	33350	2082.63	6987	435.76
B	1038	64.14	2651	167.68	1659	103.54
C	2967	185.48	11032	689.08	1937	120.58
D	3823	237.96	9948	623.04	3646	226.81
E	9561	595.86	27876	1747.28	13758	854.4
F	713	45.14	2499	155.42	1765	110.13
G	1083	66.38	3213	199.84	1802	111.68
H	861	54.67	4396	274.27	6703	417.25
I	4602	288.35	16780	1049.49	6890	431.07
J	433	26.81	717	44.29	0	0
K	0	0	0	0	851	52.84
L	4688	294.46	9804	610.14	4606	288.02
M	2923	179.7	10483	655.25	3141	196.3
N	6041	379.02	12975	813.63	6105	382.42
O	7937	497.45	21918	1373.9	7788	486.6
P	1786	111.47	6200	386.7	1442	90.09
Q	522	32.78	1113	69.63	68	4.05
R	5250	327.47	17390	1088.59	5945	371.28
S	5242	326.98	17091	1064.02	6067	375.95
T	2858	177.96	11976	745.84	9220	575.64
U	3126	193.69	7677	481.34	4046	252.46
V	1017	64.17	5104	318.92	383	23.63
W	7	0.4	45	2.65	2502	155.87
X	42	2.68	706	44.92	92	5.86
Y	1201	75.56	481	30.44	2335	145.48
Z	577	37.08	1154	73.78	45	2.85

Tabela 11 – Ocorrências registadas por letra para os contadores exato e aproximado para 100 execuções

V. Algoritmo de Metwally et al.

Para implementar um contador que privilegiasse a poupança de memória foi selecionado o algoritmo de Metwally et al., desenvolvido precisamente para este efeito. O seu funcionamento consiste no estabelecimento de um número máximo de contadores k inicializados a 0, após preencher os primeiros k contadores, quando uma nova letra é tida em conta, caso o contador desta já exista é incrementado em 1 unidade caso contrário o contador com menor número de ocorrências contabilizadas é substituído por um novo contador desta última letra, que assume o número de ocorrências do contador descartado e incrementa-o em 1 unidade. Em termos de implementação foi feita da seguinte forma:

```
def space_saving_counter(file, k):
    counter = Counter()
    stream = read_file(file)
    for letter in stream:
        if letter in counter:
            counter[letter] += 1
        elif len(counter) < k:
            counter[letter] = 1
        else:
            min_letter = min(counter, key=counter.get)
            counter[letter] = counter[min_letter] + 1
            del counter[min_letter]

    return counter.most_common(k)
```

Utilizando funções já referidas anteriormente, é possível perceber que primeiro é verificado se a letra já existe no contador, em caso afirmativo o contador é incrementado, em caso negativo é verificado se o número de contadores já atingiu o máximo k, se esse máximo não tiver sido atingido é criado um contador, se sim é selecionado o contador com o valor mínimo registado é substituído por um contador para a nova letra com o valor do anterior incrementado em 1. O valor retornado pela função utiliza a função *most_common* apenas para colocar o resultado final no mesmo formato das funções anteriores, nenhum contador é eliminado nesta operação uma vez que o algoritmo mantém como número máximo de contadores esse mesmo valor k.

Sendo o objetivo do algoritmo a poupança de memória não só no fim da execução, mas também durante a mesma, esta abordagem apresenta desvantagens, principalmente quando o número de contadores é muito reduzido já que quanto menos diversidade de letras registadas mais fácil é surgir uma letra "nova", o que provoca alterações constantes das letras atribuídas aos contadores, aproximando-os uns dos outros, e desvirtuando as reais letras mais frequentes. Para além disso é provável que o último contador não registe a letra mais frequente naquela posição de ordem, mas sim a última letra avaliada.

Para verificar o funcionamento do algoritmo o mesmo foi executado de maneira a guardar as 3, 5 e 10 letras mais frequentes de cada livro

Book	La venganza de Don Mendo
Counter	Metwally et Al. 3
Precision	0.0
Max Abs Error	18353.0
Min Abs Error	7.0
Avg Abs Error	4064.16
Max Rel Error	2.3123346352526144
Min Rel Error	1.0
Avg Rel Error	1.101989828861259

Tabela 12 – Métricas do algoritmo Metwally et al. do livro "La venganza de Don Mendo" para as 3 letras mais frequentes

Book	La venganza de Don Mendo
Counter	Metwally et Al. 5
Precision	0.0
Max Abs Error	10524.0
Min Abs Error	7.0
Avg Abs Error	3160.88
Max Rel Error	2.0045714285714284
Min Rel Error	0.4926192278576836
Avg Rel Error	1.0298230381874722

Tabela 13 – Métricas do algoritmo Metwally et al. do livro "La venganza de Don Mendo" para as 5 letras mais frequentes

Book	La venganza de Don Mendo
Counter	Metwally et Al. 10
Precision	0.4
Max Abs Error	4114.0
Min Abs Error	1.0
Avg Abs Error	1442.4
Max Rel Error	1.3160588611644273
Min Rel Error	0.0000946252838759
Avg Rel Error	0.7722478062636199

Tabela 14 – Métricas do algoritmo Metwally et al. do livro "La venganza de Don Mendo" para as 10 letras mais frequentes

Book	O Mysterio da Estrada de Cintra
Counter	Metwally et Al. 3
Precision	0.0
Max Abs Error	56948.0
Min Abs Error	45.0
Avg Abs Error	12275.68
Max Rel Error	2.598229765489552
Min Rel Error	1.0
Avg Rel Error	1.111691810299841

Tabela 15 – Métricas do algoritmo Metwally et al. do livro "O Mysterio da Estrada de Cintra" para as 3 letras mais frequentes

Book	O Mysterio da Estrada de Cintra
Counter	Metwally et Al. 5
Precision	0.4
Max Abs Error	30228.0
Min Abs Error	45.0
Avg Abs Error	9517.2
Max Rel Error	1.768650166754432
Min Rel Error	0.418920539730135
Avg Rel Error	1.030607373976884

Tabela 16 – Métricas do algoritmo Metwally et al. do livro "O Mysterio da Estrada de Cintra" para as 5 letras mais frequentes

Book	O Mysterio da Estrada de Cintra
Counter	Metwally et Al. 10
Precision	0.5
Max Abs Error	11432.0
Min Abs Error	2.0
Avg Abs Error	4457.52
Max Rel Error	1.0905275207478775
Min Rel Error	0.0000599700149925
Avg Rel Error	0.7778715154298077

Tabela 17 – Métricas do algoritmo Metwally et al. do livro "O Mysterio da Estrada de Cintra" para as 10 letras mais frequentes

Book	The Tragedy of Romeo and Juliet
Counter	Metwally et Al. 3
Precision	0.0
Max Abs Error	25473.0
Min Abs Error	45.0
Avg Abs Error	5521.4
Max Rel Error	3.2708012326656393
Min Rel Error	1.0
Avg Rel Error	1.1718373151050456

Tabela 18 – Métricas do algoritmo Metwally et al. do livro "The Tragedy of Romeo and Juliet" para as 3 letras mais frequentes

Book	The Tragedy of Romeo and Juliet
Counter	Metwally et Al. 5
Precision	0.2
Max Abs Error	13066.0
Min Abs Error	45.0
Avg Abs Error	4411.24
Max Rel Error	1.8963715529753267
Min Rel Error	0.4505742113679314
Avg Rel Error	1.0772126421276988

Tabela 19 – Métricas do algoritmo Metwally et al. do livro "The Tragedy of Romeo and Juliet" para as 5 letras mais frequentes

Book	The Tragedy of Romeo and Juliet
Counter	Metwally et Al. 10
Precision	0.3
Max Abs Error	4949.0
Min Abs Error	0.0
Avg Abs Error	2057.16
Max Rel Error	1.074468085106383
Min Rel Error	0.0
Avg Rel Error	0.7707331509574812

Tabela 20 – Métricas do algoritmo Metwally et al. do livro "The Tragedy of Romeo and Juliet" para as 10 letras mais frequentes

Pelos resultados das tabelas 12 a 20 podemos perceber que a percentagem de acertos do Metwally et al. é bastante baixa, quando não é mesmo nula, para quantidades de contadores muito pequenas. Das experiências realizadas a que obteve uma melhor precisão, de 50%, foi para o livro em português "O Mysterio da Estrada de Cintra" quando utilizamos 10 contadores. As piores foram as que utilizaram 3 contadores e a do livro castelhano quando utilizou 5 contadores que obtiveram um total de 0 letras na posição certa. Para além disto é possível perceber pelos erros relativos que em todas as vezes há contadores que contabilizaram mais vezes do que aquelas que realmente ocorriam na realidade. Isto deve-se à atualização do contador de valor mínimo para a última letra a ser considerada pois ignora completamente as suas ocorrências anteriores e atribui-lhe um valor arbitrário à sua frequência. Também é interessante o facto de para o livro em inglês "The Tragedy of Romeo and Juliet" o erro mínimo ser 0, o que reflete que o algoritmo acertou na frequência exata de uma letra pelo menos 1 vez.

Os resultados também nos permitem concluir que o aumento do número de contadores disponibilizado em cada execução permite aumentar a precisão do algoritmo, eventualmente tornando os seus resultados mais fiáveis.

Em termos das letras mais frequentes registadas e dos valores de ocorrências registados, esses resultados podem ser encontrados nas tabelas 21, 22 e 23 que se seguem.

Spanish		Portuguese		English	
Exact	Metwally	Exact	Metwally	Exact	Metwally
A: 10568	E: 26290	A: 33350	R: 78868	E: 13758	I: 33262
E: 9561	C: 26290	E: 27876	A: 78867	T: 9220	N: 33261
O: 7937	A: 26290	O: 21918	H: 78866	O: 7788	S: 33261

Tabela 21 – Ocorrências registadas por letra para os contadores exato e Metwally et al. para as 3 letras mais frequentes

Spanish		Portuguese		English	
Exact	Metwally	Exact	Metwally	Exact	Metwally
A: 10568	O: 15774	A: 33350	A: 47321	E: 13758	E: 19957
E: 9561	S: 15774	E: 27876	R: 47321	T: 9220	N: 19957
O: 7937	E: 15774	O: 21918	O: 47320	O: 7788	S: 19957
N: 6041	C: 15774	R: 17390	H: 47320	A: 6987	I: 19957
R: 5250	A: 15774	S: 17091	C: 47319	I: 6890	F: 19956

Tabela 22 – Ocorrências registadas por letra para os contadores exato e Metwally et al. para as 5 letras mais frequentes

Spanish		Portuguese		English	
Exact	Metwally	Exact	Metwally	Exact	Metwally
A: 10568	A: 10569	A: 33350	A: 33352	E: 13758	E: 13758
E: 9561	E: 9563	E: 27876	E: 27879	T: 9220	T: 9579
O: 7937	O: 7947	O: 21918	O: 21961	O: 7788	O: 9558
N: 6041	S: 7347	R: 17390	R: 21919	A: 6987	N: 9557
R: 5250	N: 7242	S: 17091	I: 21915	I: 6890	S: 9556
S: 5242	Z: 7241	I: 16780	N: 21915	H: 6703	I: 9556
L: 4688	C: 7241	N: 12975	U: 21915	N: 6105	R: 9555
I: 4602	R: 7240	T: 11976	D: 21915	S: 6067	M: 9555
D: 3823	M: 7240	C: 11032	C: 21915	R: 5945	U: 9555
U: 3126	U: 7240	M: 10483	H: 21915	L: 4606	F: 9555

Tabela 23 – Ocorrências registadas por letra para os contadores exato e Metwally et al. para as 10 letras mais frequentes

VI. Tempos de Execução

Em termos de tempos de execução para os 10 termos mais frequentes os resultados são os seguintes:

Counter	Execution Time
Exact	0.006715700030326843 s
Approximate 10	0.008617999963462353 s
Metwally et Al. 10	0.05518629983998835 s

Tabela 24 – Tempos de execução para os 3 algoritmos de contagem para as 10 letras mais frequentes no livro “La venganza de Don Mendo”

Counter	Execution Time
Exact	0.019521299982443452 s
Approximate 10	0.022206499939784408 s
Metwally et Al. 10	0.18135449988767505 s

Tabela 25 – Tempos de execução para os 3 algoritmos de contagem para as 10 letras mais frequentes no livro “O Mysterio da Estrada de Cintra”

Counter	Execution Time
Exact	0.007647099904716015 s
Approximate 10	0.010901100002229214 s
Metwally et Al. 10	0.07579810009337962 s

Tabela 26 – Tempos de execução para os 3 algoritmos de contagem para as 10 letras mais frequentes no livro “The Tragedy of Romeo and Juliet”

VI. Conclusão

Das 3 estratégias de contagem de letras o contador exato foi aquele que apresentou melhores resultados, quer em termos de tempo de execução, quer em termos de precisão dos dados obtidos. No entanto os algoritmos que foram comparados serviam objetivos diferentes, enquanto o contador exato fornece dados das frequências absolutas de cada letra num texto, os outros algoritmos tinham como maior propósito reduzir a memória utilizada em diferentes fases. Quer o contador aproximado quer o algoritmo de Metwally et al. propõem-se a ordenar por frequência aproximada as letras do texto a ser analisado utilizando menos memória durante o tempo de execução e no fim desta.

Concluímos assim que para textos relativamente pequenos o contador exato continua a ser a melhor solução, no entanto para quantidades de dados enormes será mais útil utilizar uma das outras duas estratégias já que serão necessárias quantidades enormes de memória virtual para executar com sucesso o contador exato, que as nossas máquinas podem não ter. E mesmo entre as duas abordagens há que conseguir calcular a partir de que número de contadores é que o algoritmo de Metwally et al. começa a produzir resultados fiáveis, ou quantas vezes vale a pena executar o contador aproximado para obter uma ordem correta dos termos contabilizados.

Referências

[1] AA_2324_Trab_3.pdf

[2] AA 09 Probabilistic Counters

[3] AA 11 Data Stream Algorithms I

[4] An Integrated Efficient Solution for Computing Frequent and Top-k Elements in Data Streams