Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Informática

Ciência da Computação
Processos Estocásticos

Trabalho Avaliativo

Professor: Omar Cleo Neves Pereira

129037	Murilo Luis Calvo Neves

Data: 20/05/2025

Conteúdo

1 Introdução

No contexto de linguagens, a análise estatística de letras e palavras pode ser muito interessante por diversos motivos, como quebrar cifras, decifrar línguas, analisar origens e famílias etimológicas, etc. Neste trabalho, será utilizado um olhar estatísticos aplicado juntamente com Cadeias de Markov e outras técnicas para se realizar a implementação e análise de dois artefatos: um gerador de palavras fictícias e um gerador de frases.

2 Preparação dos dados

A fim de ser obter um corpus de texto simples e limpo, foi-se utilizado como *dataset* um conjunto textual constituído da Constituição Brasileira, do Código Penal, Código Civil e Código de Defesa do Consumidor.

O texto de cada uma das fontes foi limpo e concatenado para um único arquivo textual, cujo tamanho em caracteres é de aprox. 1.3 milhões. No entanto, devido à limitações de processamento da biblioteca Spacy utilizada, foi-se limitado em 1 milhão de caracteres. Após isso, foi-se realizada uma análise preliminar a fim de buscar um possível limite superior no tamanho das palavras e frases.

Para o tamanho das palavras, o resultado que se observou foi:

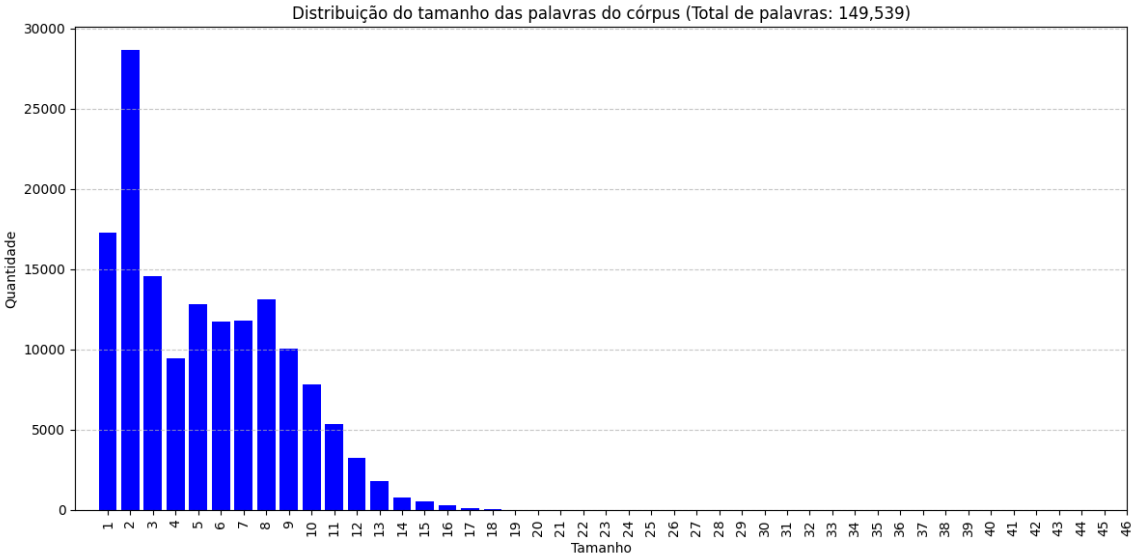


Figura 1: Distribuição do tamanho das palavras no córpus utilizado

Em experimentações com possíveis *datasets* diferentes, observa-se que a distribuição tem caráter similar a uma análise anterior exploratória feita com um *dataset* maior composto de palavras extraídas de artigos da *Wikipedia*, cuja distribuição é:

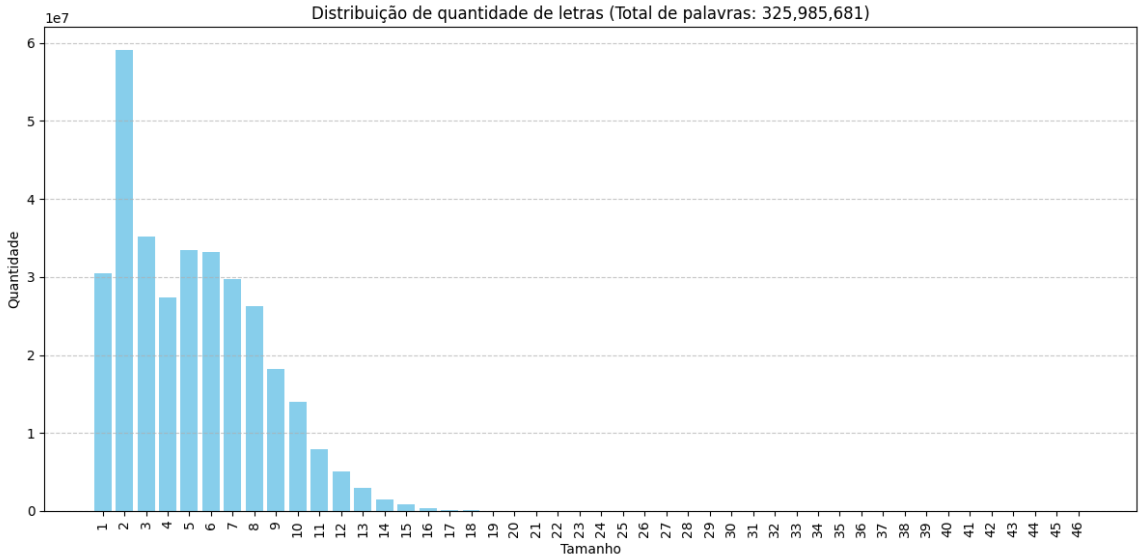


Figura 2: Distribuição do tamanho das palavras em explorações prévias

No entanto, apesar do tamanho do *dataset* da *Wikipedia* possuir um tamanho aprox. 2180 vezes maior, neste trabalho foi-se optado pela utilização do córpus de textos jurídicos devido a limitações de processamento, bem como questões de padronização e qualidade.

Outro detalhe importante de se observar é que, apesar de que os gráficos seguem até o valor 46, neste trabalho foram apenas utilizadas palavras de até aprox. 22 letras.

3 Implementação 1: Gerador de palavras

3.1 Fundamentação

Um gerador de palavras é um algoritmo que permite a criação de palavras fictícias novas, e pode ser implementado de diversas abordagens. Uma primeira abordagem que poderia ser pensada para a implementação de um gerador de palavras novas é um método puramente aleatório, com pesos baseados nos obtidos pela análise da distribuição na linguagem. Essa abordagem é rasa pois não leva em consideração relações entre letras, como por exemplo, a letra 't' não vem seguida de 'x' no português. Logo sendo uma ideia simples porém ineficiente.

Outra ideia a ser considerada é a utilização de uma CM simples (homogênea no tempo), que já de início traria resultados melhores do que uma abordagem que não considera a relação entre letras (este método é abordado na análise do comportamento da matriz na seção posterior). No entanto, a fim de se buscar observar tanto as relações entre as letras quanto entre letras e posições na palavra, utilizar-se-á o método descrito a seguir.

Seja $p = a_0a_1a_2...a_n$ uma sequência, onde a_i é uma letra do alfabeto latino (acrescido dos seguintes artefatos: letras acentuadas, palavra vazia (SOW) e fim de palavra (EOW)) e $1 \leq n \leq 22, n \in \mathbb{Z}$ (limite definido para este trabalho). Serão então montadas 22 matrizes (na realidade, é uma matriz com três dimensões 42x42x22), denotadas por $M^{(j,j+1)}$, tais que, $\forall i \in \{1, ..., 22\} | M^{(i,i+1)}$ é uma CM que representa a distribuição de letras observadas entre a_ia_{i+1} na base textual inicial.

A construção dar-se-á da seguinte forma:

- Inicia-se l como ϕ, i.e, $a_0 = SOW$ - *Start of word*, uma cadeia vazia
- Escolhe-se a_1 por meio de aleatoriedade, utilizando-se a matriz $M^{(0,1)}_{a_0=SOW}$, i.e, a linha SOW da primeira matriz.

- Verifica-se se o carácter escolhido foi EOW, se sim, a palavra criada foi l, senão, acrescenta-se o carácter escolhido à l e repete-se o passo anterior para $a_1, a_2, ..., a_{21}$, utilizando-se, respectivamente,as matrizes $M_{a1}^{(1,2)}, M_{a2}^{(2,3)}, ..., M_{a21}^{(21,22)}$.

Ao fim do processo, l representa uma palavra criada. Note que aqui não se leva em consideração prefixos, sufixos nem radicais, o que necessitaria de uma análise mais aprofundada na parte linguística, o que não é o foco deste trabalho.

A principal vantagem de se utilizar esse método ao invés de uma única CM simples é que ele permite o 'aprendizado' de padrões que são dependentes da posição da palavra. Um exemplo é a sequência '-ção', que é um padrão comum porém que não ocorre no início de palavras.

Outra vantagem, também importante, é que, como cada matriz possui uma chance de levar ao estado EOW, espera-se que a distribuição amostral do tamanho das palavras seja próximo ao observado no corpus sem a necessidade de se adicionar critérios extras de parada ou alterar de maneira artificial a geração.

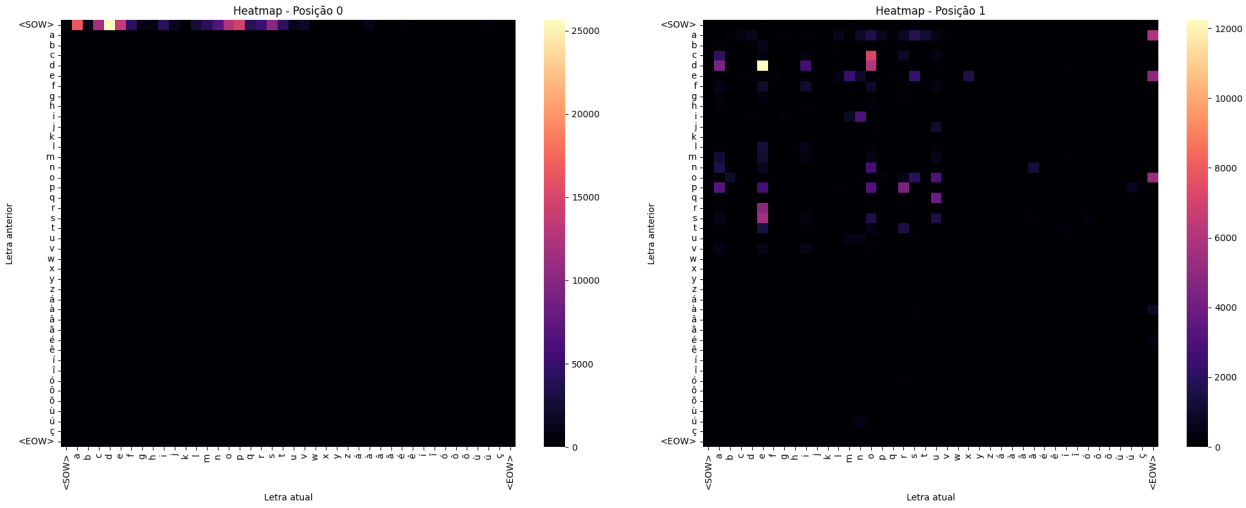
3.1.1 Implementação 1.1

A fim de se realizar análises mais findadas em Cadeias de Markov tradicionais, utilizar-se-á também uma implementação muito parecida com a anterior, porém com todas as matrizes sendo agrupadas em uma única. O processo de construção se dá de maneira igual, porém com a diferença é que agora a utilização não mais varia com as linhas, i.e, $M^{(0,1)} = M^{(1,2)} = \cdots = M^{(21,22)}$, cujos valores são dados por $M' = \sum_{n=0}^{21} M^{(n,n+1)}$.

3.2 Análise

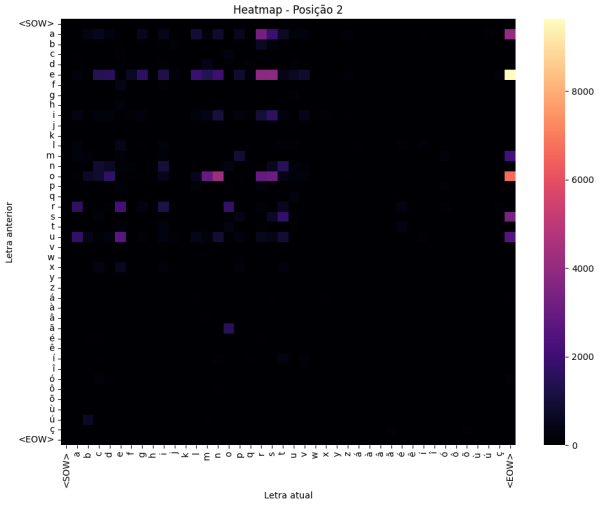
Uma observação importante é que, para facilitar algumas operações de E/S e outros detalhes de implementação, as matrizes foram implementadas com valores brutos ao invés de probabilidades. Os valores brutos são transformados em probabilidades durante a execução e conforme a necessidade.

Em um primeiro momento, é interessante se observar um mapa de calor bruto das 22 matrizes relevantes, que estão dispostas logo a seguir:

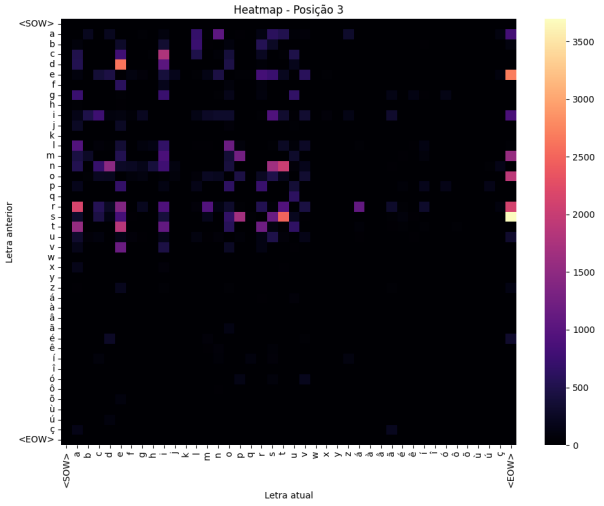


(a) Heatmap 0

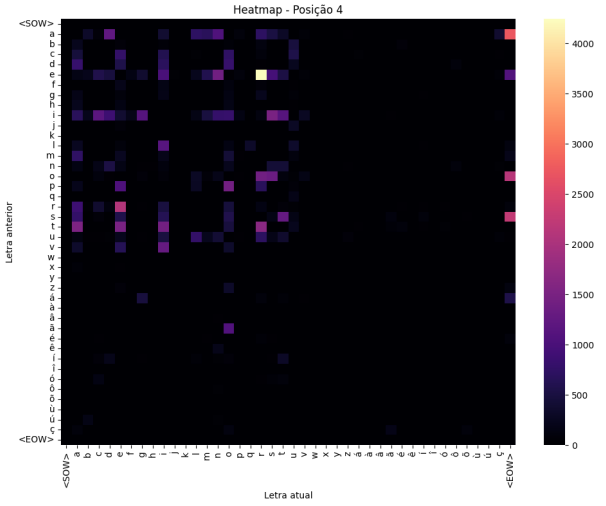
(b) Heatmap 1



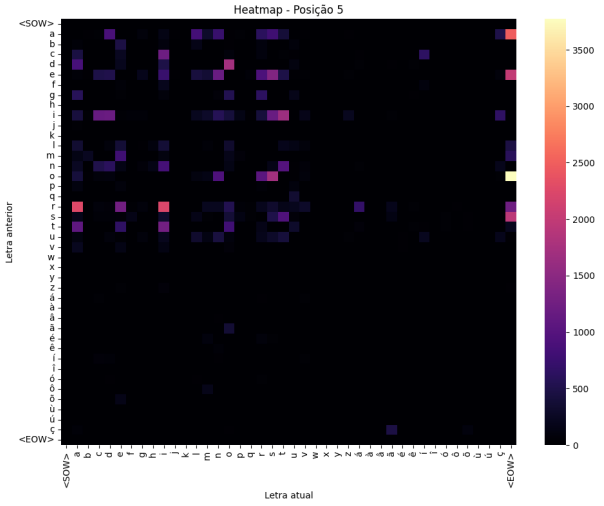
(c) Heatmap 2



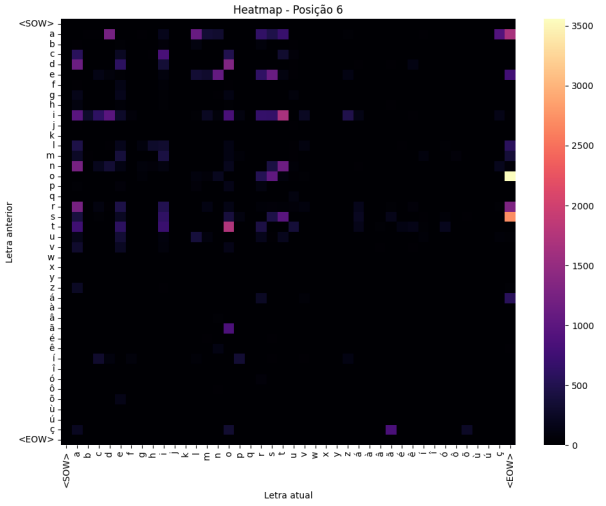
(d) Heatmap 3



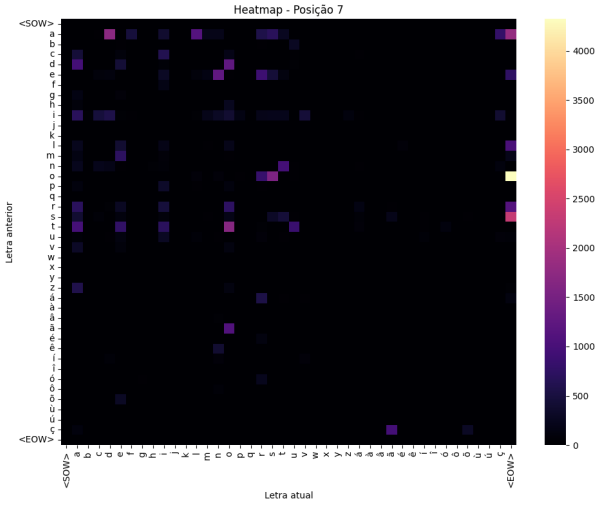
(e) Heatmap 4



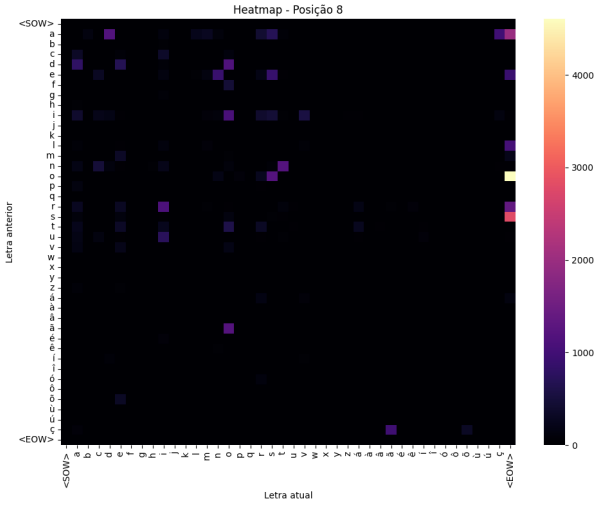
(f) Heatmap 5



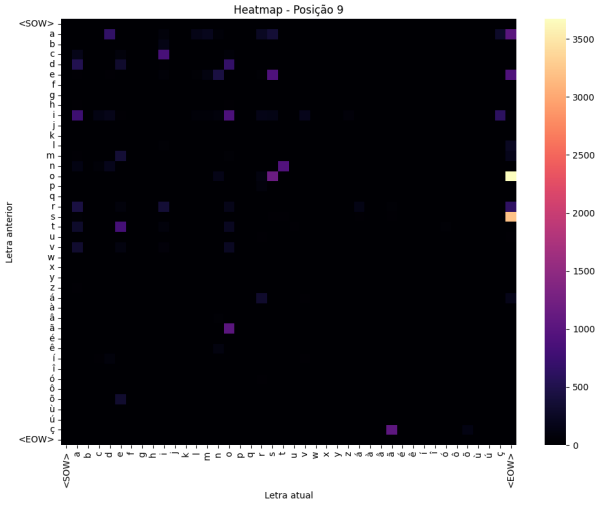
(g) Heatmap 6



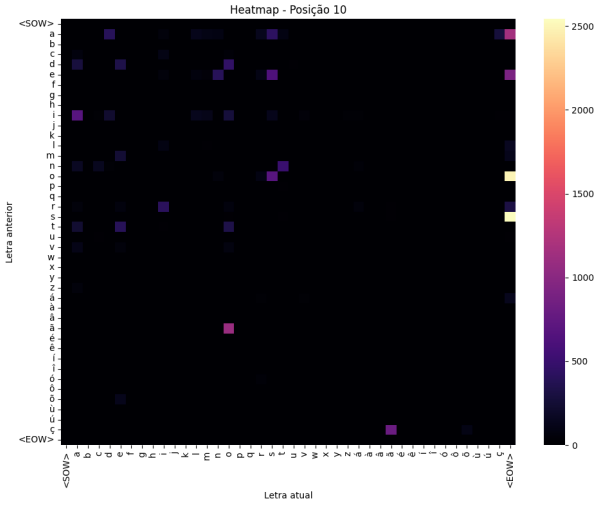
(h) Heatmap 7



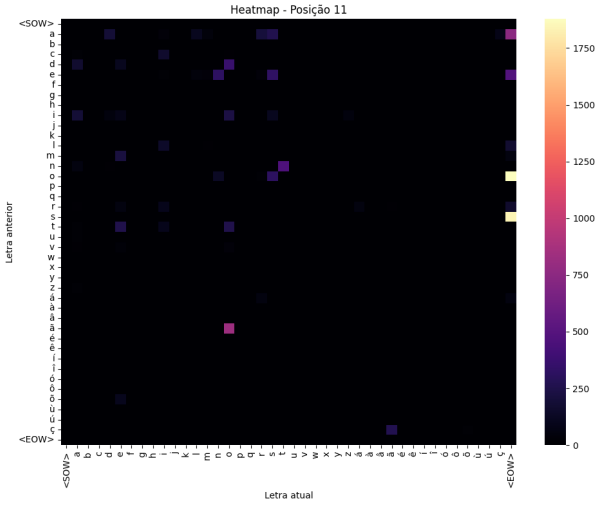
(i) Heatmap 8



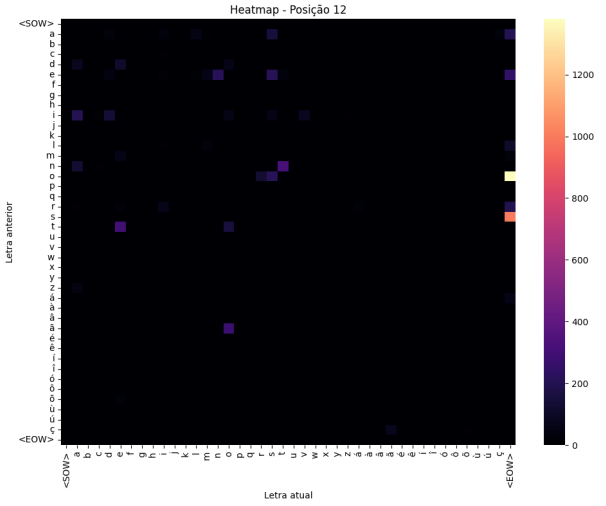
(j) Heatmap 9



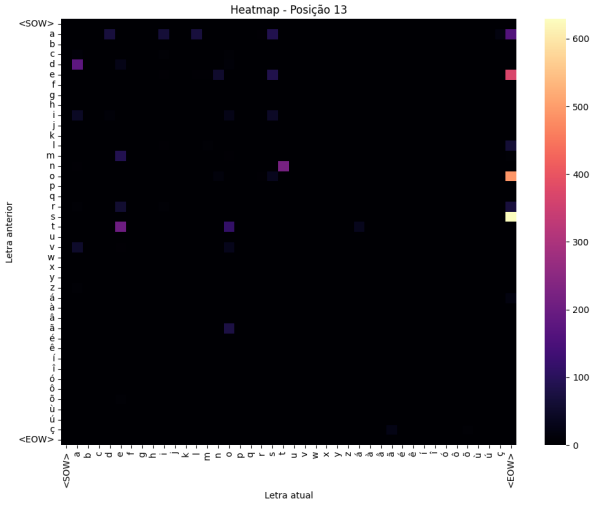
(k) Heatmap 10



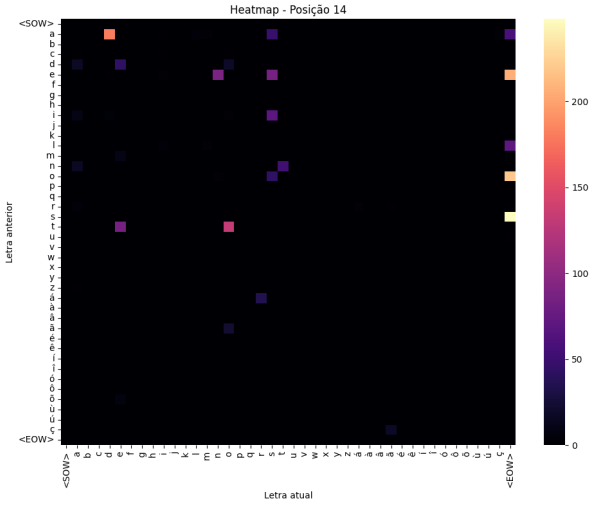
(l) Heatmap 11



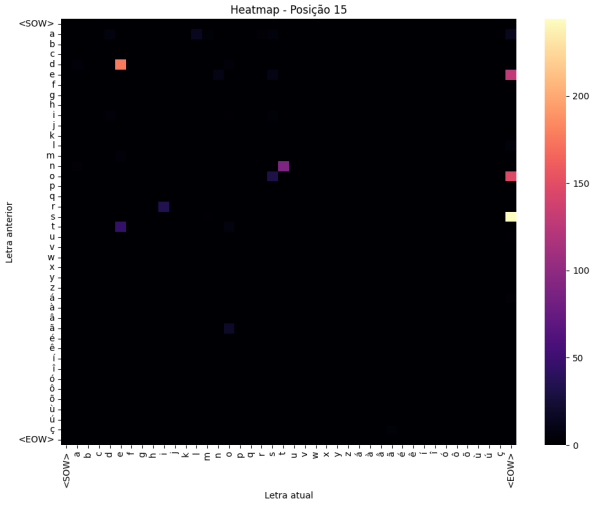
(m) Heatmap 12



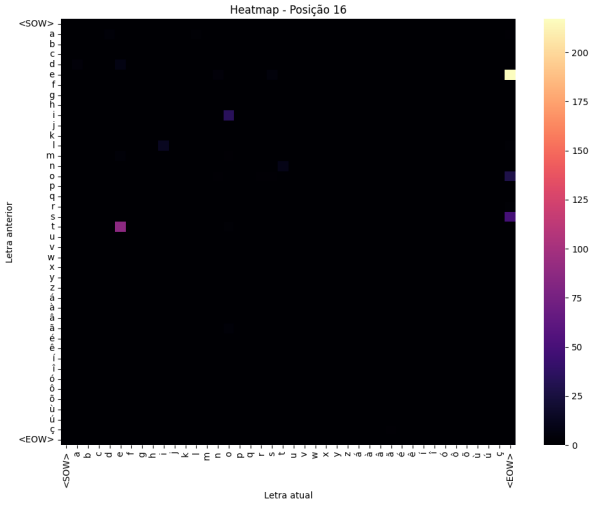
(n) Heatmap 13



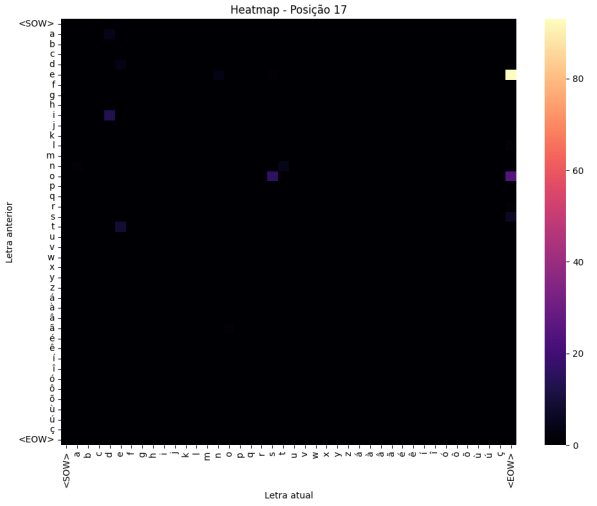
(o) Heatmap 14



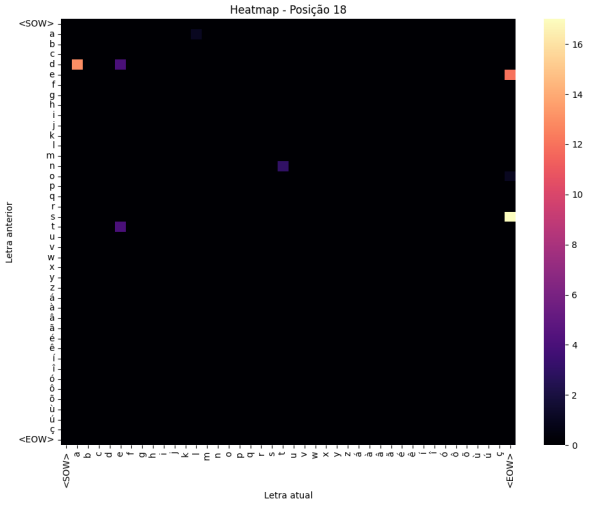
(p) Heatmap 15



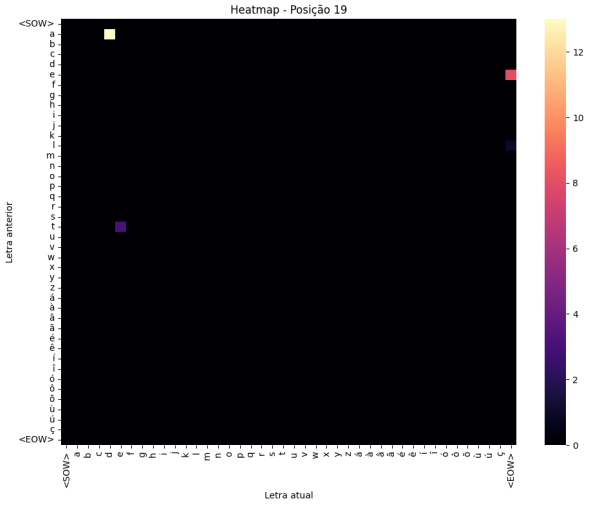
(q) Heatmap 16



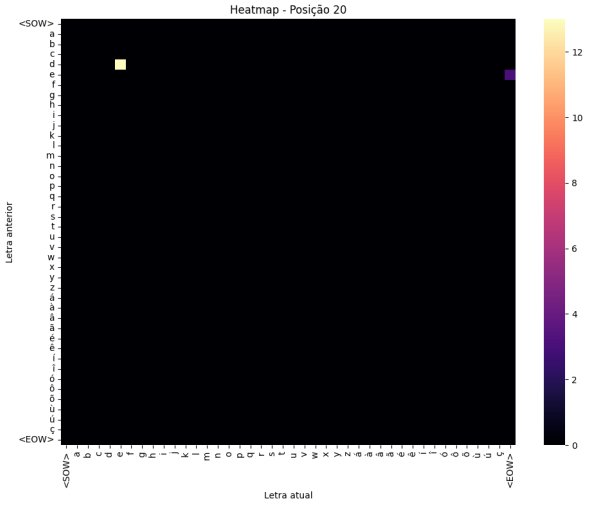
(r) Heatmap 17



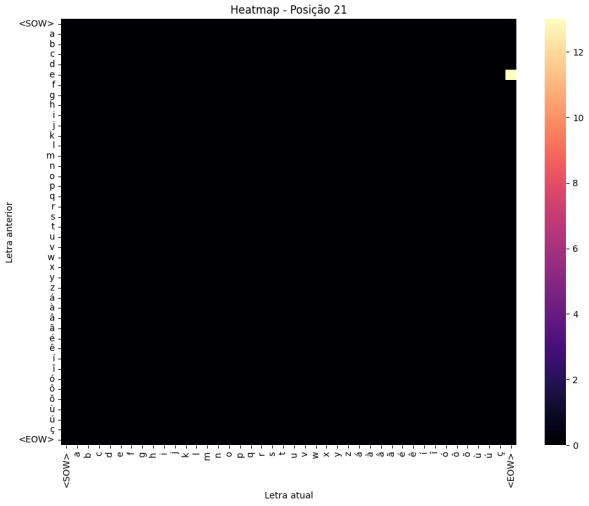
(s) Heatmap 18



(t) Heatmap 19



(u) Heatmap 20



(v) Heatmap 21

Observando os mapas de calor, é interessante notar alguns artefatos:

1. O *heatmap* 0 (Figura 3a) é distinto dos demais pois, como toda palavra se inicia com $< SOW >$, todas as outras linhas da matriz senão essa possuem valores 0. É interessante notar que esta linha coincide com a distribuição da quantidade de vezes que uma palavra se inicia com uma determinada letra, e a matriz poderia ser reduzida a um vetor representado as possibilidades de estados iniciais.

2. É interessante acompanhar nos *heatmaps* 1 e 2 a construção de palavras comuns de duas letras, como "de, do, da, os, as, no, em, ...". Elas aparecem com bastante frequência, inclusive levando a um pico no tamanho das distribuições das palavras.

3. Uma distinção importante que pode ser observada do *heatmap* 6 em diante e que não aparece com vigor anteriormente são dois pontos: "ç $\longrightarrow$ ã"e "ã $\longrightarrow$ o". Isso é devido a terminação '-ção', que é observada mais frequentemente em palavras maiores. Outras terminações podem ser observadas também, como a terminação '-ente'.

4. O item anterior é uma grande justificativa do porquê de se realizar a separação em 22 matrizes diferentes. Uma matriz única possuirá menos espaço para tais construções surgirem naturalmente durante a construção.

Com a observação das matrizes, pode-se realizar uma análise para se aferir se, de fato, a construção de palavras por esse método leva a uma distribuição similar a observada no córpus textual. Para isso, foram realizadas 9999 medições, e o resultado obtido foi:

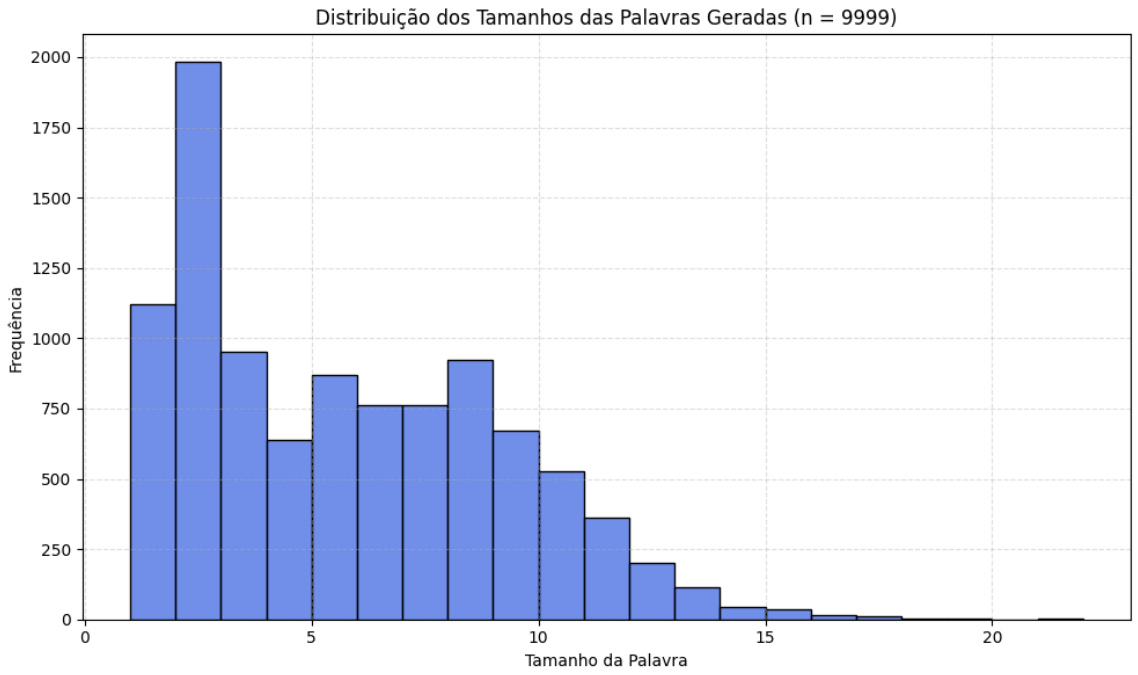


Figura 3: Distribuição no tamanho de palavras obtidas experimentalmente

Aqui, foi observada a média 5.3, mediana 5 e a moda 2. Alguns exemplos ilustrativos de palavras que foram geradas por esse método incluem:

- pasidala
- sentedimes
- cráritoricio
- netremíntação
- sócorações
- umaseita
- cossoco
- peladelontas
- distrão
- ler
- servistora
- ronão
- câmido
- serifípios

Além disso, é interessante notar que, ocasionalmente, esse método gera palavras existentes, alguns exemplos gerados incluem: ”eco, não, coração, dirá, fatal, acha, somos, expulso, obras, doca, pompos, ler”, entre outras. Esses são os exemplos com 3 ou mais letras, porém palavras simples como ”de, do, da, os, as, em, na, ...”são geradas frequentemente também.

Com isso, considera-se que a implementação 1 foi realizada com sucesso. No entanto, a fim de se obter uma análise mais findada em Cadeias de Markov tradicionais, também foi realizada a implementação 1.1, que possui como matriz a soma de todas as 22 matrizes observadas anteriormente, que equivale ao processo tradicional de amostragem com uma única matriz, uma vez que as matrizes utilizaram valores brutos.

3.2.1 Implementação 1.1

Com a implementação 1.1, a única matriz obtida possui o visual:

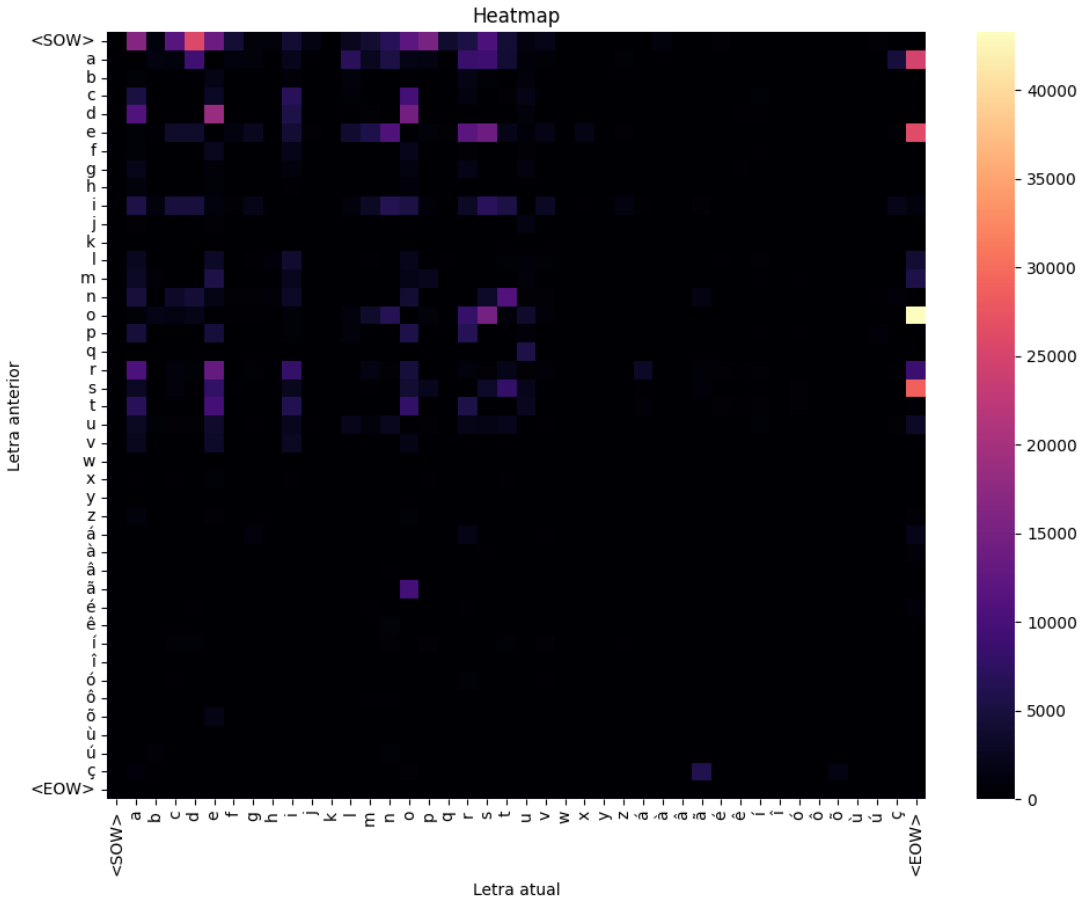


Figura 4: *Heatmap* acumulado (valores brutos)

No entanto, para essa matriz é interessante a normalização dos valores:

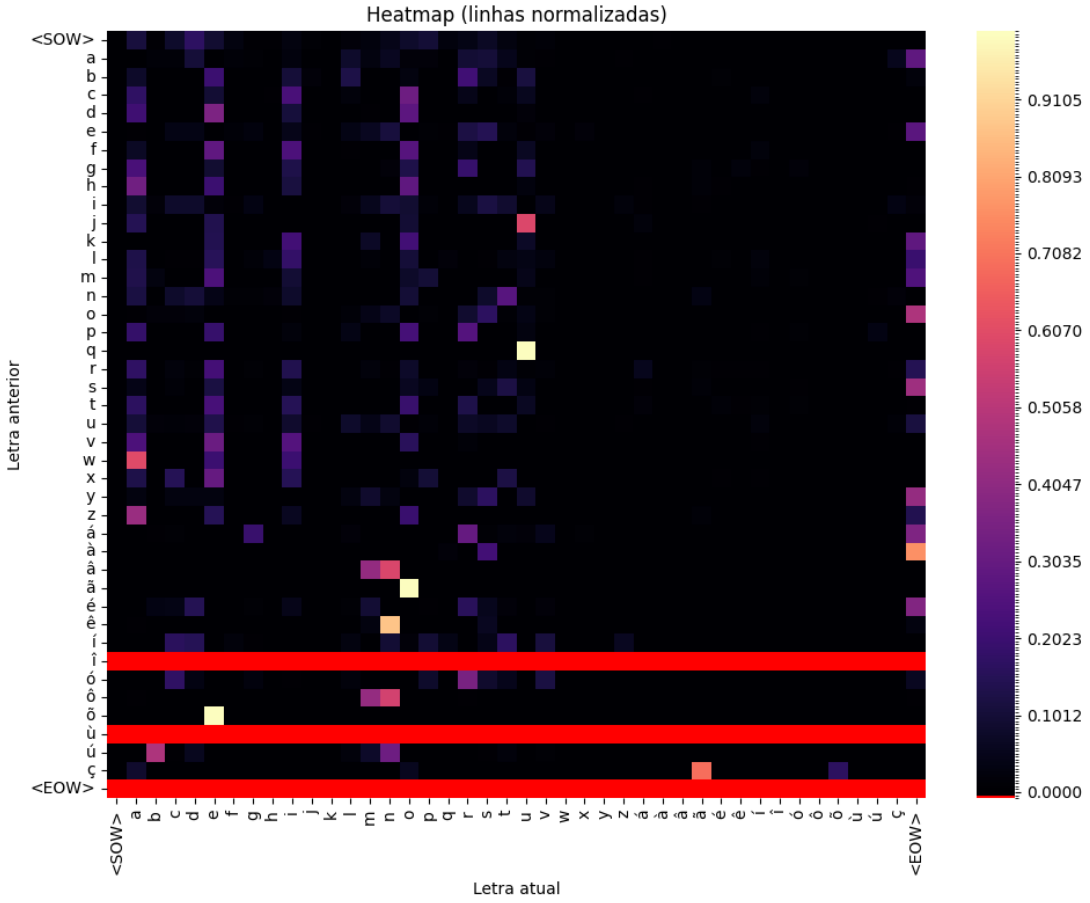


Figura 5: *Heatmap* acumulado (valores normalizados)

Um ponto importante de se destacar é que, apesar das letras î, ù terem sido adicionadas ao conjunto de letras (por motivos de compatibilidade com *datasets* explorados anteriormente que continham palavras de diferentes épocas históricas), essas letras não apareceram no corpus utilizado, por isso todos os valores de suas linhas são zero, logo, ficando vermelhas.

Outro ponto interessante é que, como o processo termina com o encontro de $< EOW >$, o processo não contabilizou nenhuma ocorrência que parte de $< EOW >$ para outra letra. No entanto, para os fins de análise Markoviana, aqui considera-se que o estado $< EOW >$ transiciona para si mesmo com probabilidade 1 indefinidamente.

Com esses pontos enunciados, pode-se chegar de fato na matriz final para essa implementação:

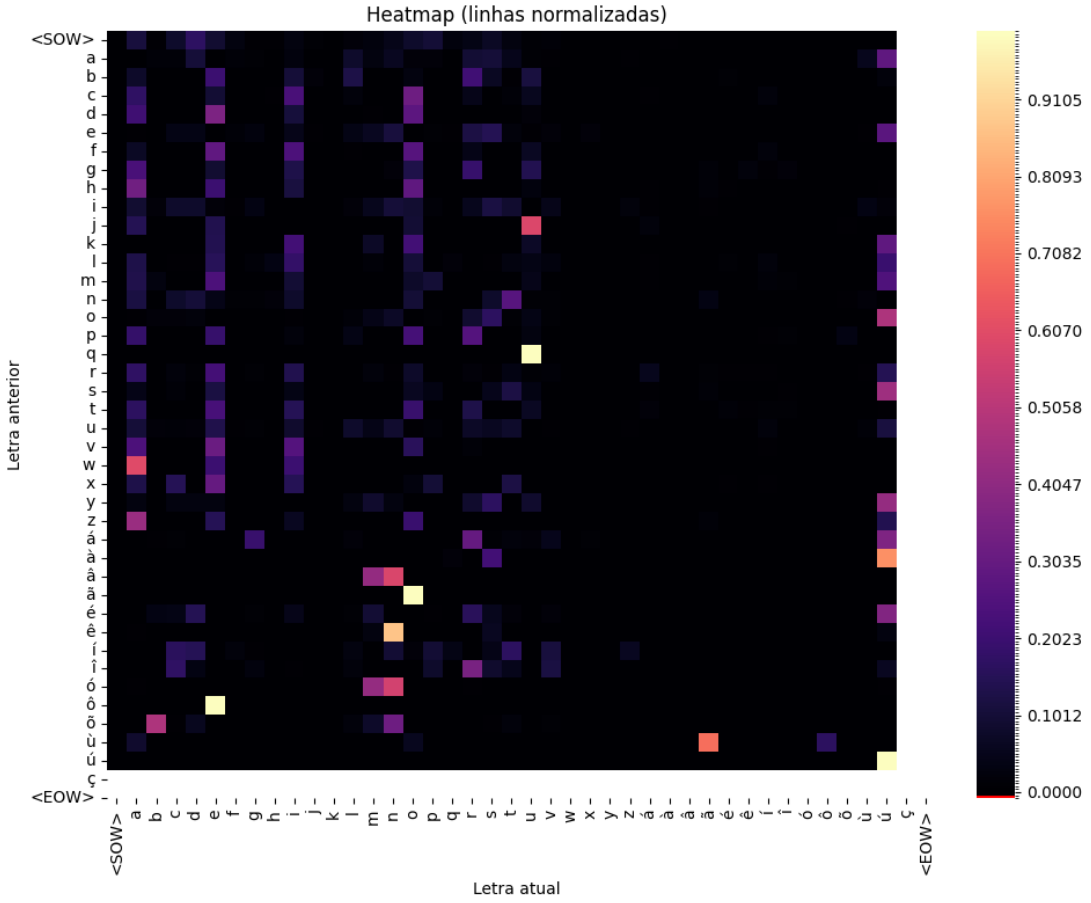


Figura 6: *Heatmap* final acumulado (valores normalizados e limpos)

Um artefato muito interessante de ser observado nesta matriz é o aparecimento de 'linhas verticais' de probabilidades onde os estados transicionam para vogais. Uma possível explicação é o fato de que, em uma sílaba, há diversas possibilidades de consoantes que realizam a transição para um conjunto bem menor de vogais, assim levando ao surgimento destas linhas.

No entanto, para efeitos de geração de palavras, ela possui um "desempenho"pior (efeito subjetivo), levando à geração de palavras menos plausíveis, como por exemplo:

- ividinirentordategarm
- danstristostexe
- dotecuanomo
- tinfocarar
- emisevemer
- eturórciza
- plitintise
- fodestreutuziruicaoram
- abrefrico
- ditadecadecu

- rastpirticicíza

A distribuição dos tamanhos das palavras também foi afetada:

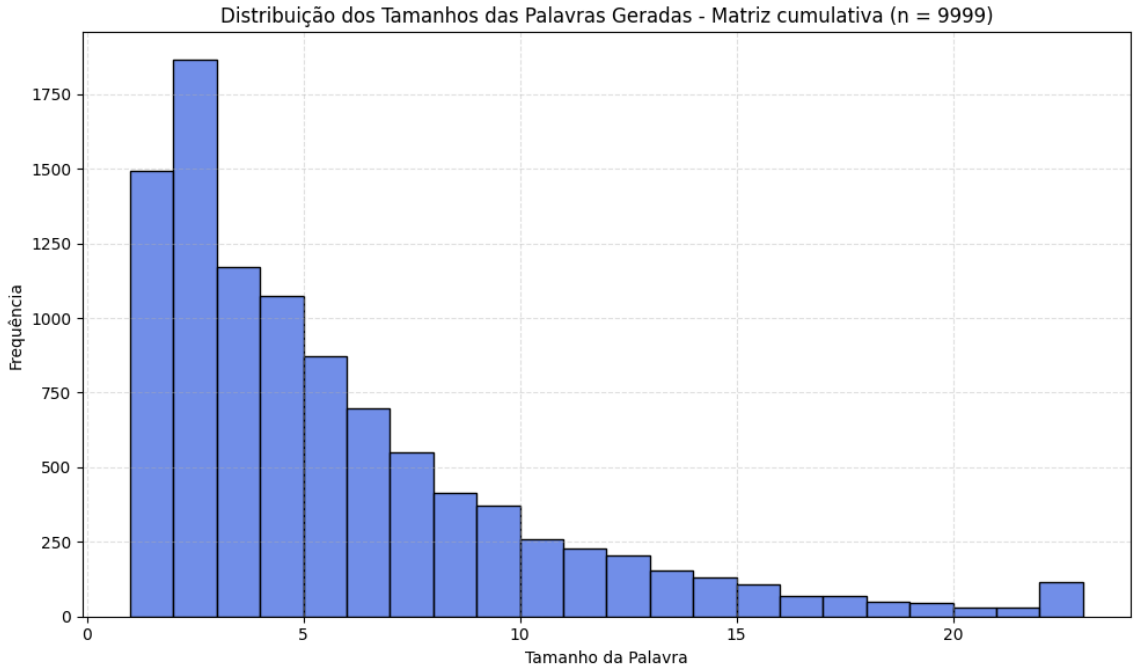


Figura 7: Distribuição no tamanho de palavras obtidas experimentalmente - implementação 1.1

Aqui, foi observada a média 5.28, mediana 4 e a moda 2. A média é similar a obtida com o método anterior, porém a distribuição em si é alterada.

Análise do primeiro passo

Como a matriz gerada nessa implementação 1.1, ao contrário da anterior, é contida em uma única matriz de probabilidades, ela representa de fato uma Cadeia de Markov. Com isso, é possível realizar-se algumas análises vistas em sala de aula sobre ela.

Primeiramente, um fato a se considerar é que, como há apenas 1 estado absorvente ($< EOW >$), não há sentido em calcular os valores de U, pois todos os começos levam para esse estado. Além disso, não será realizada a visualização da Cadeia de Markov por meio de um grafo pois, com 40 estados, haveriam 1600 transições nesse, o que torna a visualização imprática e poluída. Para isso, a visualização será feita por meio de *heatmaps*.

No entanto, pode-se calcular a matriz de trabalho W bem como o tempo médio que leva para cada estado ser absorvido, V. A matriz $W = (I - Q)^{-1}$ pode ser visualizada no *heatmap* a seguir:

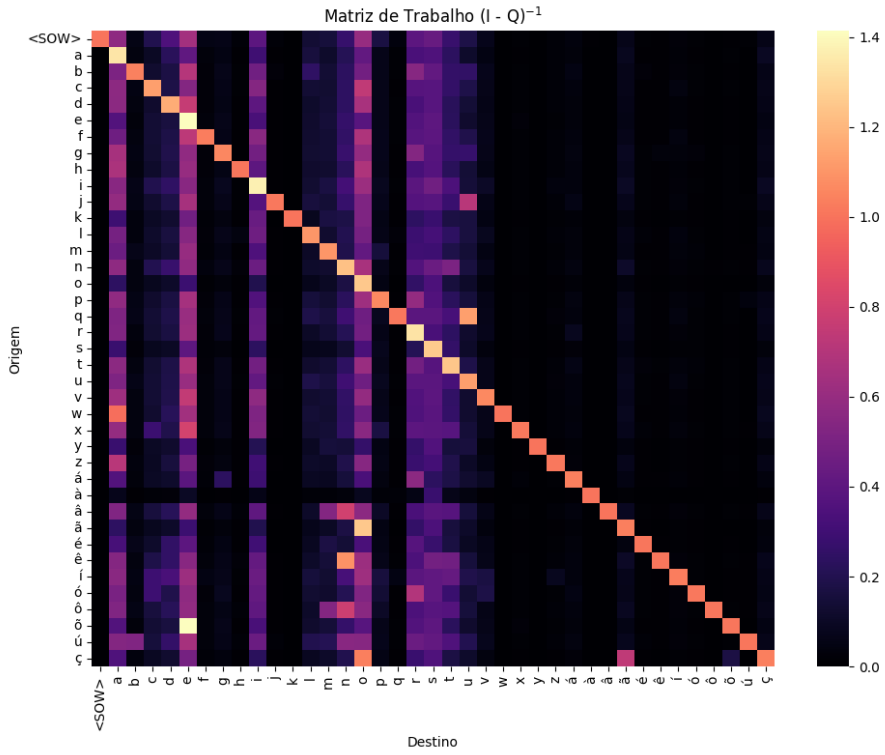


Figura 8: Matriz de trabalho

Por sua vez, o tempo que cada estado leva para ser absorvido pode ser observado na tabela a seguir. É interessante notar que, para o estado inicial $< SOW >$, a palavra tem 0 letras, então a quantidade média de letras que uma palavra terá será de $V_{<SOW>} - 1$, que, no caso, é de 5.3517, o que se aproxima da distribuição amostral observada de 5.28.

Estado	Tempo médio até absorção
$< SOW >$	6.3517
a	4.8614
b	6.1354
c	5.9709
d	5.7459
e	4.9393
f	6.0302
g	6.2271
h	5.7038
i	6.2137
j	6.3409
k	4.5977
l	5.2731
m	4.9483
n	6.3518
o	3.6912
p	5.9346
q	6.8264
r	5.4132
s	3.8939
t	6.0105
u	5.8264
v	6.0769
w	6.1474
x	6.4812
y	3.9186
z	5.0384
á	4.6606
à	1.9586
â	6.7701
ã	4.6931
é	4.4467
ê	6.9739
í	6.7987
ó	6.2493
ô	6.7254
õ	5.9393
ú	7.0550
ç	5.8545

Tabela 1: Tempo médio de absorção.

4 Implementação 2: Gerador de frases

4.1 Fundamentação

Essa implementação não é o foco principal deste trabalho e foi feita por curiosidade de se explorar o tópico, uma vez que a base de dados e o processamento são bem parecidos, não necessitando de grandes alterações.

Um gerador de frases é um algoritmo similar ao anteriormente apresentado, porém que busca realizar a escolha de palavras como estados ao invés de letras individuais. Assim, segue-se a seguinte

notação: $M^{(i,i+1)}$ é a matriz representando a CM da posição i para $i+1$ da frase. A frase é uma sequência $f = a_0 a_1 ... a_n$, onde cada a_i é uma palavra da língua portuguesa ou um *token* especial. Os tokens especiais incluem:

- $< SOP >$: *Start of phrase*, que representa uma frase vazia, ainda sem palavras
- $< EOP >$: *End of phrase*, que representa o fim de uma frase.

Note que aqui o token EOP poderia ser trocado por pontos finais (.), pontos de exclamação (!) e pontos de interrogação (?). No entanto, devido ao corpus utilizado, essa representação não foi necessária então utilizou-se o token EOP.

No entanto, há dois problemas principais de se utilizar CMs para esse fim:

- Uma única matriz de transição ocuparia um tamanho exorbitante de memória. Em um primeiro momento, utilizando-se todas as palavras, a construção das matrizes tentou alocar 32.1 GiB de memória, o que é inviável.
- Esse método de se gerar as sentenças não possui contexto/memória sobre o que está sendo gerado. Assim, as frases muitas vezes acabam não fazendo sentido lógico. Por exemplo, a frase: *"O dia estava nublado e chuvoso, ..."* pode ser completada com *"O dia estava nublado e chuvoso, a rainha da Antártica comeu frango roubou carros"*.

Para se evitar o primeiro problema do limite de memória, foi-se realizado um recorte do vocabulário de tal maneira que palavras que foram observadas menores que um número k=5 vezes durante o processamento não foram inseridas na matriz. Isso conseguiu reduzir o tamanho do vocabulário de 11961 para apenas 1748 palavras, permitindo o processamento da matriz.

No entanto, quanto ao problema de falta de contexto, apesar de diversas tentativas de se implementar uma espécie de memória baseada nas palavras anteriores, nenhuma delas surtiu efeito notável positivo na coerência, logo, não serão colocadas neste relatório.

4.2 Análise

Como o intuito desta exploração não é o foco deste trabalho, bem como por causa da quantidade de estados (1748), a visualização das matrizes se torna problemática. Logo, a análise estatística dessa matriz não será feita igual a implementação anterior. No entanto, o método de geração funciona, e alguns exemplos de frases geradas foram:

- *a lei disporá sobre as normas gerais do estado do negócio jurídico não será inferior a sua vontade se agiu de cada fração de qualquer modo que lhe*
- *o terceiro não terá o direito de autorização do conselho da república não havendo*
- *o mandatário é permitido ao portador se faz seus herdeiros a assegurar formação básica comum e a prescrição antes da nova*
- *na dúvida entre os efeitos da laje responderá pelos preceitos*
- *o dolo do cumprimento das sociedades de seus créditos em que realizar avaliação das instituições da obrigação de indenização*
- *todos os casos previstos em lei complementar referida neste artigo só extingue a sociedade nacional ou em que tenha*
- *subseção da eficácia do registro não será de contas do credor contra o direito de crédito rural*

- *o regimento interno poderá o faça nos parágrafos anteriores será o casamento é exercida em qualquer veículo de qualquer interessado contra ele senão de seus atos pelo valor das*
- *o vendedor somente poderá o faça nos quinze dias da operação interesse contrário ao regime especial para o dono deste o fato ao distrito federal ou do consentimento*
- *outras receitas dos indígenas será de um dos casos de iniciativa dos ministros que tenham destinação do interesse a medida*
- *a lei definirá os ministros aposentados do risco de direito a firma ou para as condições de iniciativa do fato*
- *a insolvência pode ser feito ao credor e por expressa disposição de pessoa cujo direito de sua vontade diretamente ou os bens do cargo o cálculo a taxa*
- *capítulo da organização dos bens conferidos ao registro do sistema penal da dívida e o fiador pode opor ao órgão do ministério público e a do direito de imóveis*
- *a cláusula expressa em favor de recursos públicos em falta de efeitos perante terceiros dos recursos e no de vontade de que por seu responsável ou da data das*
- *ressalvados os negócios jurídicos que somente poderão dispor sobre a realização de sua residência com ele e novo*
- *se o adquirente dos bens conferidos ao mandatário a retribuição prevista em ato e o caput deste fato do poder executivo as destinadas à defesa do ônus de evicção*
- *o uso da apólice deverá constar da autoridade competente para qualquer lesão*
- *o direito cabe ao titular da nacionalidade brasileira sem reserva de que houver sido estipulada no objeto e o qual poderá ser realizada de precatórios por ele o vizinho*
- *as alíquotas de prescrição por objeto da sucessão provisória só com as prestações pagas até a câmara dos municípios não tiver sido*
- *quando o impedimento for de mais da lei de quota*
- *nos contratos de deputados à sua responsabilidade*
- *o supremo tribunal superior do conselho fiscal será reconhecido ao dono do interesse contrário ao presidente da administração pública federal para os seus direitos e dos membros da pessoa*

Aqui, a geração de frases observa a distribuição disposta a seguir. É importante notar que o tamanho das frases foram limitadas a 30 palavras.

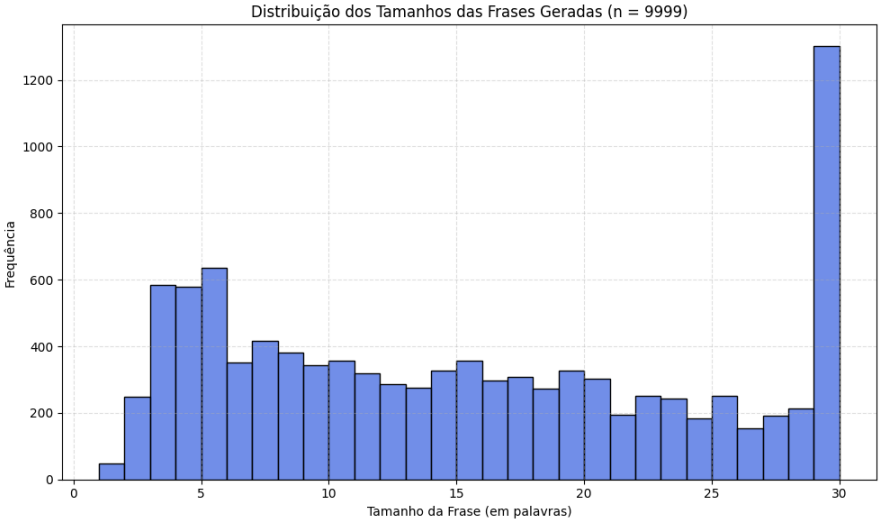


Figura 9: Distribuição no tamanho de frases obtidas experimentalmente

No entanto, ao contrário da implementação 1, o formato da distribuição não foi mantido quando comparado com a distribuição observada no corpus puro. Isso ocorreu pois, devido ao processo de redução de vocabulário, possíveis caminhos que levariam à frases mais lexicalmente diversos foram cortados em detrimento do tamanho necessário. Isso alterou o formato da distribuição. O formato observado originalmente está a seguir.

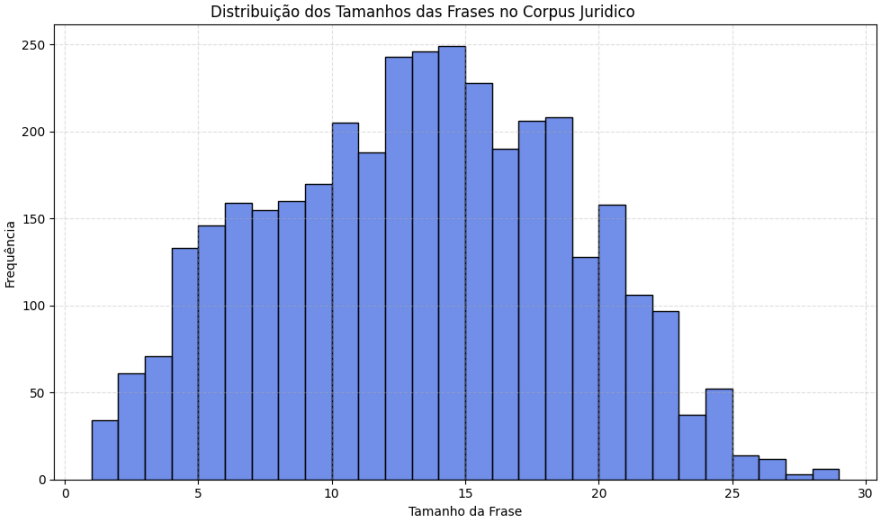


Figura 10: Distribuição no tamanho de frases observadas no corpus jurídico

5 Conclusão

Este trabalho apresentou a construção e análise de modelos geradores de palavras e frases baseados em Cadeias de Markov, utilizando um corpus jurídico como base textual. A abordagem de múltiplas matrizes posicionais mostrou-se eficaz para capturar padrões fonotáticos e estruturais da língua portuguesa, resultando em palavras mais plausíveis em comparação com o modelo simplificado de matriz única. Além disso, a ampliação do método de trabalho para utilizar palavras como estados permitiu a geração de frases que, em grande parte, corretamente seguem sequências sintáticas da língua portuguesa.

6 Mensagem

Olá professor, tudo bem?

Essa mensagem não é parte do trabalho, mas decidi colocá-la para agradecê-lo. Eu sempre tive interesse em explorar esse tema de maneira mais aprofundada, e realizei algumas implementações bem mais simples antigamente. No entanto, não tinha a base teórica para conseguir um bom resultado.

Achei muito interessante explorar isso com Cadeias de Markov pois elas guardam a relação entre letras, e acredito que dê para ir mais a fundo caso sejam trabalhados com *tokens* (partes de palavras). Inclusive, essa relação é muito atual hoje em dia pois é de maneira similar que ferramentas de IA muito populares (modelos generativos) sintetizam linguagem, e é muito interessante que seja possível utilizar a estatística desta forma (claro, os resultados não serão os mesmos que os feitos com técnicas mais complexas e muito treinamento, mas apenas o fato de ser algo possível de ser feito é muito interessante por si só). Acredito que consegui entender muito melhor o quanto a estatística é a base de muito o que está atual hoje, e a importância dos dados no mundo.

Imagino que esses tipos de modelagens possam ter muitas possíveis aplicações escondidas por aí no dia a dia, conseguem envolver muitas áreas de estudo também (biologia, agronomia, física, computação, economia, etc.), e demonstra o quão o quão poderosas elas podem ser quando utilizada de maneiras diferentes. Achei demasiadamente interessante, até por que a modelagem se parece em alguns pontos com os Autômatos Determinísticos Não-Determinísticos (AFNDs) estudados em computação, e eles são métodos de modelagens bem poderosos também.

Muito obrigado, professor

Referências

[1] LEWAND, R. E. *Cryptological mathematics*. American Mathematical Soc., 2000. v. 16.

[2] DOBROW, R. P. *Markov chains: First steps*. John Wiley Sons, Ltd, 2016. Cap. 2, p. 40–75.

[3] DOBROW, R. P. *Markov chains for the long term*. John Wiley Sons, Ltd, 2016. Cap. 3, p. 76–157.