Atribuição de Autoria usando PPM

Bruno Cunha Coutinho[1,2], Jalmaratan Luís de Melo Macêdo[1,3]
Aroldo Rique Júnior[1], Leonardo Vidal Batista[1]

[1]Departamento de Informática – Universidade Federal da Paraíba (UFPB)
Cidade Universitária – João Pessoa – PB – Brasil

[2]Procuradoria da República no Município de Campina Grande
Rua Elias Asfora, 67 - Centro – Campina Grande – PB – Brasil

[3]Procuradoria da República no Amazonas
Av. André Araújo, 358 - Aleixo – Manaus – AM – Brasil

bruno@prpb.mpf.gov.br, jalmaratan@yahoo.com.br,
aroldo.rique@terra.com.br, leonardo@di.ufpb.br

Abstract. *Many natural language processing techniques have been applied to the problem of authorship attribution, including neural networks, bayesian classifiers, and the Cusum method. Efficent data compression algorithms, such as prediction by partial matching (PPM) and Lempel-Ziv algorithms have also been investigated. However, there seems to be very few researches in authorship attribution for Portuguese texts. This paper presents a method based on PPM-C compression algorithm for authorship attribution applied to brazilian literature texts.*

Resumo. *Diversas técnicas de processamento de linguagens naturais já foram aplicadas ao problema de atribuição de autoria, incluindo redes neurais, classificadores bayesianos e o método Cusum. Algoritmos eficientes de compressão de dados, tais como o prediction by partial matching (PPM) e os algoritmos Lempel-Ziv, também foram alvo de investigação. Contudo, parece haver poucos pesquisadores em atribuição de autoria para textos em Português. Este trabalho apresenta um método baseado no algoritmo de compressão PPM-C para attribuição de autoria aplicada em textos da literatura brasileira.*

1. Introdução

Classificação (ou categorização) de texto consiste em atribuir um dado texto $\boldsymbol{x}$ a uma classe C_i, $i=1,2,...,N$. Essas classes podem representar autores (atribuição de autoria), estilos (determinação de escola literária), línguas (determinação de idioma/dialeto) ou qualquer outra característica lingüística que venha a convir. Características discriminantes, ou modelos estruturais ou estocásticos são usados para caracterizar as classes, e a classificação é realizada de acordo com alguma medida de similaridade entre $\boldsymbol{x}$ e cada classe.

Numa classificação supervisionada, sabe-se previamente que determinados *corpora* pertencem a determinadas classes. Cada *corpus* pré-classificado é usado para construir modelos ou para definir características discriminantes para cada classe, num processo denominado treinamento ou aprendizagem. Características discriminantes e modelos precisos são atributos importantes na construção de classificadores eficientes.

No âmbito da atribuição de autoria, diversas abordagens para construção de modelos já foram empregadas tais como redes neurais, classificadores bayesianos [1],

Cusum [2]. Contudo, algoritmos de compressão sem perdas têm sido aplicados na classificação de imagens [13, 14], na resolução de problemas de classificação de textos, etc., devido às suas habilidades em construir modelos estatísticos precisos. O método de compressão de dados *Prediction by Partial Match* (PPM) é o estado-da-arte em termos de algoritmo de compressão, e vários trabalhos relacionam o seu uso na classificação de textos [3, 4].

Classificadores baseados em modelos de compressão de dados tem várias vantagens potenciais quando comparados a modelos clássicos de treinamento: não existe seleção de característica, nenhuma informação é descartada – os modelos descrevem a classe como um todo [5]; não é necessário fazer considerações simplificadoras a respeito das distribuições de probablidades; a capacidade de construção adaptativa de modelos, por parte dos algoritmos de compressão, oferece um modo uniforme de classificar diferentes fontes de informação [5]; os métodos de classificação baseados em modelos de compressão são muito simples [6].

Este trabalho apresenta resultados obtidos em determinação de autoria de textos utilizando classificador baseado em compressor PPM. As principais contribuições do trabalho consistem em aferir pela primeira vez a eficiência do PPM na classificação de textos da língua portuguesa do Brasil e em evidenciar os resultados quando varia-se a ordem de Markov e os tamanhos dos arquivos de treinamento e classificação. Deve-se ressaltar que a língua portuguesa apresenta algumas peculiaridades que devem ser consideradas em uma etapa de pré-processamento dos textos, tais como a utilização de diacríticos. A Seção 2 aborda alguns conceitos fundamentais, a Seção 3 apresenta o algoritmo PPM, a Seção 4 descreve o método empregado, a Seção 5 apresenta os resultados experimentais e, finalmente, a Seção 6 discute os resultados obtidos e apresenta conclusões e sugestões para desenvolvimentos futuros.

2. Entropia e Modelo de Markov

Seja S uma fonte de informação discreta e estacionária que gera mensagens sobre um alfabeto finito $A = \{a_1, a_2,..., a_M\}$. A fonte escolhe símbolos sucessivos de A de acordo com alguma distribuição de probabilidade que, em geral, depende do símbolo precedente. Uma mensagem genérica é modelada como um processo estocástico estacionário $x=...x_{-2}x_{-1}x_0x_1x_2...$ com $x_i \in A$. Seja $x_n = x_1x_2...x_n$ representação de uma mensagem de tamanho n. Uma vez que $|A| = M$, a fonte pode gerar M^n mensagens diferentes de tamanho n. Seja x_i^n, $i = 1, 2,..., M^n$ a i-ésima dessas mensagens, de acordo com alguma ordenação, e assumindo que a fonte segue uma distribuição de probabilidade P, então a mensagem x_i^n é produzida com probabilidade $P(x_i^n)$.

Seja

$$G_n(P) = -\frac{1}{n}\sum_{i-1}^{M} P(x_i^n)\log_2 P(x_i^n) \qquad (1)$$

$G_n(P)$ decresce monotonicamente com n [8] e a entropia da fonte é:

$$H(P) = \lim_{n\to\infty} G_n(P) \text{ Bits/Símbolo} \qquad (2)$$

Uma formulação alternativa para $H(P)$ usa probabilidades condicionais. Seja $P(x_i^{n-1}, a_j)$ a probabilidade da seqüência $x_i^n = x_i^{n-1}a_j$ (x_i^{n-1} concatenado com $x_n = a_j$) e seja $P(a_j \mid x_i^{n-1}) = P(x_i^{n-1}, a_j).P(x_i^{n-1})$ a probabilidade de $x_n = a_j$ condicionada a x_i^{n-1}. Uma aproximação para a entropia de n-ésima ordem para $H(P)$ [8] é:

$$F_n(P) = -\sum_{i=1}^{M}\sum_{j=1}^{M} P(x_i^{n-1}, a_j)\log_2 P(a_j \mid x_i^{n-1}) \qquad (3)$$

$F_n(P)$ decresce monotonicamente com n [8] e a entropia da fonte é:

$$H(P) = \lim_{n\to\infty} F_n(P) \text{ Bits/Símbolo} \qquad (4)$$

A equação 4 envolve a estimação das probabilidades condicionada a uma infinita seqüência de símbolos anteriores. Quando uma memória finita é assumida as fontes podem ser modeladas por um processo de Markov de ordem n-1, tanto que $P(a_j \mid ...x_{n-2}x_{n-1}) = P(a_j \mid x_1...x_{n-1})$. Neste caso, $H(P) = F_n(P)$.

Defina a *taxa de codificação* de um esquema de codificação como o número médio de bits por símbolo que o esquema usa para codificar a saída da fonte. Um *compressor sem perda* é um esquema de codificação unicamente decodificável cujo objetivo é alcançar uma taxa de codificação tão pequena quanto possível. A taxa de codificação de qualquer esquema de codificação unicamente decodificável é sempre maior ou igual à entropia da fonte [8]. Esquemas de codificação ótimos têm uma taxa de codificação igual ao limite inferior teórico $H(P)$, assim alcançando compressão máxima.

Para processos de Markov de ordem n-1, a codificação ótima é alcançada se, e somente se o símbolo $x_n = a_j$ ocorrendo depois de x_i^{n-1} é codificado com $-\log_2 P(a_j \mid x_i^{n-1})$ bits [8, 9]. Entretanto, pode ser impossível estimar precisamente a distribuição condicional $P(.\mid x_i^{n-1})$ para grandes valores de n devido ao crescimento exponencial do número de diferentes contextos, que traz problemas bem conhecidos, como a diluição de contextos [9].

3. O PPM

O algoritmo PPM, introduzido em 1984 por Cleary e Witten [10], tornou-se o algoritmo de compressão sem perdas estado-da-arte. Por sua vez, Moffat apresentou uma variação do PPM, o denominado PPM-C [7], cuja implementação apresentou excelente desempenho, permitindo assim seu uso comercial.

PPM consiste numa técnica de modelagem estatística de contexto finito. A idéia por trás do PPM é gerar a probabilidade condicional para o caracter atual armazenando o contexto dos últimos n caracteres. Modelos que condicionam suas predições em símbolos imediatamente anteriores são chamados modelos de contexto finito de ordem k, onde k é o número de símbolos precedentes usados na predição. PPM emprega um conjunto de modelos contextuais de ordem fixada com diferentes valores de k, limitado superiormente a um valor, para predizer os caracteres subseqüentes.

A maneira mais simples para construir o modelo é manter um dicionário para toda possível seqüência s de n caracteres, e para cada seqüência armazenar contadores para cada caracter x que segue s. O tamanho máximo do contexto é uma constante. Probabilidades de predição para cada contexto no modelo são calculadas a partir dos contadores de freqüência. Estes últimos são atualizados adaptativamente. O símbolo que ocorre atualmente é codificado com a distribuição de probabilidades predita. A probabilidade condicional de x no contexto s, $P(x|s)$ é então estimada por $C(x|s) / C(s)$, onde $C(x|s)$ é o número de vezes que x segue s e $C(s)$ é o número de vezes que s é encontrada.

As distribuições de probabilidades são então usadas por um *Codificador Aritmético* [7] para gerar a sequência de bits. A atual implementação do PPM mantém contextos de todos os tamanhos inteiros abaixo de k, e efetivamente combina as diferentes distribuições, usando um mecanismo de *escape*. O modelo com o valor mais alto de k é, por padrão, o

primeiro usado para a codificação. Se um novo caracter é encontrado no contexto, significa que o mesmo não pode ser usado para a codificação do caracter, então um símbolo de *escape* é transmitido como sinal para a saída, situação denominada evento de *escape*. Sendo assim, o algoritmo continua sua busca no contexto para o próximo valor inferior de k . Este processo é repetido para tamanhos cada vez menores de k até que se encontre o símbolo em questão. Nesse caso, o caracter é codificado com a distribuição de probabilidades daquele modelo. Na prática, uma única estrutura de dados é usada para armazenar todas as probabilidades nos diferentes modelos de contexto, como mostrado na Tabela 1 [3]. Para assegurar que o processo termina, o PPM assume um modelo abaixo do menor nível de k (normalmente -1), contendo todos os caracteres do alfabeto codificado.

Pode parecer que quanto maior o tamanho do contexto, maior a compressão. Entretanto, o espaço de armazenamento requerido pelo PPM cresce exponencialmente com k. Isto impõe um limite prático para o tamanho de k. Além do mais, quanto maior o tamanho de um dado contexto, menos freqüentemente ele ocorre. Conseqüentemente, as probabilidades derivadas dos contadores de freqüência dos caracteres não são suficientemente confiáveis devido à diluição do contexto. Na prática, tem sido observado que os modelos de ordem 5 tendem a dar os melhores resultados de compressão [11].

A idéia básica do PPM pode ser estendida a modelos baseados em palavras nos quais estas são tratadas individualmente como símbolos e as probabilidades condicionais são calculadas de maneira similar.

Tabela 1 - Modelo PPM-C depois do processamento da *string abracadabra*

Ordem k = 2					Ordem k = 1					Ordem k = 0				Ordem k = -1			
Predição			c	p	Predição			c	p	Predição		c	p	Predição		c	p
ab	→	r	2	$\frac{2}{3}$	a	→	b	2	$\frac{2}{7}$	→	a	5	$\frac{5}{16}$	→	A	1	$\frac{1}{\|A\|}$
	→	Esc	1	$\frac{1}{3}$		→	c	1	$\frac{1}{7}$	→	b	2	$\frac{2}{16}$				
						→	d	1	$\frac{1}{7}$	→	c	1	$\frac{1}{16}$				
ac	→	a	1	$\frac{1}{2}$		→	Esc	3	$\frac{3}{7}$	→	d	1	$\frac{1}{16}$				
	→	Esc	1	$\frac{1}{2}$						→	r	2	$\frac{2}{16}$				
					b	→	r	2	$\frac{2}{3}$	→	Esc	5	$\frac{5}{16}$				
ad	→	a	1	$\frac{1}{2}$		→	Esc	1	$\frac{1}{3}$								
	→	Esc	1	$\frac{1}{2}$													
					c	→	a	1	$\frac{1}{2}$								
br	→	a	2	$\frac{2}{3}$		→	Esc	1	$\frac{1}{2}$								
	→	Esc	1	$\frac{1}{3}$													
					d	→	a	1	$\frac{1}{2}$								
ca	→	d	1	$\frac{1}{2}$		→	Esc	1	$\frac{1}{2}$								
	→	Esc	1	$\frac{1}{2}$													
					r	→	a	2	$\frac{1}{3}$								
da	→	b	1	$\frac{1}{2}$		→	Esc	1	$\frac{1}{3}$								
	→	Esc	1	$\frac{1}{2}$													
ra	→	c	1	$\frac{1}{2}$													
	→	Esc	1	$\frac{1}{2}$													

4. A Metodologia Empregada

Devido à capacidade de construir modelos precisos, algoritmos modernos de compressão sem perdas podem ser utilizados como classificadores baseados em modelos. Qualquer compressor poderia ter sido utilizado, todavia, o PPM apresenta-se mais adequado porque sua forma de modelagem por predição de *n-grama* ajusta-se perfeitamente para a predição de letras.

A implementação proposta por Moffat , a variante PPM-C, foi a escolhida devido à

sua performance [7]. Uma vez que o resultado da compressão não seria utilizado, o PPM-C foi ligeiramente adaptado, de modo a apenas construir o modelo para estimar a compressão, ignorando a codificação.

Para os testes propriamente ditos, determinam-se os parâmetros que serão utilizados para todo o processo, a ordem de Markov a ser empregada, o tamanho de cada texto usado para a construção do modelo e o tamanho de cada texto a ser usado para classificação. O processo em si é composto de três etapas abaixo descritas.

4.1. Pré-Processamento dos Textos

A título de uniformização, todos os textos que serão utilizados nas fases de treinamento e classificação, são preparados previamente de modo que sejam livres de formatação, estejam limitados a um mesmo alfabeto de símbolos, podendo ocorrer supressão de caracteres (ie. &, numerais romanos) ou redução de caracteres (ie. á = a); e de modo que seqüências de espaços vazios sejam eliminadas. O pré-processamento utilizado limita o alfabeto de símbolos ao conjunto de caracteres da língua portuguesa (caracteres latinos, acentuados, maiúsculos e minúsculos, e espaço) e um pequeno conjunto de sinais de pontuação (,;:.!?). Dessa forma, os textos apresentam-se compactos e isomórficos entre si.

4.2. Fase de Treinamento

Nesta etapa, o número N de classes (autores) C_i, onde $i = 1, 2, 3, ... N,$ é definido. No nosso experimento, cada classe C_i possui quatro textos (obras de cada autor). Um conjunto de treinamento T_i formado por grupos de três textos de cada classe C_i, é selecionado. O quarto texto de cada classe fica separado, isto é, não participa do treinamento, para uso na posterior fase de classificação. O algoritmo PPM-C então constrói adaptativamente cada modelo M_i a partir dos três textos pré-processados que constituem o conjunto T_i. Esse processo é repetido de modo a explorar todas as combinações possíveis de três textos em cada classe C_i (um total de quatro combinações, já que cada classe C_i possui quatro textos).

4.3. Fase de Classificação

Nesta etapa, o quarto texto de cada classe C_i (os textos que ficaram fora do treinamento) são comprimidos estaticamente usando PPM-C. Após a compressão, calcula-se a razão de compressão RC_i que o texto obteve com o modelo M_i. O texto é então atribuído à classe C_i cujo modelo apresente a maior razão de compressão RC_i.

5. Resultados

Dez autores foram selecionados e constituíram doze classes. Como já foi mencionado, para cada classe foram escolhidas quatro obras, sendo que três delas viriam a constituir o modelo e a quarta seria usada para classificação. Houve um revezamento das obras de modo que todas tiveram a chance de serem usadas para classificação e treinamento, perfazendo quatro combinações possíveis. A Tabela 1 relaciona os autores escolhidos e as obras utilizadas.

Tabela 2 - Autores e Obras empregados nos testes

Autores	**Obras**
Adolfo Caminha (AC)	A Normalista; Bom-Crioulo; No País dos Ianques; Tentação;
Aluísio de Azevedo (AA)	A Mortalha de Alzira; Casa de Pensão; O Cortiço; O Mulato;

Euclides da Cunha (EC)	À Margem da História; Contrastes e Confrontes; Os Sertões; Peru versus Bolívia;
Joaquim Manuel de Macêdo (JMM)	A Luneta Mágica; A Moreninha; Memórias da Rua do Ouvidor; O Moço Loiro;
José de Alencar 1(JA-1) (Romance Regional)	O Garatuja; O Gaúcho; O Sertanejo; Til;
José de Alencar 2(JA-2) (Romance Urbano)	Diva; Encarnação; Lucíola; Senhora;
Lima Barreto (LB)	Histórias e Sonhos; Os Bruzundangas; Recordações do Escrivão Isaías Caminha; Triste Fim de Policarpo Quaresma;
Machado de Assis 1 (MA-1) (Conto)	Histórias da Meia-Noite; Histórias sem Data; Papéis Avulsos; Relíquias da Casa Velha;
Machado de Assis 2 (MA-2) (Romance)	A Mão e a Luva, Dom Casmurro, Helena, Memorial de Aires
Paulo Coelho (PC)	História para Pais, Filhos e Netos; O Alquimista; O Livro dos Sábios; Veronika Decide Morrer;
Raul Pompéia (RP)	As Jóias da Coroa; Contos; O Ateneu; Uma Tragédia no Amazonas;
Visconde de Taunay (VT)	A Retirada da Laguna; Ao Entardecer; Inocência; No Declínio;

Os testes foram realizados usando ordens de Markov 4, 5 e 6; tamanhos de arquivo de treinamento e classificação assumindo valores entre 15, 30, 45, 60, 75, 90, 105, 120 kilobytes. A divisão das classes dos autores Machado de Assis e José de Alencar em outras duas classes para cada um justifica-se pela riqueza e diferenças textuais apresentadas pelos autores. A escolha dos autores e de suas obras foi fortemente influenciada pela facilidade/dificuldade de encontrar versões eletrônicas das mesmas.

A seguir apresenta-se a Matriz de Confusão (Tabela 3) dos testes, onde verifica-se o resultado global da classificação. O número de classificações totalizam 12x4x8x8=768, cujos fatores representam o número de classes, o número de obras por classe, número de tamanhos diferentes para os arquivos de treinamento e o número de tamanhos diferentes para os arquivos de classificação respectivamente. A interpretação da matriz é feita da seguinte forma: a primeira linha identifica as classes, e a primeira coluna identifica os autores; cada célula contém o número de classificações do autor da linha *i* atribuídas à classe da coluna *j*. Por exemplo na Tabela 3, a linha JMM, coluna AC indica que 1 obra de Joaquim Manuel de Macêdo foi atribuída (erroneamente) a Adolfo Caminha. Os valores em negrito indicam os acertos.

Tabela 3 - Matriz de Confusão englobando todos os testes

	AC	AA	EC	JMM	JA-1	JA-2	LB	MA-1	MA-2	PC	RP	VT
AC	**719**	21	0	0	0	0	0	0	0	0	22	6
AA	12	**610**	0	0	0	0	0	0	0	0	146	0
EC	0	0	**768**	0	0	0	0	0	0	0	0	0
JMM	1	44	0	**369**	0	105	10	4	2	0	229	4
JA-1	4	25	0	0	**576**	0	0	0	0	0	134	29
JA-2	0	0	0	0	0	**768**	0	0	0	0	0	0
LB	0	0	0	0	0	0	**756**	0	0	0	12	0
MA-1	0	2	0	0	0	3	0	**475**	278	0	0	10
MA-2	0	0	0	0	0	10	0	308	**450**	0	0	0
PC	0	0	0	0	0	0	192	0	0	**576**	0	0
RP	4	7	0	0	0	0	5	0	0	0	**750**	2
VT	95	8	38	0	11	1	7	0	0	0	170	**438**

A taxa de acertos é medida pela CCR (*Correct Classification Rate* – Taxa de

Classificação Correta), definida como:

$$CCR = \frac{c}{t} \times 100 \qquad (5)$$

onde c é o numero de textos classificados corretamente, e t é o número de textos classificados.

Os gráficos (Figuras 1 e 2) resumem os resultados obtidos. O melhor caso foi encontrado em três ocorrências. Considerando o terno (ordem de Markov, tamanho do arquivo de treinamento, tamanho do arquivo de classificação), a maior taxa de acerto (87,5%) foi encontrada nas tuplas (6, 120, 30), (6,30,15), (6,60,15).

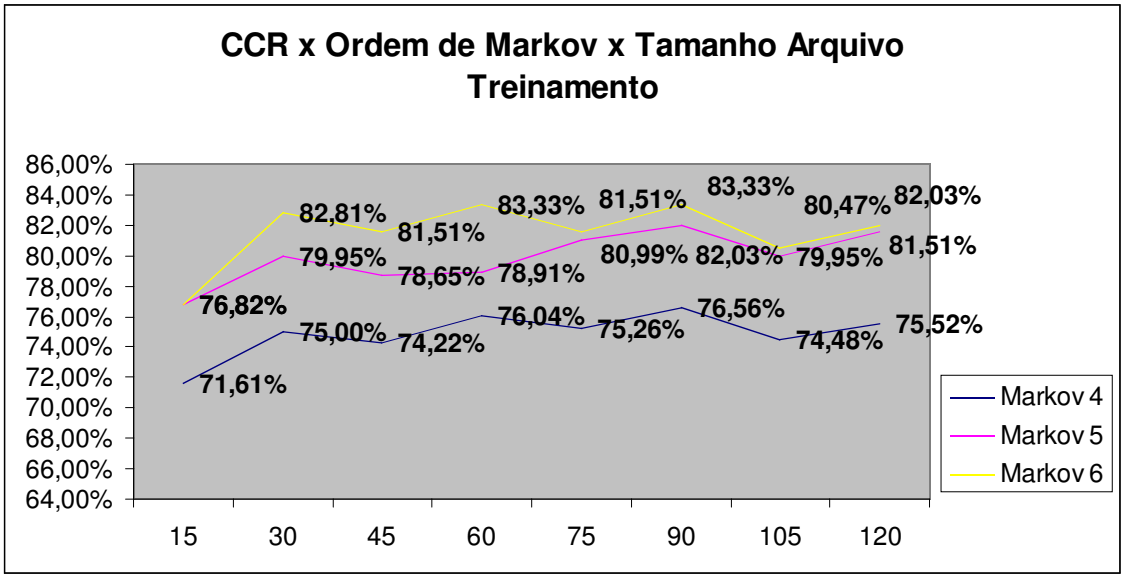

Figura 1: CCR em função do tamanho do arquivo de treinamento e ordem de Markov

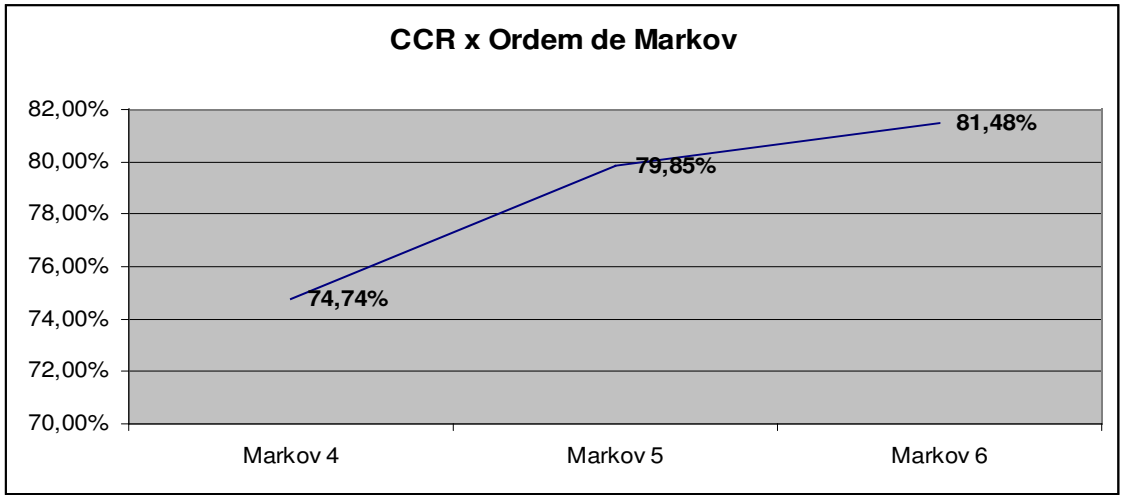

Figura 2: CCR em função da ordem de Markov

A Tabela 4 sumaria os resultados sob o ponto de vista de acertos por autor.

Tabela 4 - CCR (%) x Autor

Markov	AC	AA	EC	JMM	JA-1	JA-2	LB	MA-1	MA-2	PC	RP	VT
4	87,50	78,52	100,00	38,67	75,00	100,00	97,66	57,03	46,88	75,00	96,09	45,70
5	96,88	78,52	100,00	50,00	75,00	100,00	98,83	64,06	64,06	75,00	98,05	57,81
6	96,48	81,25	100,00	50,47	75,00	100,00	98,83	64,45	64,84	75,00	98,83	67,58
Média	93,62	79,43	100,00	48,04	75,00	100,00	98,44	61,85	58,59	75,00	97,66	57,03

Mudando a perspectiva e observando a quais autores foram atribuídas erroneamente alguma classificação, a Tabela 5 apresenta os seguintes resultados:

Tabela 5 - Atribuições Errôneas (%) x Autor

Markov	AC	AA	EC	JMM	JA-1	JA-2	LB	MA-1	MA-2	PC	RP	VT
4	3,37	2,80	0,87	0,00	0,41	3,16	3,67	6,78	5,61	0,00	12,03	0,71
5	2,14	1,22	0,41	0,00	0,10	1,68	3,82	4,69	4,64	0,00	12,24	0,61
6	0,41	1,43	0,66	0,00	0,05	1,22	3,42	4,44	4,03	0,00	12,09	1,27

6. Discussão e Conclusões

Considerando que os autores podem apresentar uma grande diversidade de escrita, ora mantendo um estilo uniforme, ora com grande variação, ora escrevendo de acordo com a escola literária, ora destoando desta, o resultado geral obtido foi satisfatório, média geral de 78%. Principalmente ao comparar com resultado obtido por Thaper[4], que atingiu taxas próximas a 78% apenas quando usou grandes arquivos de treinamento e classificação.

De modo geral, percebe-se que o aumento da ordem de Markov implica em um aumento, ainda que discreto, da eficácia do método (Figura 2). A escolha das três ordens de Markov usadas foi motivada pela relação "exploração de informação anterior" x "complexidade computacional". Ordens de Markov abaixo de 4 não exploram significativamente bem o contexto das fontes de informação, resultando em uma predição ruim. Já ordens de Markov acima de 6 demandam muito recurso computacional devido a complexidade e tamanhos dos modelos criados. Ademais, como se pode ver nos resultados, o aumento da ordem (5 para a 6, por exemplo) não traz um aumento significativo dos resultados em relação à demanda computacional exigida. Teahan também verificou que a ordem 5 tende a dar o melhor resultado em termos de compressão [11].

Nota-se ainda que o aumento do tamanho do arquivo de treinamento, a partir de 30KB, incorre em oscilações no resultado (Figura 1). Uma possível explicação é o fato de que os textos não foram testados em sua íntegra, apenas a parte inicial. Cada autor, dentro do seu estilo, demora mais ou menos a desenvolver a trama da obra. Com isso, pode ser que para algumas obras, o início não permita a construção de um modelo preciso. Isso também pode ser verificado pela diversidade do tamanho das obras completas. Algumas não atingem 150KB, outras ultrapassam 1MB. Consequentemente, o truncamento realizado nas obras, a fim de padronização, permite apenas transparecer parte do estilo do autor na mesma.

Neste experimento, os melhores resultados foram obtidos considerando 90KB como tamanho dos arquivos de treinamento nas três ordens de Markov. As taxas de acertos para esse melhor caso foram de 76,56%, 82,03% e 83,33% respectivamente para ordens de Markov 4, 5 e 6.

Alguns autores se destacaram, ora pela singularidade de seu estilo, ora pela riqueza dele. Foi o caso de José de Alencar 2 (fase urbana) e Euclides da Cunha. Os dois tiveram 100% de acerto, o que demonstra seus estilos personalizados de escrita, mesmo sendo de escolas literárias diferentes (romantismo e pré-modernismo). Outro autor que merece destaque é Raul Pompéia. Ele é quem mais confude o classificador. Suas obras causaram grande confunsão ao método. Isso demonstra uma peculiaridade intrigante do autor. Sua forma de escrever assemelha-se à de outros autores no decorrer de suas obras, como se ele mudasse várias vezes de estilo dentro de um mesmo texto.

Embora não ter tido um desempenho satisfatório, Joaquim Manuel de Macêdo demonstrou uma característica importante nos resultados obtidos. Junto com Paulo Coelho, não atraiu erroneamente nenhuma autoria. Isso mostra que a forma de escrita desses dois autores não confunde o classificador, ou seja, possuem particularidades que os outros autores não têm, embora possuam algumas de suas características.

O modelo de Paulo Coelho teve uma taxa de classificação correta de 75%. Observamos que a única obra do autor que não foi classificada corretamente foi "O livro dos Sábios". Sua classificação obteve 100% de erros, provocando perdas no total de classificações dos modelos do autor (25% do total). Um estudo desse fato se fez necessário. Como muitas obras foram retiradas da Internet, acreditávamos que a referida obra pertencia a ele, por assim estar classificada nas fontes de pesquisa. Depois de algumas consultas na lista oficial de obras de Paulo Coelho, verificamos que "O livro dos Sábios" não pertence a ele. Reagindo como esperado, o classificador não atribuiu a autoria da obra, nenhuma das vezes a Paulo Coelho, o que justifica a queda de desempenho dos seus resultados. Embora esse deslize tenha influenciado negativamente o resultado geral, reforça positivamente a eficácia do classificador, quando não classifica como própria uma obra que não pertence ao modelo.

As duas classes de José de Alencar não confundiram-se entre si, o que demonstra diferenças importantes entre as duas épocas do autor. O mesmo não acontece com Machado de Assis. Sabe-se que as obras do autor oscilam entre o Romantismo e o Realismo, e os conjuntos de testes, primaram pelo tipo de obra "romance/conto" devido a maior facilidade de obtenção de suas versões eletrônicas. Isso explica a grande confusão feita pelo classificador, quando mostrou-se incapaz, em alguns momentos, de distinguir o "Machado de Assis Conto" do "Machado de Assis Romancista" e quando confundiu o "Machado de Assis Romântico" com o "Machado de Assis Realista".

Testes foram realizados para estudar a influência de resultados específicos de cada autor. Um dos fatos mais marcantes diz respeito a Machado de Assis. Considerando as três ordens de Markov, se o classificador considerasse correta a atribuição de obras de "Machado de Assis Conto" ao "Machado de Assis Romancista", teria-se um aumento de 278 acertos de classificação, ou seja, 58,52% do total original de acertos (475) para o "Machado de Assis Conto" e 36,20% do total de classificação para o autor. Da mesma forma, se o classificador considerasse correta a atribuição de obras de "Machado de Assis Romancista" ao "Machado de Assis Conto", encontraria-se um aumento de 308 acertos, ou seja, 68,44% do total original de acertos (450) para o "Machado de Assis Romancista" e 40,10% do total de classificação para o autor. Considerando a taxa geral de acerto do classificador, 78,72%, o aumento da eficácia do classificador é significativo quando considera-se correta a atribuição de "Machado de Assis Conto" ao "Machado de Assis Romancista", 82,06%, e quando considera-se correta a atribuição de "Machado de Assis Romancista" ao "Machado de Assis Conto", 81,74%. Mais significativo ainda é o aumento da taxa geral de acerto quando consideram-se corretas as atribuições entre os "Machado de Assis": 85,08%.

O PPM mostrou-se excepcional na classificação de textos, não apenas para atribuição de autoria, como também para análise da diferenciação estilística para um mesmo autor (romance urbano x romance regional), entre os autores (estilos de escrita), e para diferenciação temporal (autores modernos x autores de outros séculos). Indicações de pesquisas futuras incluem, treinamento com arquivos de tamanhos distintos, determinação da melhor parte do texto a ser usada para treinamento e classificação, além de uso de outros classificadores para fins de comparação de eficácia.

7. Bibliografia

[1]KJELL, Bradley. *Autorship Attribution of Text Samples using Neural Networks and Bayesian Classifiers*. IEEE: International Conference on Systems Man and Cibernetic, San Antonio, USA; pp: 1660-1664; ISBN 0-7803-2129-4;1994.

[2]MCCOMBE, Niamh. *Methods of Author Identification*. Ireland, 2002.

[3]TEAHAN, W.J. *Text classification and segmentation using minimum cross-entropy*. Proceeding of RIAO'00, 6th International Conference "Recherche d'Information Assistee par Ordinateur", Paris, France, 2000.

[4]THAPER, Nitin. *Using Compression For Source Based Classification Of Text.* MIT, 2001.

[5]FRANK, E., CHI, C., WITTEN, I. H. *Text Categorization Using Compresssion Models*. Proceedings of the Data Compression Conference, Salt Lake City, pp. 500, 2000.

[6]TEAHAN, W. J., HARPER, D. J. *Using Compression Base Language Models for Text Categorization.* Workshop on Language Modeling and Information Retrieval, pp. 83-77, 2001.

[7]MOFFAT, Alistair. *Implementing the PPM Data Compression Scheme.* IEEE Transactions on Communications, 38(11):1917-1921, 1990.

[8]SHANNON, C. E. "*A Mathematical Theory of Communication*" Bell Syst. Tech. J., vol. 27, pp. 379-423, (1948).

[9]BELL, T. C., CLEARY, J. G., WITTEN, I. H., *Text Compression*, Prentice-Hall, Englewood Cliffs, 1990.

[10]CLEARY, J. G., WITTEN, I. H. *Data compression using adaptive coding and partial string matching*. IEEE Transactions on Communications, 32(4):396–402.

[11]TEAHAN, W. J. *Modelling english text*. DPhil thesis, Univ. of Waikato, N.Z., 1997.

[12]BLELLOCH, Guy E. *Introduction to data compression*. Course notes for: .Algorithms for the real world, 2000.

[13]BATISTA, L. V., MEIRA, Moab Mariz; *Texture Classification Using the Lempel-Ziv-Welch Algorithm*. Lecture Notes in Computer Science. Berlin, v.3171, p.444 - 453, 2004.

[14]BATISTA, L. V., MEIRA, Moab Mariz. *Texture Classification using Histogram Equalization and the Lempel-Ziv-Welch Algorithm.* Anais do XXI Simpósio Brasileiro de Telecomunicações - SBT'2004, v.1. p.1-6, Belém, 2004.