Relatorio Final Cocada

July 18, 2024

Aplicação da Decomposição em Componentes Principais (PCA) em Textos Criptografados

---

Disciplina: Computação Científica e Análise de Dados

Professor: João Paixão

Nome: Pedro Henrique Honorio Saito

DRE: 122149392

---

1 Resumo

O projeto busca visualizar e compreender comparativamente palavras extraídas de diversas obras literária e suas versões criptografadas utilizando técnicas de vetorização e redução de dimensionalidade. As palavras são convertidas em vetores de 300 dimensões usando o modelo de linguagem *FastText* do Facebook, que permite representar tanto palavras conhecidas quanto desconhecidas (OOV). Após a vetorização, as palavras originais e criptografadas são projetadas em um espaço vetorial de três dimensões através da técnica de Análise de Componentes Principais (PCA), que maximiza a variância das projeções para facilitar a análise visual.

O processo inclui pré-processamento do texto, centralização dos dados, e cálculo de autovetores da matriz de covariância para encontrar as principais componentes. A visualização resultante permite identificar padrões e diferenças entre as palavras originais e suas formas criptografadas, sendo uma ferramenta poderosa para análise comparativa nesse subespaço.

Para esta análise, concentrei-me principalmente nos métodos de criptografia de Cifra de César e RSA, pois são amplamente conhhecidos e abordados em outras matérias do curso de Ciência da Computação. No entanto, para diversificar a análise, também incluí um exemplo de cifra multiplicativa.

---

2 Formulação Matemática

Objetivo Desejo visualizar comparativamente um conjunto de palavras extraídas de uma obra literária com suas versões criptografadas. A visualização será feita por meio de um processo denominado vetorização, que consiste na conversão de palavras em vetores de dimensão arbitrária (os vetores estão originalmente em dimensão 300 em função do modelo) com base em características semânticas e fonéticas. Após a vetorização, as palavras originais e suas formas criptografadas serão representadas em um espaço vetorial de 3 dimensões a fim de facilitar as análises.

Para a facilitação das análises, foi escolhido o modelo de linguagem *FastText* do Facebook, notadamente pré-treinado com um vocabulário em português. A escolha pelo modelo se baseia no fato dele permitir a representação de palavras fora do vocabulário, ou também conhecidas por OOV (Out of Vocabulary Words). Entretanto, ainda é necessário passar tais palavras para o modelo para que ele possa atribuir um vetor correspondente. Nesse sentido, irei abordar superficialmente o processo de conversão de palavras em vetores e aprofundar mais na parte de decomposição em componentes principais de forma matemática.

Para transformar palavras em vetores e aplicar técnicas de agrupamento (clustering), seguimos os seguintes passos:

2.1 Representação de Palavras em Vetores

O modelo *FastText* gera vetores tanto para palavras presentes no vocabulário em que foi treinado, quanto para palavras fora desse escopo, mediante as subcomponentes das palavra (subpalavras). Podemos visualizar o procedimento da seguinte forma:

- Temos um conjunto original de palavras $W = \{w_1, w_2, ..., w_n\}$ em que n equivale à quantidade de palavras no texto original.
- Temos um modelo de vetorização das palavras no vocabulário $\phi : W \to \mathbb{R}^d$

Portanto, para cada palavra $w_i \in W$, é atribuído um vetor correspondente $\phi(w_i)$ de dimensão d. No caso do modelo escolhido, estamos trabalhando com $d = 300$.

2.2 Modelo FastText

De antemão, é importante considerar que o texto original deve ser tratado antes que passe pelo proceso de vetorização. Esse tratamento envolve a remoção de acentos e pontuações indesejadas, sendo conhecido por **tokenização**.

Como havia mencionado, o modelo *FastText* leva em consideração tanto as palavras originais quanto suas componentes no processo de vetorização. Com efeito, um vetor de uma palavra w é obtido:

$$\phi(w) = \sum_{g \in G(w)} \mathbf{v}_g$$

De modo que:

- $G(w)$ é o conjunto de subpalavras de w.
- $\mathbf{v}_g$ é o vetor da subcomponente g.

2.3 Decomposição em Componentes Principais (PCA)

Após a conversão tanto do texto original, quanto das palavras fora do vocabulário (OOV) em vetores, começamos o processo de redução de dimensionalidade da nossa matriz de entrada por meio do PCA. Vou detalhar exatamente o que está acontecendo:

Primeiramente, os vetores correspondentes às palavras no texto original são convertidos em uma matriz onde cada linha representa um vetor de dimensão 300 e cada coluna corresponde a uma característica ("**coisas**") específica dos vetores.

Vamos denotar cada vetor convertido $\phi(w_i)$ por $c_i,\ \forall i \in [0, n]$.

Feito isso, ajustando cada vetor em linhas de uma matriz $M_{n\times 300}$ teremos algo como:

$$M = \begin{matrix} \text{"coisa 1"} & & \text{"coisa n"} \\ \begin{bmatrix} c_{1,1} & \cdots & c_{1,300} \\ c_{2,1} & \cdots & c_{2,300} \\ & \vdots & \\ c_{n,1} & \cdots & c_{n,300} \end{bmatrix} & & \end{matrix}$$

Mas, antes de aplicar o PCA, precisamos centralizar nosso conjunto de dados. Essa etapa envolve substrair a média de cada variável dos dados, de modo que cada dado tenha média zero.

$$M_{\text{centralizado}} = M - \text{média}(M)$$

Caso não tivessemos feito essa parte de centralização dos resultados, as componentes principais encontradas teriam chances de serem afetadas pelo deslocamento dos dados

2.4 Interpretação com Função de Três Variáveis e Multiplicadores de Lagrange

Para compreendermos melhor o problema de decomposição em componentes principais, vamos nos concentrar nos seguintes passos:

Em primeiro lugar, o problema de decomposição em componentes principais (PCA) envolve encontrar uma nova base para os dados de modo que a variância projetada (das nossas amostras) seja maximizada. Portanto, sendo M a matriz cujas linhas são os vetores e as colunas indicam suas características, estamos interessados em obter:

$$v_1, v_2, v_3 = \text{argmax}_{\|v_i\|=1} \|Mv\|^2$$

Dado que:

- $\|Mv\|^2$ equivale ao tamanho das projeções sobre as componentes principais (vetores v_1, v_2 e v_3).
- $\|v_i\| = 1$ corresponde à restrição dos vetores serem todos unitários.

Como podemos ver, a primeira expressão $\|Mv\|^2$ pode ser modelada da seguinte forma:

$$\|Mv\|^2 = \left\| \begin{bmatrix} - & c_1 & - \\ - & \vdots & - \\ - & c_n & - \end{bmatrix} \begin{bmatrix} | \\ v \\ | \end{bmatrix} \right\|^2$$

Como desejamos obter o módulo do resultado desse produto interno, podemos simplificar isso tudo para:

$$|c_1^t v|^2 + \cdots + |c_n^t v|^2$$

Com efeito, podemos modelar essa expressão como uma equação de três variáveis, isto é, considerando que nossas componentes principais v_1, v_2 e v_3 estão contidas em $\mathbb{R}^3$. Logo,

$$f(x, y, z) = |c_1^t v|^2 + \cdots + |c_n^t v|^2, \quad \|v_1\| = \|v_2\| = \|v_3\| = 1$$

Dado que desejamos maximizar uma função de três variáveis, sujeita à restrição dos vetores serem unitários, podemos utilizar o conceito de **Multiplicadores de Lagrange**. Segundo esse conceito, para maximizar (ou minimizar) uma função $f(x, y, z)$ sujeita a uma ou mais restrições $g_i(x, y, z) = 0$, devemos encontrar os pontos onde os gradientes da função objetivo e das restrições são lineares. Isto é, estamos interessados nos pontos nos quais os vetores gradientes das funções f e g são paralelos entre si. Portanto, o método estabelece que, para maximizar (ou minimizar) a função $f(x, y, z) = 0$, devemos encontrar os fatores λ tais que:

$$\nabla f(x) = \lambda \nabla g(x)$$

Com base nessa equação, após calcular as derivadas parciais em ambos os lados, obteremos como resultado um sistema com o seguinte formato:

$$\begin{aligned} \nabla f(v) &= \lambda \nabla g(v) \\ &= 2(A^T A)v = \lambda(2v) \\ &= (A^T A)v = \lambda v \\ &= Mv = \lambda v \end{aligned}$$

Nesse sentido, descobrimos que nossa matriz M corresponde à matriz $A^T A$, também conhecida na Estatística como matriz de covariância, para a qual desejamos encontrar o autovetor associado ao maior autovalor λ com o objetivo de maximizar o tamanho das projeções e, consequentemente, aumentar a variância com relação as nossas amostras.

2.5 Encontrando os Vetores Reduzidos

Agora que compreendemos um pouco mais sobre as origens da matriz de covariância, precisamos retornar ao problema de obter os vetores reduzidos de palavras tanto para as originais quanto para as criptografadas. Relembrando que todas as palavras originalmente se encontravam em dimensão 300 após o processo de vetorização, desejamos reduzir sua dimensão para facilitar a análise e visualização dos dados.

Como havia mencionado, encontramos que a matriz de covariância corresponde à multiplicação matricial $M^T M$, o próximo passo é centralizar os resultados obtidos da seguinte forma:

$$C = \frac{M^T M}{n-1}$$

Feito isso, nos resta calcular os autovetores da matriz de covariância, mas antes, vamos interpretar o significado dos autovetores nesse contexto:

Ao calcular a matriz de covariância $A^T A$, estamos, na realidade, obtendo uma matriz de características por características. Se lembrarmos bem, eu havia definido a matriz M como sendo:

$$M_{\text{Palavras} \times \text{Características}}$$

Portanto, a matriz de covariância calculada corresponde à matriz de características por características e, nesse contexto, ao calcularmos seus autovetores, estamos obtendo o quanto cada característica contribui para determinadas "coisas" (componentes principais) que descrevem a maior variância nos dados.

Por esse motivo, estamos interessados na matriz de projeções que pode ser obtida multiplicando M pela matriz de autovetores recém descoberta, ou seja, para projetar os vetores originais nos componentes principais podemos multiplicar:

$$M_{\text{Projetado}} = M \cdot V$$

Alternativamente, podemos calcular a matriz de covariância de M de outra forma, utilizando MM^T, que é uma matriz de vetores por vetores. Os autovetores dessa matriz representam o quanto cada vetor (ou palavra) possui das "coisas" (componentes principais) pelas quais estamos avaliando. Essa última última abordagem é mais eficiente, pois conseguimos obter diretamente o quanto cada palavra possui das componentes principais sem ter que calcular as projeções.

Por fim, encontraremos uma matriz com dimensões $n \times 3$ em que n representa o número de palavras do texto original e 3 indica o posto ou também a quantidade de componentes principais que estamos interessados. Cada linha dessa matriz representa o quanto cada palavra específica possui das três características para as quais estamos avaliando, usamos esses valores para modelar o gráfico em $\mathbb{R}^3$ representativo das palavras.

3 Primeiros Passos: Coleta de Dados

A coleta dos dados foi feita por meio da escolha de textos e poemas da cultura brasileira, com o objetivo de analisar a interpretação dos vetores finais reduzidos das palavras. O primeiro texto escolhido foi um parágrafo da obra brasileira *Capitães de Areia*, descrito abaixo:

A cidade, já escura, tinha muitas pessoas andando nas ruas. E havia a alegria, o som do mar.
A cidade era feita de alegrias e tristezas. Pedro Bala se afastou do cais e entrou nas ruas escuras,
cheias de sombras e de gente que caminhava apressada. Ele conhecia bem aquelas ruas, cada beco,
cada viela. Os Capitães da Areia dominavam aquele lugar.

Chegou à Praça dos Mártires, onde os meninos costumavam se reunir. A praça estava vazia,
mas ele sabia que logo estariam todos ali. Sentou-se num banco e ficou esperando.
Pensou em Dora, a menina que haviam encontrado dias atrás e que agora fazia parte do grupo.
Dora era uma flor no meio daquelas vidas duras. Pedro Bala sentia uma coisa diferente
quando pensava nela, algo que não sabia explicar.

Logo chegaram os outros. Sem-Pernas, com seu andar manco e seu jeito revoltado, Professor,
sempre com um livro na mão, e o Gato, ágil e esperto como poucos. Todos sentaram-se ao redor
de Pedro Bala, esperando as ordens do líder. Ele tinha um plano. Precisavam conseguir comida e roupas.
A noite prometia ser longa e cheia de aventuras.

Pedro Bala explicou o plano. Iria dividir o grupo em dois. Um ficaria de olho nas ruas, para dar
o alarme se aparecesse algum policial. O outro iria até o armazém do Seu Antônio.
Era um velho sovina, mas o armazém tinha tudo que precisavam.
E Pedro Bala sabia como entrar sem ser visto.
Todos ouviram com atenção, e logo estavam prontos para agir.

A noite estava quente. A lua cheia iluminava as ruas, dando um ar quase mágico à cidade.
Pedro Bala sentia-se vivo, mais do que nunca. Com um gesto rápido, deu o sinal,
e os Capitães da Areia se espalharam pela noite, cada um cumprindo sua parte no plano.
A aventura estava apenas começando.

Por fim, a segunda amostra coletada foi um poema famoso chamado *Soneto de Fidelidade*, de Vinicius de Moraes:

> De tudo, ao meu amor serei atento
> Antes, e com tal zelo, e sempre, e tanto
> Que mesmo em face do maior encanto
> Dele se encante mais meu pensamento.
>
> Quero vivê-lo em cada vão momento
> E em louvor hei de espalhar meu canto
> E rir meu riso e derramar meu pranto
> Ao seu pesar ou seu contentamento.
>
> E assim, quando mais tarde me procure
> Quem sabe a morte, angústia de quem vive
> Quem sabe a solidão, fim de quem ama
>
> Eu possa me dizer do amor (que tive):
> Que não seja imortal, posto que é chama
> Mas que seja infinito enquanto dure.

Na seção `Analisand Outros Textos`, também considerei um trecho pequeno da introdução de Memórias Póstumas de Brás Cubas:

> Ao verme que primeiro roeu as frias carnes do meu cadáver, dedico com saudosa lembrança estas memórias póstumas. Não tive filhos, não transmiti a nenhuma criatura o legado da nossa miséria

Para ambas as amostras, irei aplicar um método de encriptação simples, conhecido popularmente por **Cifra de César**. Esse método consiste na substituição de cada letra no nosso texto por outra que aparece um número fixo de posições à frente ou atrás no alfabeto. Para o nosso caso, considerei o deslocamento em 3 unidades para simplificar a análise.

Feito isso, vou analisar como ficaria com um método de criptografia mais robusto, e amplamente utilizado, conhecido pela sigla **RSA** (Rivest-Shamir-Adleman). Este método utiliza um par de chaves, uma pública e outra privada, para encriptar e desencriptar mensagens de forma segura, garantindo a confidencialidade e integridade dos dados.

4 Tratamento dos dados: Tokenização

Logo após a coleta dos dados, é necessário convertê-los em uma forma legível para que o modelo treinado os identifique. Por esse motivo, tratamos nosso conjunto de palavras e removemos acentos, pontuações indesejadas e palavras repetidas, de modo que o conjunto final terá a seguinte cara:

[6]:
```
# Exibindo tokens normais e criptografados
print('Aqui estão os 5 primeiros tokens usando Cifra de César (d=15):\n')

with open('tokens/poema_token.txt', 'r') as amostra, \
     open('tokens/poema_cripto_token.txt', 'r') as amostra_criptografada:
    tokens_texto_original = amostra.read().split()
    tokens_texto_criptografado = amostra_criptografada.read().split()

    max_len = max(len(token) for token in tokens_texto_original[:35])

    print(f'{"Entrada".ljust(max_len)}  |  Criptografado')
    print('-' * (max_len + 20))

    for tn, tc in zip(tokens_texto_original[:5], tokens_texto_criptografado[:
 ↪35]):
        print(f'{tn.ljust(max_len)}  |  {tc}')
```

```
Aqui estão os 5 primeiros tokens usando Cifra de César (d=15):

Entrada         |  Criptografado
---------------------------------
tarde           |  ghoh
ao              |  hvsdokdu
quem            |  srvwr
ama             |  ghuudpdu
enquanto        |  hqtxdqwr
```

5 Treinamento do Modelo para as Palavras Fora do Vocabulário

Como mencionei anteriormente, os modelos pré-treinados não englobam palavras fora do vocabulário. No entanto, para a análise, optei pelo modelo *FastText*, que permite a geração de vetores para palavras desconhecidas por meio da associação das subpalavras que as compõe.

Com efeito, para tratar das palavras criptografadas, basta compilar o modelo e executar a seguinte linha:

[ ]:
```
./fasttext print-word-vectors /caminho/para/modelo < /caminho/para/
 ↪token_criptografado.txt > output.txt
```

6 Resultados e Análises

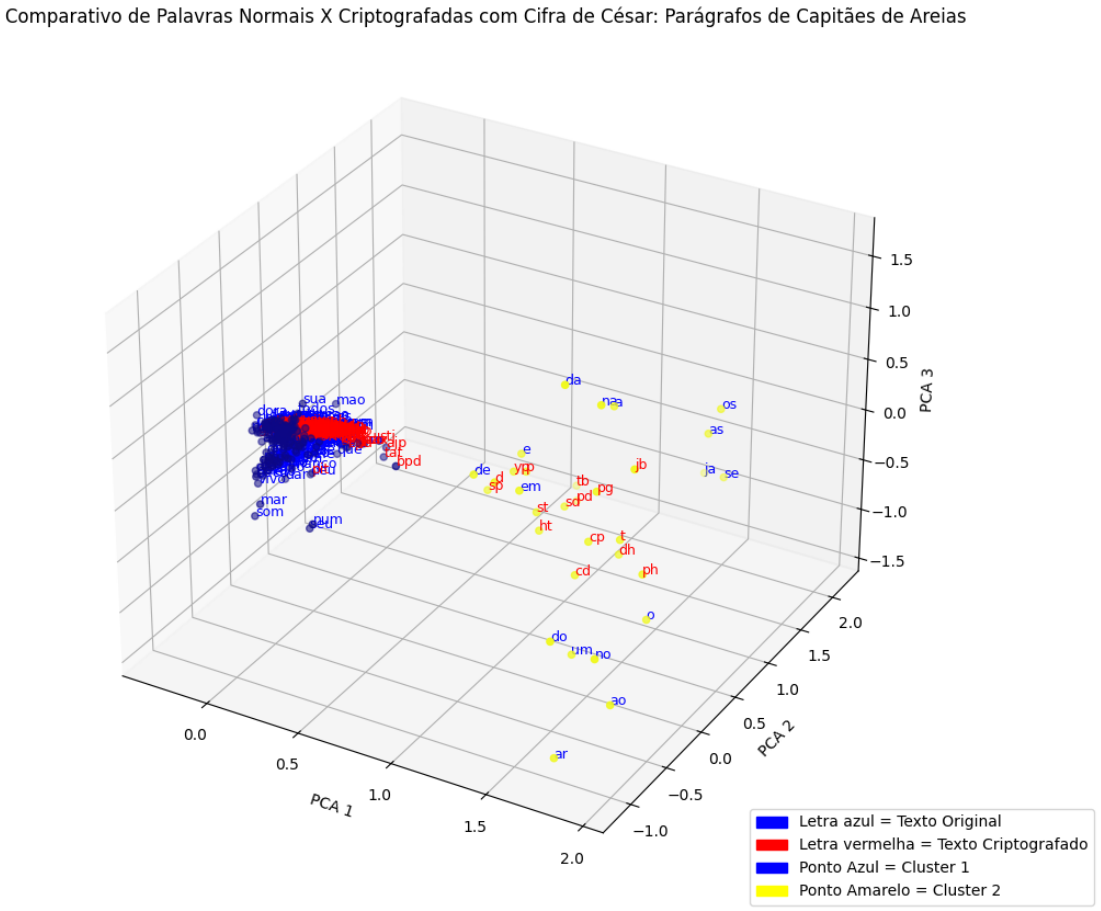
6.1 Visualização dos resultados: Parágrafos Capitães de Areia + Cifra de César com Deslocamento de 15 Unidades

Agora vamos analisar os resultados obtidos para cada amostra, começando com alguns parágrafos da obra Capitães de Areia e a aplicação da **Cifra de César simples** como forma de criptografia:

[4]:
```
import fasttext
from base_funcoes import *
import numpy as np

amostra = 'paragrafos'
metodo = 'cifra_cesar'
deslocamento = 15
titulo = 'Comparativo de Palavras Normais X Criptografadas com Cifra de César:␣
 ↪Parágrafos de Capitães de Areias'

vetores_reduzidos_originais, vetores_reduzidos_criptografados,␣
 ↪tamanhos_projecoes = processamento_plotting(amostra=amostra, metodo=metodo,␣
 ↪deslocamento=deslocamento, titulo=titulo)
```



[5]:
```
import numpy as np
from scipy.spatial.distance import pdist, squareform

# Calculando distância par a par dos vetores
distancias = pdist(vetores_reduzidos_criptografados)

# Determinando a distância média
distancia_media = np.mean(distancias)

print("Distância média entre cada par de pontos criptografados:",␣
 ↪distancia_media)

print(f'Tamanho total das projeções para {amostra} usando {metodo}: ', np.
 ↪sum(tamanhos_projecoes))
```

```
Distância média entre cada par de pontos criptografados: 0.33102722384517136
Tamanho total das projeções para paragrafos usando cifra_cesar:
37.79294420935701
```

6.1.1 Interpretação dos Resultados

Embora o gráfico dos vetores de palavras originais e criptografadas seja um pouco confuso, podemos extrair as seguintes conclusões:

Temos dois grupos significativos de palavras:

Primeiro grupo (Pontos Azuis): Substantivos e verbos contidos no texto original + Substantivos e verbos criptografadas.

- *Alguns exemplos incluem:* Mar, som, vazia, cheia..

Segundo grupo (Pontos Amarelos): Conectivos, pronomes, preposições + Suas versões criptografadas.

- *Alguns exemplos envolvem:* De, do, eu, na, os, as..

A distinção entre ambos os grupos é realizada por meio de similaridades semânticas e pela composição geral das palavras no caso do texto original. Embora, o modelo pré-treinado não seja capaz de inferir o significado das palavras criptografadas, dado que essas não possuem significado intrínseco, ele é capaz de analisar o ordernamento relativo dos caracteres que as compõe. Portanto, a busca por semelhanças é efetuada com base nas subpalavras que as compõem, conforme o funcionamento do modelo FastText.

Semântica e Construção das Palavras e a Proximidade entre Elas. A semântica é especialmente relevante para o primeiro cluster de substantivos e verbos, sobretudo quando se trata das palavras **do vocabulário**. Conseguimos identificar a relação entre grupos de palavras semelhantes, como por exemplo “lugar” e “redor”, dentre outros exemplos que me aprofundarei na próxima seção. Dito isso, a construção das palavras e a ordem dos caracteres são fatores importantes. Isso é evidente

na relação entre “mão” e “não”. Embora essas palavras não compartilhem o mesmo significado, elas estão possuem o mesmo valor para uma de suas componentes principais.

Por fim, ao analisarmos as palavras tanto normais quanto criptografadas, observamos que aquelas com quantidades semelhantes de caracteres tendem a ser posicionadas próximas umas das outras. Isso se torna mais evidente ao analisarmos o segundo grupo, composto por conectivos, pronomes e preposições gerais. Nesses casos, a quantidade de caracteres e a ordem relativa deles nas palavras influenciam significativamente nessa subdivisão entre os dois clusters.

Distância de média dos pontos criptografados entre si de 0.3 **a** 0.4 **aproximadamente.** A distância média registrada para cada par de pontos de palavras criptografadas foi de 0.3 que se trata de um valor mediano se considerarmos a escala em que estamos lidando. Nesse sentido, o tamanho total das projeções dos vetores criptografados sobre o autovetor composto pelas componentes principais foi de 37 que configura um valor alto na escala que estamos trabalhando.

Essa elevada distância média está diretamente relacionada à variância apresentada pelos pontos criptografados. Isso nos informa que, quanto mais dispersos e menos concentrados estão os pontos, maior será o tamanho das projeções desses pontos criptografados sobre o subespaço formado pelas componentes principais.

Como veremos posteriormente, ao usar o método de criptografia RSA a distância entre cada par de pontos criptografados tende a ser muito menor em comparação à essa mesma distância ao utilizar o método de Cifra de César.

Deslocamento escolhido de letras não altera significativamente os vetores. Independentemente do deslocamento d selecionado para a Cifra de César, o resultado da análise foi bem semelhante, isto é, ambos os grupos de palavras tiveram comportamentos análogos para qualquer deslocamento escolhido.

Tamanho Médio das Projeções foi Elevado em função da Variância apresentada pelos Pontos: Como os pontos criptografados apresentaram um comportamento mais disperso e menos concentrado, a projeção desses pontos com relação às componentes principais foi consideravelmente mais elevado. A soma total das projeções dos pontos de palavras criptografadas sobre as componentes principais foi 37. Veremos, em contraste, os mesmos resultados para o RSA.

Conclusão Concluímos que, após a aplicação do mecanismo de criptografia de Cifra de César, independentemente do nível de deslocamento, as palavras criptografadas são deslocadas de forma uniforme, seguindo um padrão claramente distinto entre substantivos, verbos e do outro lado, pronomes, preposições, conjunções. Portanto, é plausível considerarmos uma função reversível e injetora $f(c_i) \to \phi$ em que ϕ representa o vetor da palavra original e c_i corresponderia à palavra criptografada

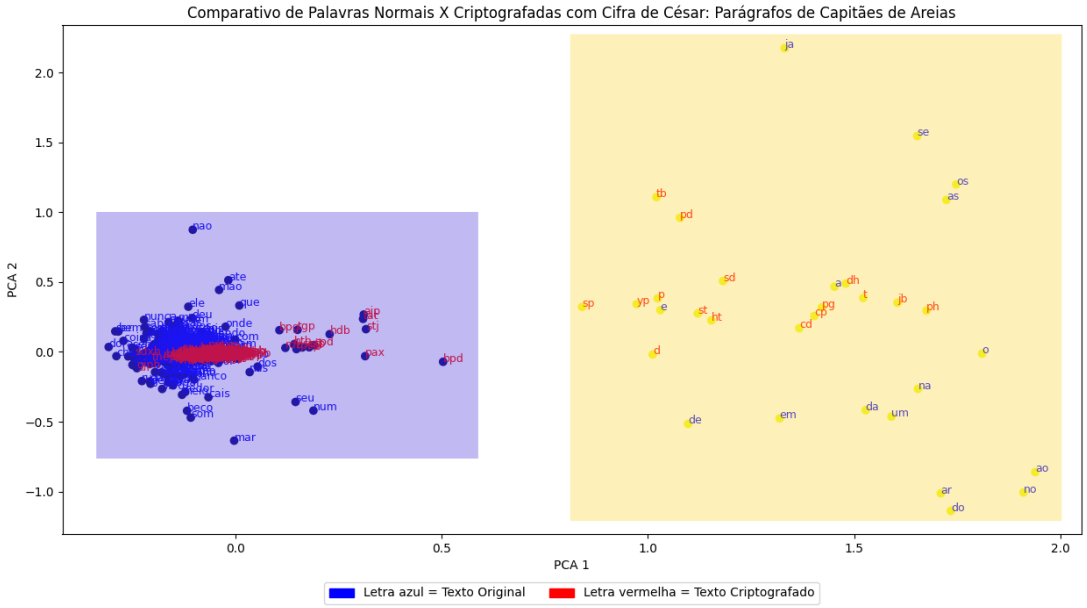
6.1.2 Interpretação em Duas Dimensões

Na figura abaixo conseguimos visualizar com facilidade os padrões na representação em duas dimensões. Detalhei ambos os grupos como havia mencionado acima:

- **Grupo Amarelo**: Representa os pronomes, conjunções e conectivos presentes no texto original e as versões criptografadas dos mesmos. No gráfico abaixo, essa seção é indicada por um cluster mais disperso de palavras.

Por exemplo, tome a preposição "de". Ao aplicar cifra de césar com deslocamento de 15 unidades, teremos "sp". Embora a palavra "sp" não possua um significado semântico e não seja composta pelas mesmas letras que "de", ambas aparecem no mesmo cluster devido à semelhança no número de caracteres e na ordem relativa dos caracteres.

- **Grupo Azul**: Representa os substantivos, verbos e nomes como havia comentado.

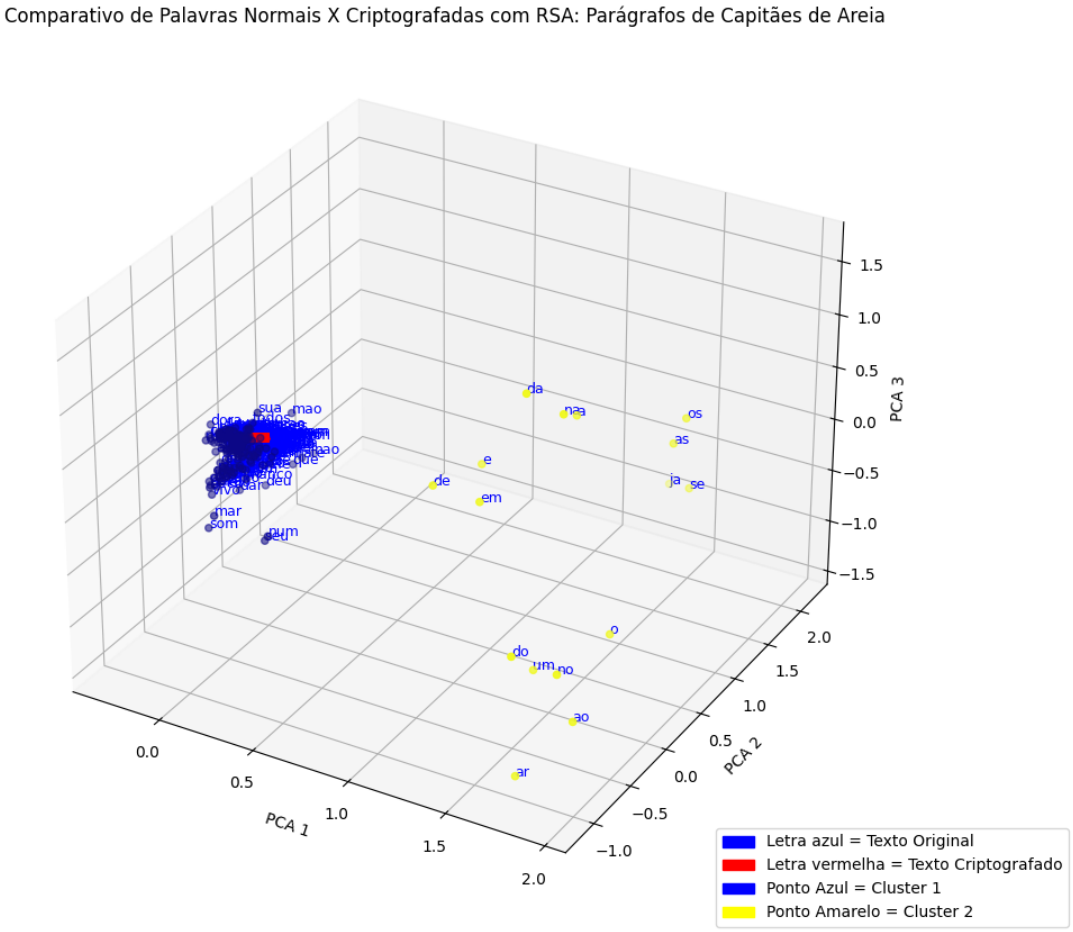


6.1.3 Visualização dos resultados: Parágrafos Capitães de Areia + RSA com Semente 1024

Agora vamos analisar os resultados obtidos para cada amostra, começando com o poema de Vinicius de Moraes e a aplicação da **Cifra de César simples** como forma de criptografia:

[6]:
```
amostra = 'paragrafos'
metodo = 'rsa'
deslocamento = 0
titulo = 'Comparativo de Palavras Normais X Criptografadas com RSA: Parágrafos␣
 ↪de Capitães de Areia'

vetores_reduzidos_originais_rsa, vetores_reduzidos_criptografados_rsa,␣
 ↪tamanhos_projecoes_rsa = processamento_plotting(amostra=amostra,␣
 ↪metodo=metodo, deslocamento=deslocamento, titulo=titulo)
```



[7]:
```
# Calculando distância par a par dos vetores
distancias = pdist(vetores_reduzidos_criptografados_rsa)

# Determinando a distância média
distancia_media = np.mean(distancias)

print("Distância média entre cada par de pontos criptografados:",␣
 ↪distancia_media)

print(f'Tamanho total das projeções dos vetores criptografados no plano␣
 ↪principal para {amostra} usando {metodo}: ', np.sum(tamanhos_projecoes_rsa))
```

```
Distância média entre cada par de pontos criptografados: 0.0050357799937115685
Tamanho total das projeções dos vetores criptografados no plano principal para
paragrafos usando rsa:  5.2160972236268455
```

6.1.4 Interpretação dos Resultados

Embora o gráfico dos vetores de palavras originais e criptografadas continue um pouco confuso, conseguimos claramente perceber resultados distintos:

Obs. Para facilitar a análise, nomeei os pontos criptografados com RSA de c_1 até c_n, tendo em vista que a palavra ao ser criptografada com RSA acaba sendo desproporcionalmente longa em comparação

Temos apenas os dois grupos de palavras normais e das criptografadas se tornou um só:

Primeiro grupo (Pontos Azuis): Substantivos, verbos e nomes contidos no texto original + Todas as palavras criptografadas.

- *Alguns exemplos incluem:* Lugar, redor, vazia, cheia..
- *Exemplo apenas do começo de uma palavra encriptografada com RSA*: `2da10f6c9dd4abd7ae9840c16783cc346c45f1d68c55caa19ed9..`

Segundo grupo (Pontos Amarelos): Conectivos, pronomes, preposições + Nenhuma contraparte criptografada.

- *Alguns exemplos envolvem:* De, do, eu, na, os, as..

Assim como na Cifra de César, os mesmos grupos de palavras prevalecem nessa análise, com a exceção das palavras criptografadas. Os pontos de palavras criptografadas formam um núcleo em vermelho no grupo dos substantivos, verbos e nomes. Nesse sentido, o modelo classifica erroneamente todas as palavras, incluindo conectivos e pronomes criptografados no primeiro cluster dos pontos azuis.

Semântica e Construção das Palavras e a Proximidade entre Elas. Novamente, as palavras do texto original estão distribuídas da mesma forma que anteriormente. Devemos nos atentar à mudança na posição dos pontos criptografados. Como podemos observar, as palavras

criptografadas foram classificadas de forma quase indistinguível entre si. Isso se deve à estrutura peculiar da criptografia RSA, que considera valores numéricos no contexto, afetando ainda mais a semântica e a construção das palavras originais. Essa característica desse método de criptografia faz com que as palavras sejam representadas de forma bastante semelhante entre elas, porém altamente distorcidas em comparação ao texto original.

Distância de média dos pontos criptografados entre si de 0.005 **a** 0.007 **aproximadamente.** A distância média entre cada par de pontos criptografados foi de 0.006 que é significativamente inferior à distância registrada para a Cifra de César. Isso nos informa a respeito não só da proximidade desses vetores, mas também da soma do tamanho das projeções desses pontos. Nessa última análise, a soma dessas projeções sobre as componentes principais foi de 5 aproximadamente, o que reforça a ideia de similaridade entre esses tokens.

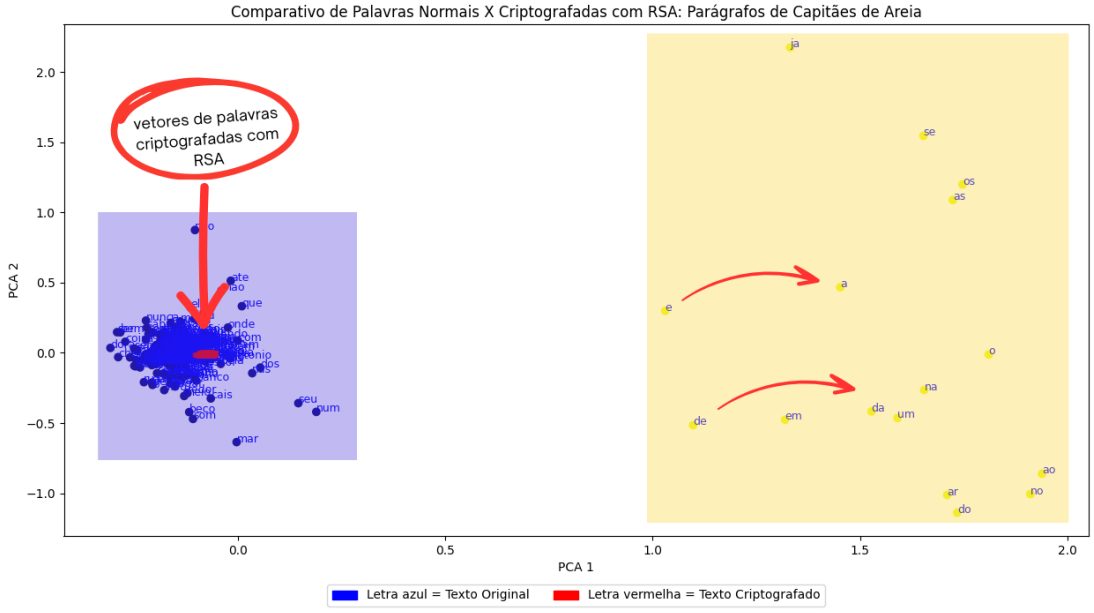
Chave selecionada não altera significativamente os vetores. A chave selecionada para a criptografia RSA não altera significativamente o resultado, desde que seja uma chave válida, ou seja, um número primo suficientemente grande. A semente mencionada no título é utilizada para gerar a chave desejada quando estamos tratando do código do RSA. Testei para algumas sementes distintas e as diferenças entre os resultados foram imperceptíveis entre si.

Conclusão Concluímos que, após a aplicação do mecanismo de criptografia de RSA, as palavras criptografadas se concentram em uma determinada região do espaço, não apresentando um comportamento disperso como observaod anteriormente. Nesse sentido, é difícil imaginar uma função que possa reverter esses vetores com facilidade. Podemos fazer uma analogia entre o comportamento da criptografia RSA e o de uma função agregadora ou de redução. Embora possamos aplicar a função de criptografia de forma direta, a inversão desse processo é complexa e computacionalmente difícil, refletindo a robustez desse sistema de segurança.

6.1.5 Interpretação em Duas Dimensões

Podemos identificar padrões com maior precisão ao visualizar o gráfico em duas dimensões da seguinte maneira:

- Conseguimos identificar os mesmos resultados já descritos para a análise com Cifra de César, porém, como podemos perceber, os pontos (ou vetores) que representam as palavras criptografadas estão **aglomerados na vizinhança de um ponto**. Isto é, todas as palavras criptografadas estão convergindo para um ponto o qual possui "coisas" (componentes principais) com valores semelhantes.
- Adicionei algumas setas a mais para destacar a relação entre as palavras presentes no vocabulário, como, por exemplo, "e" e "a", bem como "de" e "da". Algumas palavras na parte de substantivos possuem uma relação semelhante, por mais que não conseguiremos identificar os pontos em detalhe. Exemplos para esse último caso seriam: os nomes próprios "Pedro" e "Antonio" estarem próximos e os substantivos abstratos "lugar e redor" também.



Aqui estão alguns exemplos adicionais de palavras relacionadas, como "meninas" e "meninos", juntamente de "cidade" e "viela".

7 Analisando outros Textos

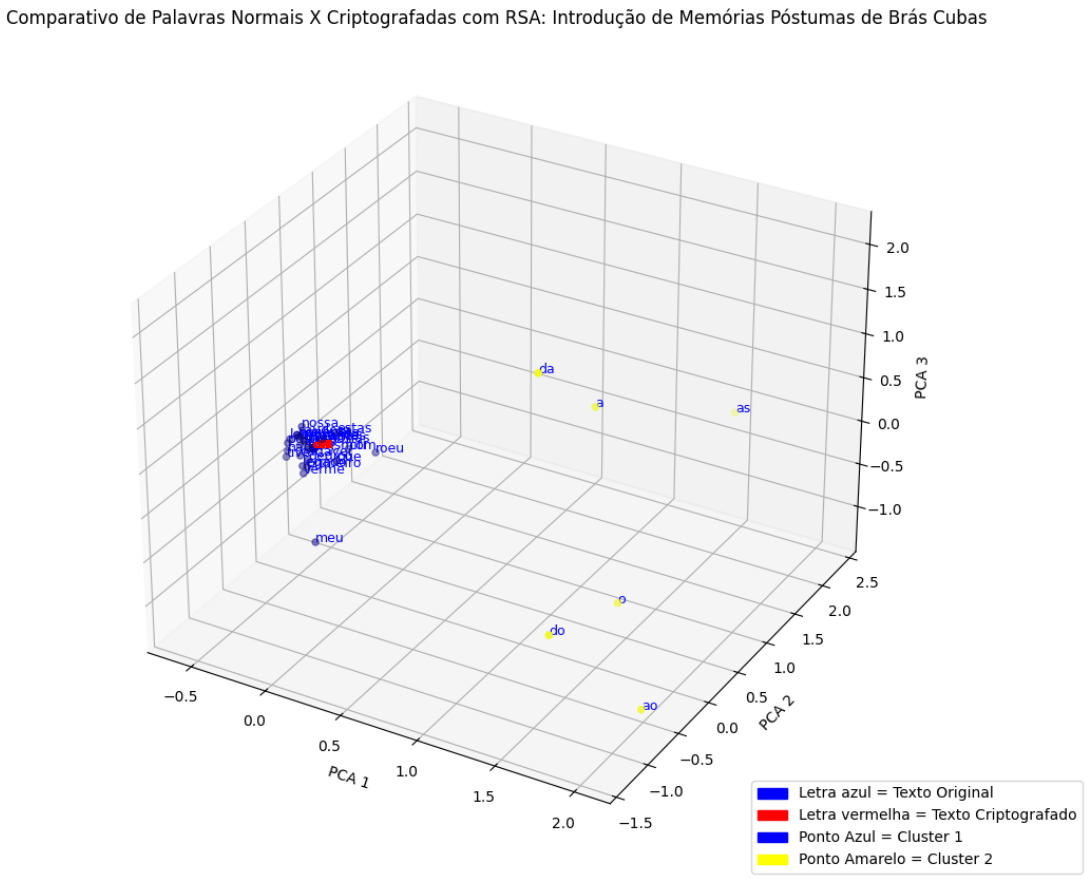
Após analisarmos esse trecho de Capitães de Areia, vamos verificar os resultados obtidos com outros textos

[8]:
```
amostra = 'memorias'
metodo = 'rsa'
deslocamento = 0
titulo = 'Comparativo de Palavras Normais X Criptografadas com RSA: '

vetores_reduzidos_originais, vetores_reduzidos_criptografados,␣
 ↪tamanhos_projecoes_memorias = processamento_plotting(amostra=amostra,␣
 ↪metodo=metodo, deslocamento=deslocamento, titulo=titulo)

print(f'Tamanho total das projeções dos vetores criptografados no plano␣
 ↪principal para {amostra} usando {metodo}: ', np.
 ↪sum(tamanhos_projecoes_memorias))
```

```
Tamanho total das projeções dos vetores criptografados no plano principal para
memorias usando rsa:  0.742421354339321
```

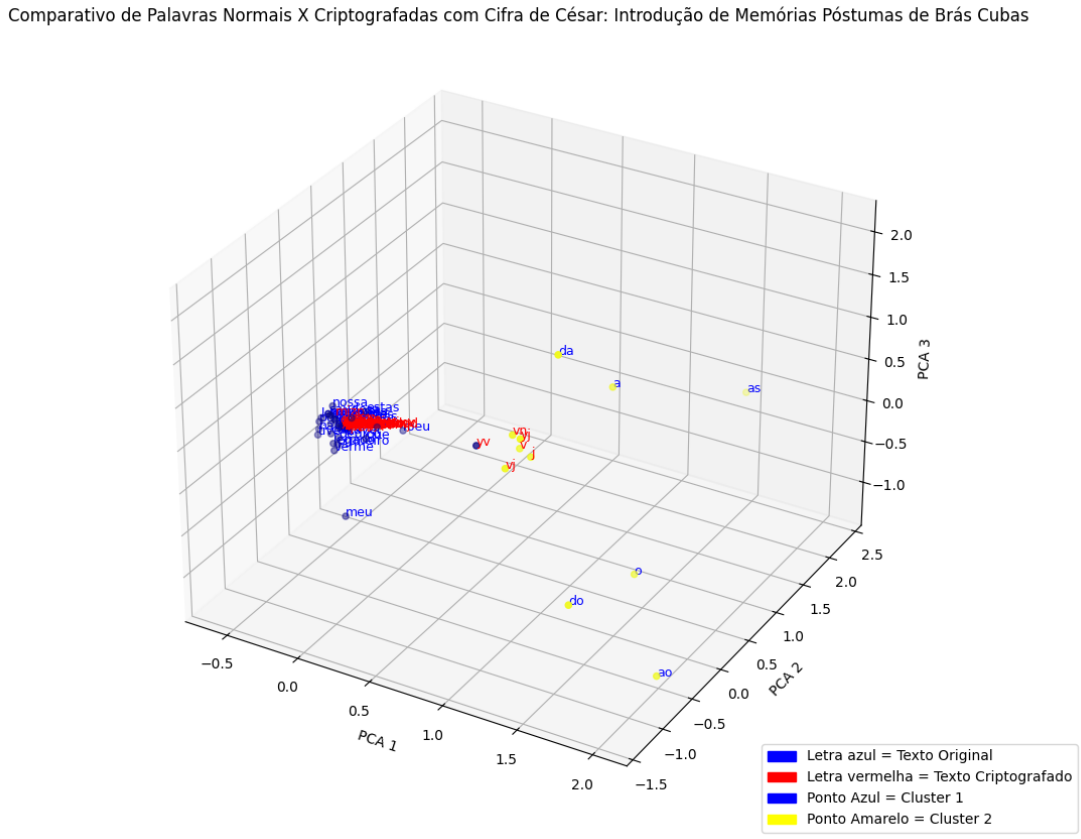


[9]:
```
amostra = 'memorias'
metodo = 'cifra_cesar'
deslocamento = 21
titulo = 'Comparativo de Palavras Normais X Criptografadas com Cifra de César:␣
 ↪Introdução de Memórias Póstumas de Brás Cubas'

vetores_reduzidos_originais, vetores_reduzidos_criptografados,␣
 ↪tamanhos_projecoes_memorias = processamento_plotting(amostra=amostra,␣
 ↪metodo=metodo, deslocamento=deslocamento, titulo=titulo)

print(f'Tamanho total das projeções dos vetores criptografados no plano␣
 ↪principal para {amostra} usando {metodo}: ', np.
 ↪sum(tamanhos_projecoes_memorias))
```

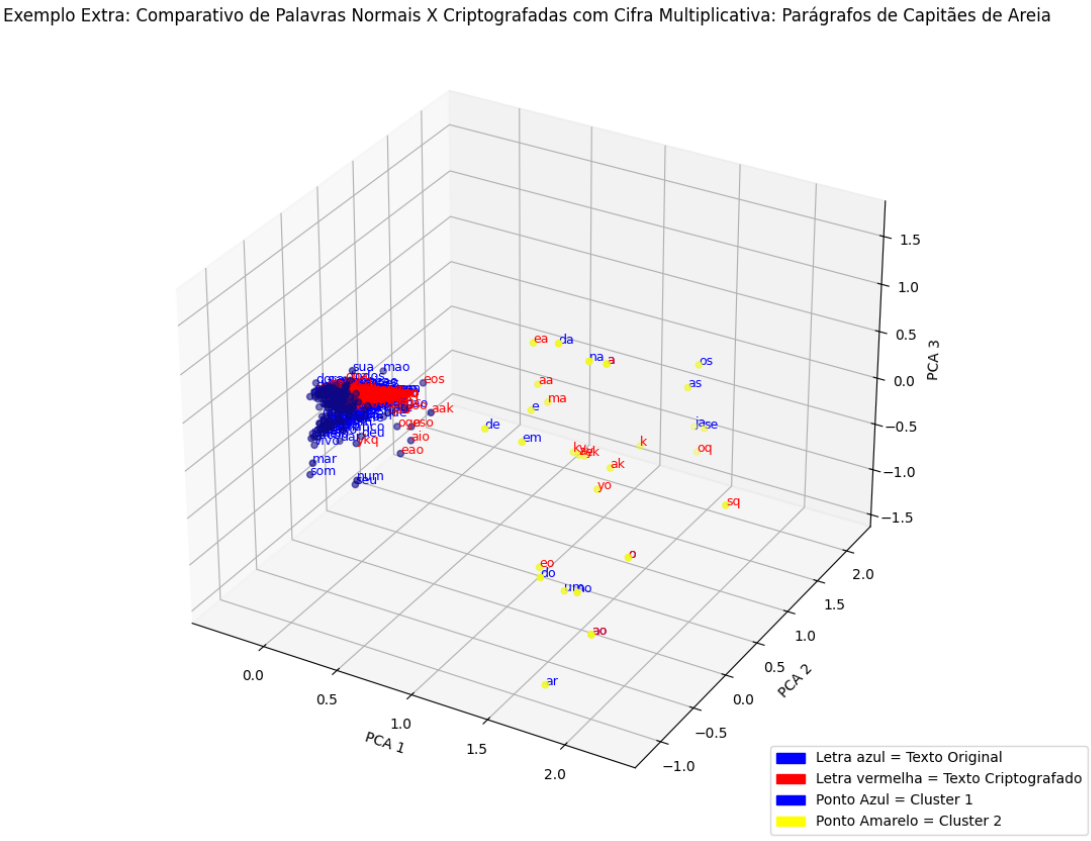
```
Tamanho total das projeções dos vetores criptografados no plano principal para
memorias usando cifra_cesar:  8.106804013808452
```



[2]:
```
amostra = 'paragrafos'
metodo = 'cifra_multiplicativa'
deslocamento = 10
```

```
titulo = 'Exemplo Extra: Comparativo de Palavras Normais X Criptografadas com␣
 ↪Cifra Multiplicativa: Parágrafos de Capitães de Areia'

vetores_reduzidos_originais, vetores_reduzidos_criptografados,␣
 ↪tamanhos_projecoes = processamento_plotting(amostra=amostra, metodo=metodo,␣
 ↪deslocamento=deslocamento, titulo=titulo)
```



[5]:
```
# Calculando distância par a par dos vetores
distancias = pdist(vetores_reduzidos_criptografados)

# Determinando a distância média
distancia_media = np.mean(distancias)

print("Distância média entre cada par de pontos criptografados:",␣
 ↪distancia_media)

print(f'Tamanho total das projeções para {amostra} usando {metodo}: ', np.
 ↪sum(tamanhos_projecoes))
```

```
Distância média entre cada par de pontos criptografados: 0.37411441666838113
Tamanho total das projeções para paragrafos usando cifra_multiplicativa:
```

```
40.71537208821029
```

7.1 Conclusões para as outras Amostras

Para a última análise, apenas considerei alguns textos e métodos de criptografia que vou descrever abaixo:

- **Introdução de Memórias Póstumas de Brás Cubas com Cifra de César** ($d = 21$) **e RSA** (**Semente** = 1024)

Os resultados observados foram bem semelhantes aos anteriores para ambos os métodos de criptografia. Ademais, a soma total das projeções seguiu o mesmo padrão observado antes:

Soma das Projeções	
Cifra de César	8
RSA	0.7

- **Parágrafos de Capitães de Areia com Cifra Multiplicativa**

O mesmo parágrafo de Capitães de Areia quando aplicado a uma cifra multiplicativa apresentou resultados semelhantes ao da Cifra de César. No entanto, os pontos criptografados se distribuiram de forma ainda mais uniforme o que contribui à soma de 40 para o tamanho das projeções em comparação ao 37 da Cifra César. Por se tratar de uma criptografia um pouco mais robusta que a Cifra de César, era esperado valores menores, porém para essa amostra observamos algo um pouco diferente.

8 Conclusão Geral

Ao compararmos os vetores resultantes dos dois tipos de criptografia, observamos o seguinte:

1. O método de Cifra de César apresenta um comportamento mais previsível, separando claramente as palavras originais e criptografadas em seus respectivos grupos. A distância média par a par dos pontos criptografados variou entre 0,4 a 0,7.
2. Por outro lado, o método RSA praticamente agrupa todas as palavras originais em um mesmo conjunto na análise de componentes principais. Os vetores de palavras criptografadas pelo RSA apresentam valores de componentes principais bastante parecidos entre si. Esse comportamento assemelha-se ao de uma matriz que reduz a dimensionalidade dos vetores no seu domínio.

A principal função da análise das componentes principais (PCA) nos textos criptografados foi fornecer uma **"medida de distorção"** com relação ao texto original, de modo que possamos identificar com mais precisão o que está acontecendo por trás da Cifra de César e do RSA, bem como suas diferenças.

Essas observações corroboram a ideia de que a função de reversão $f(c_i) \to \phi$ é menos plausível de ser obtida para o caso da criptografia usando RSA, embora não seja impossível. Chegamos a essa conclusão com base na análise da distância média no subespaço $\mathbb{R}^3$, bem como o tamanho das projeções dos vetores de palavras criptografadas sobre as componentes principais. Portanto, isso nos dá mais indícios de que o método de criptografia RSA é consideravalmente mais robusto e eficaz que a Cifra de César.

9 Referências

FACEBOOK AI RESEARCH. FastText Documentation. 2021. Disponível em: https://fasttext.cc/docs/en/support.html. Acesso em: 18 jul. 2024.

BHASKAR, Uday. Simple Word2Vec Using SVD. 2018. Disponível em: https://udibhaskar.github.io/ml_blog/Simple-Word2Vec-Using-SVD/. Acesso em: 18 jul. 2024.

AMADO, Jorge. Capitães da Areia. São Paulo: Companhia das Letras, 2008.

MORAES, Vinicius de. Soneto de Fidelidade. In: Poesia Completa e Prosa. Rio de Janeiro: Nova Aguilar, 2004.

ASSIS, Machado de. Memórias Póstumas de Brás Cubas. São Paulo: Penguin Classics Companhia das Letras, 2011.