

Trabalho de Formatura Supervisionado

Aplicação de análise morfológica para segmentação de páginas em imagens de documentos

Ricardo de Cillo

Supervisora: Nina S. T. Hirata

Departamento de Ciência da Computação
Instituto de Matemática e Estatística, IME-USP

Resumo: Neste texto apresentamos nosso estudo sobre a aplicação de operadores morfológicos à segmentação de páginas de documentos, etapa importante na análise de documentos que busca extrair informações sobre a sua estrutura: regiões com títulos, legendas, figuras e blocos de texto.

São Paulo, 11 de fevereiro de 2013

Sumário

1	Introdução	1
2	Fundamentos	3
2.1	Imagens digitais	3
2.2	Operadores de imagens	3
2.2.1	Operadores morfológicos binário	3
2.2.2	Operadores localmente definidos e invariantes por translação	3
2.3	Classificação de objetos	4
2.4	Segmentação de imagens	4
2.5	Classificação dos componentes	4
3	Operadores morfológicos automaticamente gerados	6
3.1	Formalização	6
3.2	Coleta	6
3.3	Decisão	7
3.4	Minimização	7
4	Metodologia	8
4.1	Preparação das imagens de treinamento	8
4.2	Construção dos operadores	8
4.3	Aplicação dos operadores	8
4.4	Consensualização	9
5	Experimentos	10
5.1	Base de dados	10
5.2	Mistura de publicações no conjunto de treinamento	10
5.3	Tipos de regiões	11
5.4	Quantidade de imagens de treinamento	11
5.5	Formatos de janelas	12
5.6	Tamanho da janela de consensualização	12
5.7	Aplicação	12
5.8	Avaliação da segmentação	12
6	Resultados	13
6.1	Valores observados	14
6.2	Imagens finais	29
7	Conclusão	35
8	Apêndice	35
8.1	Algoritmo de Otsu para binarização	35
8.2	Implementação	36

1 Introdução

Processamento e análise de documentos é uma importante subárea da área de reconhecimento de padrões cujo principal objetivo é a interpretação de um documento, ou seja, o entendimento da sua estrutura bem como o reconhecimento de cada um dos componentes estruturais.

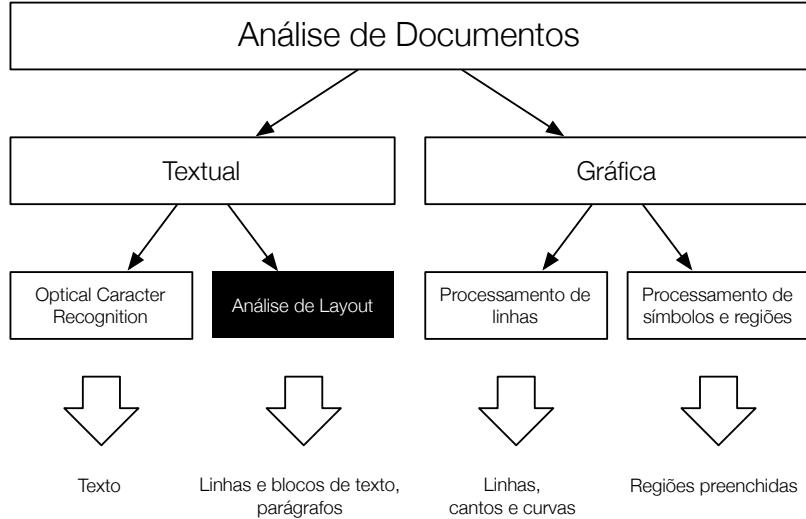


Figura 1: Contextualização do tema do trabalho entre as áreas da análise de documentos. Adaptado de [1].

Segmentação de página refere-se à tarefa de separar e rotular os diferentes componentes que fazem parte da estrutura das páginas de um documento, tais como: blocos de texto, gráficos, figuras, títulos, legendas, separadores, tabelas, fórmulas matemáticas e regiões com ruído.

Em geral, a segmentação de página é um dos primeiros passos no processo de entendimento de um documento. Uma vez identificados os blocos estruturais, processamentos específicos para cada tipo de bloco podem ser aplicados. Por exemplo, no caso de blocos de textos é conveniente fazer o reconhecimento de texto para que o mesmo possa ser armazenado em formato texto (e não imagem). Por outro lado, no caso de imagens, pode ser interessante armazená-las em alta resolução para manter a qualidade. Documentos digitalizados podem ser processados eficientemente em processos que envolvem armazenamento, edição, transmissão, ou busca, por exemplo.

Devido à grande quantidade de documentos, é interessante que o seu processamento seja realizado de forma automatizada ou pelo menos semi-automatizada. Para tal, diversas soluções computacionais vêm sendo propostas para o problema ao longo dos anos desde o surgimento desse campo de pesquisa. Automatizar esta tarefa reduz custos, aumenta a velocidade e capacidade de processamento de documentos além de possivelmente reduzir a taxa de erro humano na classificação de uma região.

Neste trabalho exploraremos a aplicabilidade de operadores morfológicos automaticamente gerados ao problema de segmentação de páginas.

Este texto está organizado da seguinte forma. Na seção 2, apresentamos as definições e conceitos básicos que serão importantes para a leitura deste texto. Na seção 3 explicamos com maior profundidade a principal ferramenta utilizada para realizar a segmentação: operadores morfológicos automaticamente gerados. Na seção 4 apresentamos como utilizamos todos os

conceitos para resolver o problema. Na seções 5 e 6 detalhamos os experimentos e apresentamos os resultados. Finalmente na seção 7 fazemos a conclusão do trabalho.

2 Fundamentos

2.1 Imagens digitais

Uma imagem digital monocromática pode ser definida como uma função $f : E \subset \mathbb{Z}^2 \rightarrow K = \{0, 1, \dots, k-1\}$, na qual k representa o número de tons de cinza. Tipicamente adota-se $k = 256$, ou seja, 8-bits de cor. Quando $k = 1$ as imagens são denominadas **binárias**; quando $k > 1$ as imagens são denominadas **tons de cinza**. Na prática, o domínio E é um retângulo finito de dimensões $m \times n$ (uma matriz de m linhas e n colunas).

Uma imagem RGB (colorida) é uma função $f : E \rightarrow K^3$, onde cada componente K representa a intensidade das cores vermelho, verde e azul, respectivamente.

2.2 Operadores de imagens

Um operador de imagens é uma função que mapeia imagens em imagens. Denotando $E = \mathbb{Z}^2$, $K = \{0, 1, \dots, k-1\}$ e todas as imagens definidas em E por K^E , podemos representar um operador de imagens como $\Psi : K^E \rightarrow K^E$.

2.2.1 Operadores morfológicos binário

Uma função $f \in \{0, 1\}^E$ pode ser vista como indicadora de um conjunto $S_f \subseteq E$, ou seja, para todo $x \in E$, $x \in S_f \Leftrightarrow f(x) = 1$. Desta forma, uma função binária pode ser vista como um conjunto. Uma função entre imagens pode ser vista como uma função entre conjuntos. Conjuntos adicionados da relação de inclusão constituem algebricamente um reticulado Booleano. Operadores morfológicos binários são operadores vistos como mapeamentos entre reticulados booleanos.

2.2.2 Operadores localmente definidos e invariantes por translação

Duas classes de operadores são importantes para o entendimento de operadores automaticamente gerados: localmente definidos e invariantes por translação.

Sejam x e y pertencentes a E , $x+y$ denota a soma vetorial em E . Dado um conjunto $X \subseteq E$, $X_z = \{x+z : x \in X\}$ denota a translação de X por z .

Seja $W \subseteq E$, chamado de janela ou elemento estruturante, um operador Ψ é dito localmente definido em relação a W , para todo $X \subseteq E$, se,

$$[\Psi(X)](x) = [\Psi(X \cap W_x)](x). \quad (1)$$

Operadores localmente definidos podem ser caracterizados por funções locais [2]. Ou seja, independentemente do tamanho do conjunto X , $\Psi(X)$ pode ser visto como uma função booleana $\psi_x : \{0, 1\}^n \rightarrow \{0, 1\}$, onde $n = |W|$, da seguinte forma:

$$[\Psi(X)](x) = \psi_x(X_{-x} \cap W), \quad (2)$$

onde $X_{-x} \cap W$ denota a atribuição de valores de $x \in X$ às variáveis x_i da função booleana, de acordo com a regra $x_i = 1 \Leftrightarrow w_i \in X_{-z} \cap W$, ou equivalentemente, se $w_i + z \in X$.

Se Ψ também for invariante por translação então teremos $\psi_x = \psi_y$ para $x, y \in E$. Operadores localmente definidos e invariantes por translação são chamados de W-operadores, operadores de janela.

Alguns exemplos de operadores desta classe são dilatação (3) e erosão (4), ditos operadores elementares.

$$\delta_B(X) = \{x \in E : B_x \cap X \neq \emptyset\} \quad (3)$$

$$\varepsilon_B(X) = \{x \in E : B_x \subseteq E\} \quad (4)$$

2.3 Classificação de objetos

Na área de reconhecimento de padrões e aprendizado computacional estudam-se métodos e técnicas para classificação de dados em geral. Os dados (padrões) a serem classificados correspondem, em geral, à representação digital de algum objeto concreto ou abstrato. O objetivo da classificação é atribuir um rótulo de classe a cada padrão observado.

Dependendo do problema, os rótulos de classe podem ser conhecidos ou não. Por exemplo, se desejamos fazer o reconhecimento de caracteres, os padrões são a imagem dos caracteres e os rótulos de classe são as identificações dos possíveis caracteres. Por outro lado, em problemas como na classificação de perfil de consumidores, pode não haver um conjunto de perfis pré-estabelecidos e o objetivo seria então identificar a possível existência de perfis. O primeiro é conhecido como problema de classificação supervisionada e o segundo como classificação não-supervisionada.

No caso da classificação supervisionada, supõe-se que os padrões são elementos de um espaço X e que o conjunto de rótulo de classe é dado por $Y = \{y_1, y_2, \dots, y_c\}$. Assim, um classificador pode ser expresso por uma função $f : X \rightarrow Y$.

Frequentemente X é um subespaço de \mathbb{R}^d . Assim, um padrão é representado por uma d -upla $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$.

2.4 Segmentação de imagens

A segmentação de imagens é um processamento comum a praticamente todos os processos que envolvem análise de imagens. Segmentar uma imagem corresponde a particionar o seu domínio, de forma que cada região resultante corresponda (do ponto de vista semântico) a uma componente de interesse na análise em questão. Este problema pode ser modelado como uma classificação de objetos, onde o conjunto de pixels de uma imagem são os objetos em X e as componentes em Y são regiões de interesse.

2.5 Classificação dos componentes

O problema de segmentação de imagens de documentos pode ser modelado como um problema de classificação de objetos, onde cada pixel da imagem é rotulado através de uma função classificadora

$$\Psi : \mathbb{E} \rightarrow \mathbb{Y} \quad (5)$$

sendo \mathbb{Y} um conjunto composto pelas regiões de interesse:

- blocos de texto: região com parágrafos
- gráficos
- figuras
- títulos
- legendas

- separadores
- tabelas
- fórmulas matemáticas
- regiões com ruído

Neste trabalho utilizaremos operadores morfológicos binários como classificadores de imagens. O processo todo será descrito na seção 4.

3 Operadores morfológicos automaticamente gerados

Construir operadores morfológicos que resolvam problemas complexos como o de segmentar uma página de documento, pode ser uma tarefa que demande muito tempo, experiência e conhecimento específico do assunto. Como estas imagens possuem características distintas dependendo da publicação (diferentes jornais e revistas), é possível que apenas um operador não consiga ser aplicado a todas as imagens. Ou seja, construir operadores com facilidade é um fator sensível para a viabilização desta abordagem.

Nesta seção apresentamos um método para projetar operadores morfológicos de forma automática utilizando técnicas de aprendizado computacional.

O método proposto na tese de mestrado de Nina S. T. Hirata [2] requer que um conjunto de pares de imagens sejam fornecidos para uma etapa inicial de treinamento. Estes pares contém uma imagem **original** e sua **ideal**, ou seja, a mesma imagem porém modificada como desejamos que o método aprenda a reproduzir em outras imagens.

A partir deste conjunto de treinamento um operador é gerado, o qual podemos aplicar a outras imagens.

3.1 Formalização

Assumimos que as imagens originais e ideias são realizações dos conjuntos aleatórios S e I , com distribuição de probabilidade conjunta $P(S, I)$. Supõe-se que esta função seja caracterizada por um processo local $(S \cap W_z, I(z))$, com $S \cap W_z \in P(W_z)$, conjunto potência de W_z , e $I(z) \in \{0, 1\}$. Ou seja, as imagens de exemplo passam a ser vistas como realizações de conjuntos aleatórios localmente definidos pela janela W . Denotaremos este processo por (X, y) .

Desta forma, o trabalho de construção de um operador l.d. e i.t. ψ que se aproxime do operador ideal Ψ , passa a ser o de minizar o erro absoluto médio (MAE), dado pela seguinte expressão:

$$MAE(\Psi) = E[|\psi(X) - y|]. \quad (6)$$

Desenvolvendo a expressão acima, chegamos ao critério para decidir o valor de $\psi(X)$ para cada X .

$$\begin{aligned} MAE(\Psi) &= \sum_{(X,y)} |\psi(X) - y| P(X, y) \\ &= \sum_{(X,0)} \psi(X) P(X, 0) + \sum_{(X,1)} |\psi(X) - 1| P(X, 1) \\ &= \sum_{X: \psi(X)=1} P(X, 0) + \sum_{X: \psi(X)=0} P(X, 1) \end{aligned} \quad (7)$$

Ou seja, para minimizar o MAE basta que escolhamos de forma adequada os valores de $\psi(X)$ de acordo com estimativas das probabilidades $P(X, 0)$ e $P(X, 1)$ através da observação dos exemplos fornecidos.

A seguir detalhamos as etapas envolvidas na construção de ψ .

3.2 Coleta

Seja $C(X_W) = \{X \cap W + z: z \in E\}$ o conjunto de todas as configurações observadas ao se transladar a janela W sobre a imagem X . Seja Y_z o valor 0 ou 1 encontrado na imagem Y na posição z . Construímos uma tabela com três colunas: $C(X_W)$, frequência observada $Y_z = 0$ e frequência $Y_z = 1$.

O resultado desta etapa é uma estimativa $\hat{P}(X, y)$ de $P(X, y)$.

3.3 Decisão

As configurações observadas na etapa anterior possuirão valores relacionados em Y iguais a 0, 1 ou ambos. No caso de tanto 1 como 0 terem sido observados, escolheremos como valor para a função final a com maior número de ocorrências. No caso de alguma configuração não ter sido observada ou de o número de ocorrências empatar, escolheremos o valor que simplifica a próxima etapa: minimização.

Formalmente, se $\hat{P}(X, 1) > \hat{P}(X, 0)$, então $\psi(X) = 1$, do contrário $\psi(X) = 0$.

3.4 Minimização

Na etapa anterior é usual que nem todos os padrões possíveis para X sejam observados. Logo precisamos completar a definição da função $\psi(X)$. Para tanto utilizamos um minimizador de funções booleanas chamado ISI. Ele não só completa a função como também produz uma representação mais compacta utilizando intervalos maximais.

Um algoritmo bastante conhecido para realizar a minimização de funções booleanas é o algoritmo de QM (Quine-McCluskey), onde através de combinações dois a dois dos mintermos de uma expressão booleana procura-se criar cubos cada vez maiores, eliminando depois os cubos sobressalentes a fim de obter uma cobertura mínima da função.

Porém esta etapa de combinação dois a dois tende a ser computacionalmente ineficiente. Além de que a etapa para identificação dos n-cubos primos (aqueles que não podem ser descartados) é muito custosa computacionalmente.

Por estes motivos, a biblioteca TRIOS [3], que implementa todo o processo descrito nesta seção, utiliza um outro algoritmo denominado ISI. Este algoritmo inicia com o n-cubo, representado na forma do intervalo $[\emptyset, W]$ e a cada iteração remove um ponto do n-cubo equivalente um mintermo negativo. Desta forma quebrando o intervalo inicial em outros N intervalos. Após eliminar todos os mintermos negativos chegamos um conjunto de intervalos maximais que representam a função. Uma descrição mais detalhada do algoritmo pode ser encontrada em [2].

Esta é a última etapa do processo. Podemos agora aplicar o operador a outras imagens.

4 Metodologia

O método proposto é baseado em operadores morfológicos automaticamente gerados, ou seja, construímos um segmentador genérico a partir de alguns exemplos de segmentação (pares de imagens). A seguir detalhamos cada uma das etapas envolvidas.

1. Preparação das imagens de treinamento
2. Construção dos operadores
3. Aplicação dos operadores
4. Consensualização

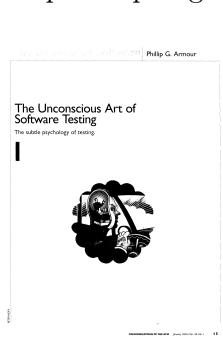
4.1 Preparação das imagens de treinamento

O primeiro passo consiste na produção dos pares de imagens de treinamento. Estes pares consistem da imagem original binarizada e de sua variante segmentada.

Para cada imagem original geramos n pares de exemplo, sendo n o número de tipos de regiões que desejamos segmentar. No caso deste trabalho nos limitamos a dois: textos de parágrafos e título.

A variante segmentada consiste da imagem original com a região de interesse apagada (em branco). A tabela 1 é um exemplo com uma imagem original, binarizada e suas variantes.

Tabela 1: Exemplo de preparação de imagem para treinamento do operador

Original	Preto e branco	Exemplo de parágrafo	Exemplo de título
			

O processo de binarização foi melhor detalhado no apêndice 8.1.

4.2 Construção dos operadores

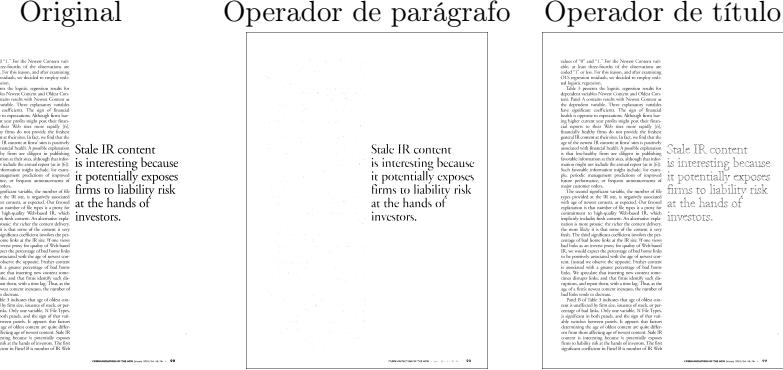
O algoritmo gerador de operadores morfológicos recebe como entrada um conjunto de pares de imagens de exemplo do passo anterior. Treinamos um operador para cada tipo de região utilizando a biblioteca TRIOS [3]. Os parâmetros para o treinamento foram ajustados experimentalmente como apresentado na seção 5.

4.3 Aplicação dos operadores

Construídos os operadores para cada tipo de região, aplicamos todos os operadores às imagens que desejamos segmentar obtendo um resultado para cada operador. Sendo n operadores e m

imagens, obtemos nm resultados. A tabela 2 demonstra a aplicação dos operadores de parágrafo e título a uma imagem.

Tabela 2: Exemplo de preparação de imagem para treinamento do operador



4.4 Consensualização

A aplicação de operadores diferentes à mesma imagem pode gerar resultados incoerentes. Pixéis classificados como pertencentes a mais de uma região fazem com que a união dos resultados seja impossível sem nenhum tipo de processamento. Para concluirmos a segmentação é necessário chegar a um consenso sobre qual operador possui a maior probabilidade de estar certo a cada pixel de classificação conflitante.

Partindo da observação de que pixel pertencente a uma certa região costuma estar cercados por pixéis da mesma região, aplicamos um processo de escolha por maior contagem de pixéis na vizinhança. Ou seja, se houver um conflito de classificação de um pixel entre texto ou título, conta-se quantos pixéis na vizinhança pertencem a uma dada região e a que obtiver maior contagem ganha.

5 Experimentos

Realizamos experimentos variando diversos parâmetros em diferentes etapas da segmentação. Apesar de usarmos uma técnica de aprendizado computacional para automatizar parte do processo, identificar quais parâmetros produzem o melhor resultado a um custo desejável (tempo de processamento e poder computacional) é uma tarefa ainda experimental.

A qualidade da solução e o tempo levado para processá-lo foram medidos a fim de identificar o lucus optimus, ou seja, a combinação que produz um bom resultado com um tempo de processamento razoável.

Os parâmetros escolhidos para análise foram:

- Mistura de publicações no conjunto de treinamento.
- Quantidade de imagens no conjunto de treinamento.
- Tipos de regiões.
- Formatos de janelas.
- Tamanhos de janelas para consensualização.

A seguir detalhamos os valores dos parâmetros e os motivos de sua escolha.

5.1 Base de dados

As imagens utilizadas nos experimentos foram obtidas de um banco de dados construído pelos pesquisadores do PRImA ao longo de anos [4]. Ele inclui um conjunto de documentos que busca simular um cenário realístico de aplicação, com layouts complexos e diferentes tipos de fontes e formatos de regiões. Isto é importante para avaliar a aplicabilidade do método em situações práticas, onde um controle sobre o formato do conteúdo seria indesejável ou inviável.

No artigo [4], os autores apresentam um conjunto de dados com páginas de revistas, artigos científicos diversos, documentos modernos e não apenas históricos.

O conjunto de dados contém não só imagens mas também arquivos XML [5] com metadados como informações bibliográficas (título, autor, publicação), informações das imagens (resolução, bit depth, modelo do scanner), características do layout (número de colunas, variedade de tamanhos de fontes) e informações administrativas (direitos autorais).

Os documentos são digitalizados com um cartão escuro por trás para minimizar a exposição da contra página. Posteriormente um algoritmo analisa possíveis falhas, como rotação do documento, marcando-os para redigitalização. Uma correção automática não é utilizada pois isto pode comprometer a qualidade da imagem.

Uma vez que a imagem foi aceita no banco de dados, inicia-se um processo manual de marcação do ground-truth. Este trabalho deve ser realizado da forma mais precisa possível pois é a base para determinar a corretude dos algoritmos segmentadores. Por se tratar de uma etapa muito custosa, uma ferramenta semi-automática chamada Aletheia [6] é utilizada para agilizar o processo. Esta ferramenta permite a uma pessoa desenhar uma região poligonal em torno de uma região de interesse. Em seguida esta região é automaticamente ajustada pelo software, como se a pessoa estivesse colocando um elástico que aperta a região.

5.2 Mistura de publicações no conjunto de treinamento

Diferentes publicações trabalham com fontes e grafismos próprios. A diferença fica evidente ao comparar um jornal antigo com uma revista sobre tecnologia. Incluir imagens de publicações

diversas no mesmo conjunto de imagens de treinamento pode produzir operadores menos precisos. Mesmo entre publicações do mesmo período, podemos ter diferentes tamanhos de fontes e famílias de tipos. Este experimento tem por objetivo analisar o impacto na qualidade por tentar criar operadores mais genéricos.

Testamos os seguintes conjuntos de imagens de treinamento:

- Communications of the ACM: 263, 674, 677, 680, 683, 685, 686, 689, 692, 695, 802 e 803. Este conjunto é chamado de CACM neste texto.
- TIME Magazine: 232, 720, 721, 723, 782, 783, 784, 785 e 786. Este conjunto é chamado de TIME neste texto.

5.3 Tipos de regiões

Os gabaritos do conjunto de dados utilizados nestes experimentos rotulam com detalhes cada região das imagens:

1. Text
 - (a) Header
 - (b) Headings (título)
 - (c) Capital (letras maiores)
 - (d) Drop Capital
 - (e) Credit
 - (f) Paragraph (parágrafo)
 - (g) Floating
 - (h) Page number
 - (i) Footer
2. Graphic
3. Math
4. Chart
5. Image
6. Noise
7. Separator
8. Table

Nos experimentos realizados nos limitamos a Text Heading e Text Paragraph por limitações de custo e tempo.

5.4 Quantidade de imagens de treinamento

A quantidade de imagens no conjunto de treinamento pode afetar a variedade e quantidade de padrões amostrados, influenciando diretamente a qualidade da solução. A sensibilidade a este fator pode depender do tipo de região a ser segmentada, caso a quantidade de amostras por imagem varie por região. O tempo para se treinar o operador também é impactado, pois deve-se coletar mais amostras.

Treinamos operadores utilizando cinco tamanhos de conjuntos (de uma a cinco imagens).

5.5 Formatos de janelas

Variando o tamanho da janela procuramos capturar padrões característicos de cada tipo de região. O tamanho da janela impacta o tempo de processamento da etapa mais custosa que é a minimização, logo descobrir um tamanho de janela com um bom custo benefício é essencial para aplicações práticas. Sabemos que nem sempre janelas maiores produzem resultados melhores, portanto este experimento procura descobrir o melhor.

Utilizamos janelas densas e esparsas, ou seja, janelas com todos os pontos preenchidos ou apenas alguns. Os tamanhos variam de 3x3 a 7x7 para densas e de 3x3 a 11x11 para as esparsas.

5.6 Tamanho da janela de consensualização

Experimentamos janelas variando de 3x3 a 15x15. Procuramos entender até quando a hipótese de que pixéis de uma determinada região não ocorrem isoladamente é verdadeira.

5.7 Aplicação

Construímos 200 operadores e os aplicamos aos nossos conjuntos de imagens de teste, cada um contendo imagens não envolvidas no treinamento dos operadores. Salvamos todas as imagens produzidas, totalizando 6.8 GB em resultados.

Após aplicarmos todos os operadores, executamos o processo de consensualização para finalmente classificarmos os pixéis como pertencentes a um parágrafo, título ou desconhecido.

5.8 Avaliação da segmentação

Os dois principais métodos utilizados para avaliar a qualidade das soluções para este problema são a comparação da classificação dos pixéis individualmente ou a classificação de regiões, como sugerido no artigo [7]. Optamos pela comparação entre pixéis pois nosso método produz como resultado a classificação de pixéis e não a delimitação de regiões. Seria necessário agrupar os pixéis em regiões para que pudéssemos realizar outros tipo de avaliação, o que foge ao escopo deste trabalho.

Uma vantagem na comparação entre pixéis e não regiões é o fato de que evitamos restrições arbitrárias na forma das regiões, porém utilizamos muito mais espaço do que o formato de regiões delimitadas por polígonos.

6 Resultados

O processo de segmentação foi avaliado após a aplicação do operador morfológico. Para cada imagem processada contamos a quantidade de classificações em cada uma das categorias de erro ou acerto da tabela 3. A partir da contagem destas classes calculamos quatro métricas diferentes tabeladas em 4. Apenas contamos quando o pixel na imagem original é ligado.

Tabela 3: Tabela com classes de resultados
Classificado como

		Classificado como	
		positivo	negativo
Gabaritado como	positivo	Verdadeiro positivo (tp). Pixel apagado corretamente.	Falso negativo (fn). Pixel deveria ter sido apagado.
	negativo	Falso positivo (fp). Pixel não deveria ter sido apagado.	Verdadeiro negativo (tn). Pixel mantido ligado corretamente.

Tabela 4: Métricas para avaliação de desempenho dos operadores

Métrica	Fórmula	Pior desempenho	Valor ótimo
Precision (Precisão)	$\frac{tp}{tp+fp}$	0	1
Recall (Sensibilidade)	$\frac{tp}{tp+fn}$	0	1
F-measure	$2 \frac{Precision \cdot Recall}{Precision + Recall}$	0	1
MCC	$\frac{tp \cdot tn - fp \cdot fn}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}}$	-1	1

- Precision (Precisão): quantidade de pixels corretamente classificados como pertencentes a uma região dividido pelo total de pixels classificados.
- Recall (Sensibilidade): quantidade de pixels corretamente classificados sobre quantidade de todos os pixels que deveria ter sido classificados.
- F-measure: média harmônica de Precision e Recall.
- MCC (Coeficiente de correlação de Mathew): Métrica bastante utilizada na avaliação de classificadores binários. Leva em consideração verdadeiros e falso positivos e negativos. O valor 0 indica que o classificador é equivalente a um classificador aleatório. Funciona bem com classes de tamanhos muito diferentes.

A métrica utilizada nas referências teóricas sobre operadores morfológicos automaticamente gerados é o MAE (sigla em inglês para Erro Absoluto Médio), cuja fórmula pode ser escrita como $\frac{|fp+fn|}{N}$, sendo N o número total de pixels na imagem. Porém esta métrica não é muito elucidadora para avaliar segmentação de páginas. Se, por exemplo, um dado operador ideal afetar 5% dos pixels de uma imagem e o operador gerado afetar 5% dos pixels da imagem não pertencentes aos 5% do ideal, o MAE seria de 5%. Ou seja, o operador errou todos os pixels que ele deveria ter modificado e ainda modificou outros indevidamente. Ela não distingue entre diferentes tipos de erro como rotulação parcial de regiões e rotulação indevida de regiões. Já o F-measure resultaria em 0 e o MCC (Coeficiente de Correlação de Matthew) poderia inclusive apresentar um valor negativo.

6.1 Valores observados

As tabelas 5, 6, 7 e 8 contém os valores de Precision, Recall, F-measure e MCC para a classificação de parágrafos do conjunto de teste CACM. As tabelas 9, 10, 11 e 12 contém os valores para classificação de parágrafos do conjunto TIME. As tabelas 13, 14, 15, 16, 17, 18, 19 e 20 contém os valores para a classificação de títulos. Cada tabela possui um gráfico correspondente com as janelas na horizontal, o valor da métrica na vertical. Cada cor de barra está relacionada a um tamanho de conjunto de treinamento.

Também apresentamos o tempo, em escala linear (figuras 2 e 3) e logarítmica (figuras 4 e 5) para a construção dos operadores de parágrafo e título.

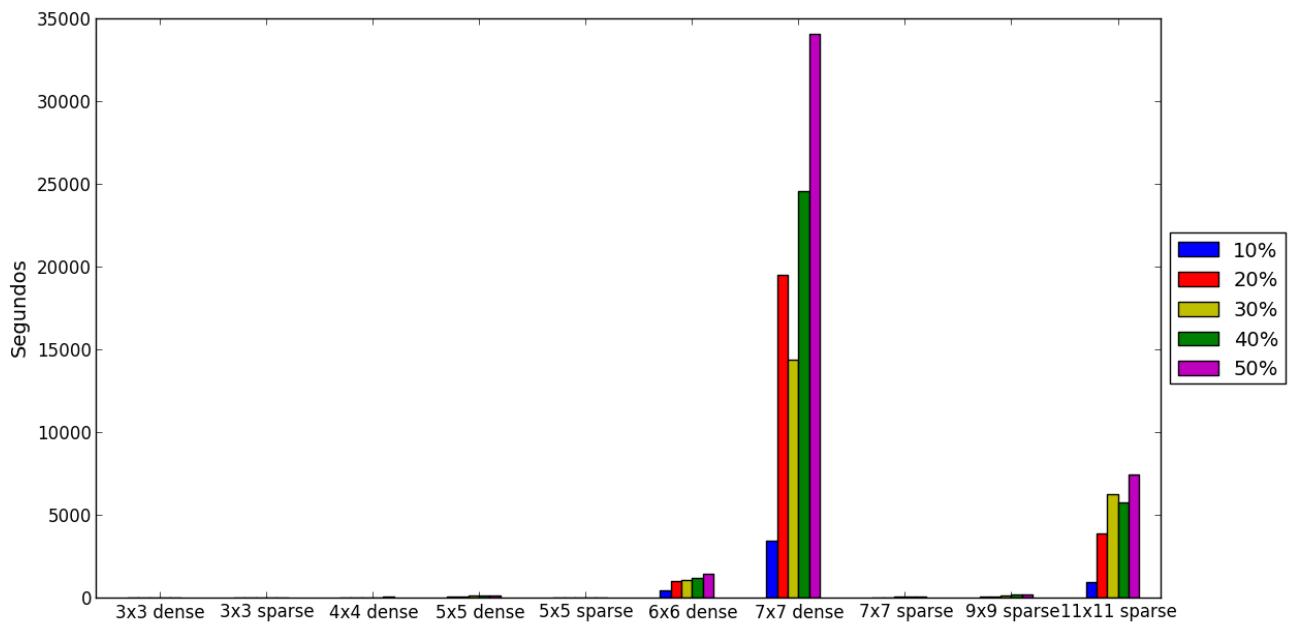


Figura 2: Tempo para treinamento dos operadores de parágrafo

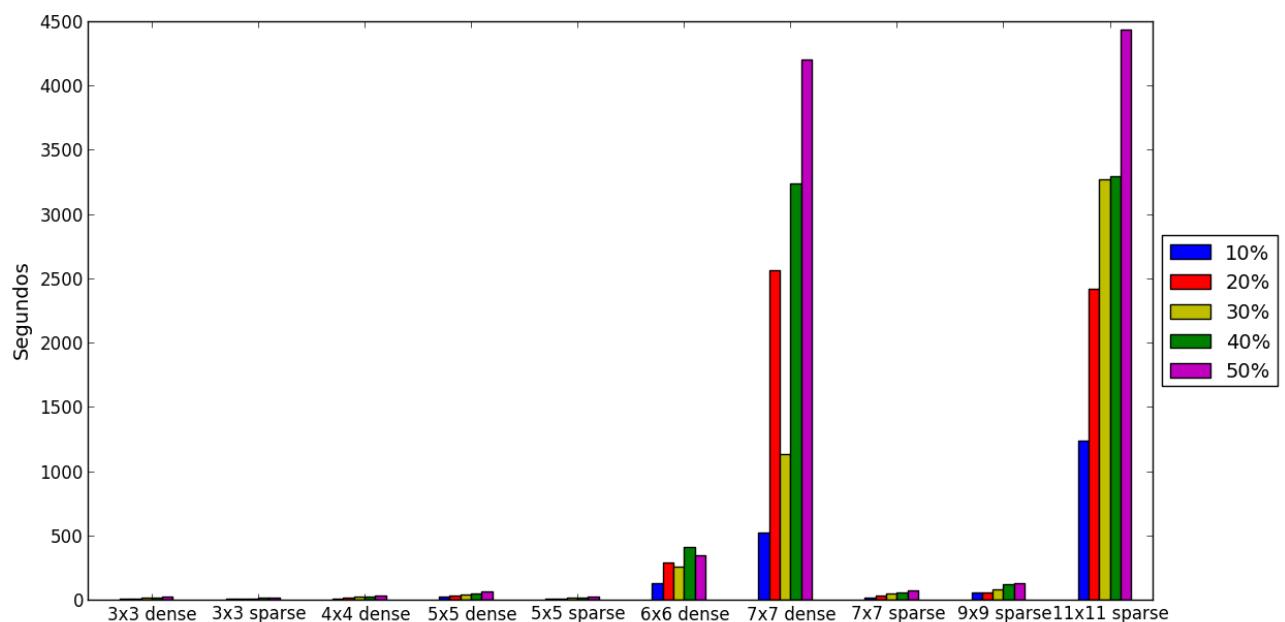


Figura 3: Tempo para treinamento dos operadores de título

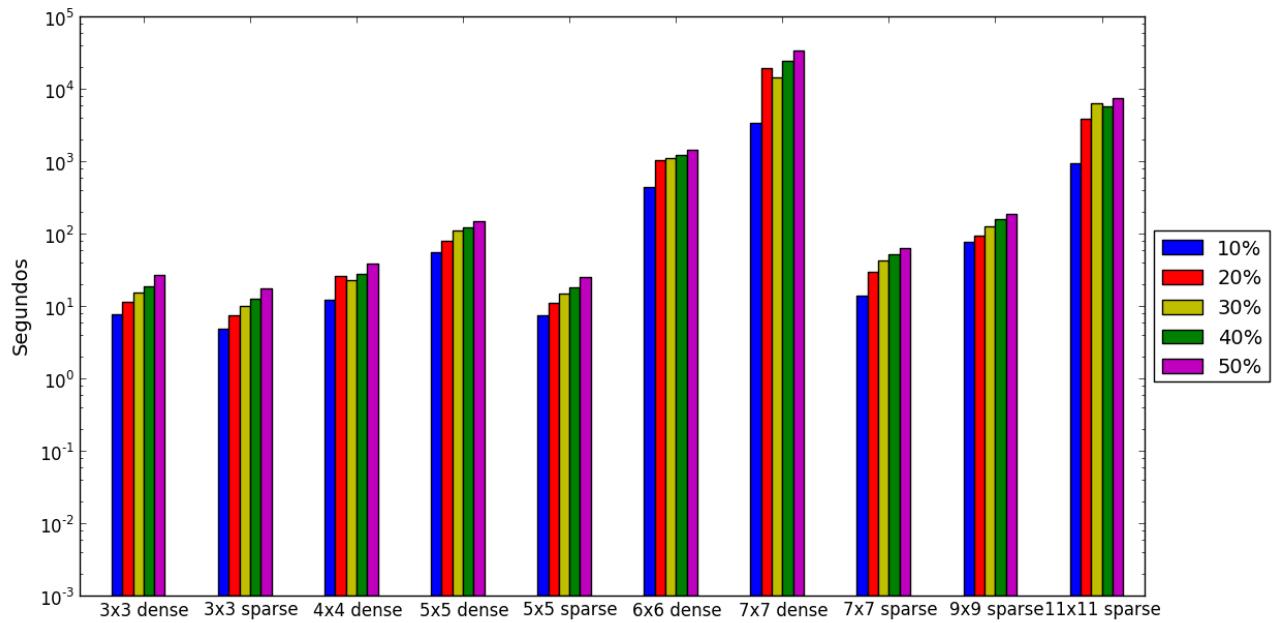


Figura 4: Tempo para treinamento dos operadores de parágrafo em escala logarítmica

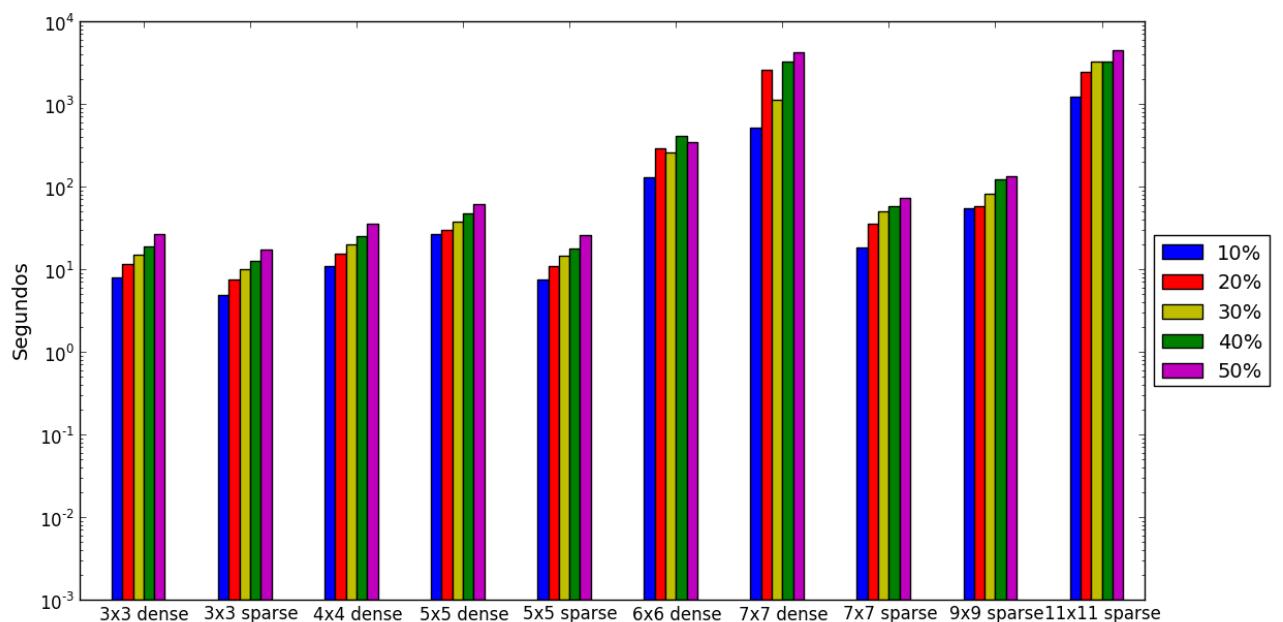


Figura 5: Tempo para treinamento dos operadores de título em escala logarítmica

Tabela 5: Média da precisão na classificação de parágrafos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.6955	0.8419	0.8850	0.9166	0.9396	0.6953	0.8718	0.9038	0.9527	0.9230
20%	0.8378	0.8557	0.8845	0.8926	0.9076	0.8376	0.8744	0.8859	0.9296	0.9029
30%	0.8383	0.8690	0.8911	0.8979	0.9141	0.8376	0.8751	0.8921	0.9345	0.9047
40%	0.8428	0.8635	0.8885	0.8943	0.9117	0.8376	0.8651	0.8820	0.9291	0.9025
50%	0.8418	0.8465	0.8826	0.8918	0.9125	0.8376	0.8601	0.8831	0.9289	0.9044

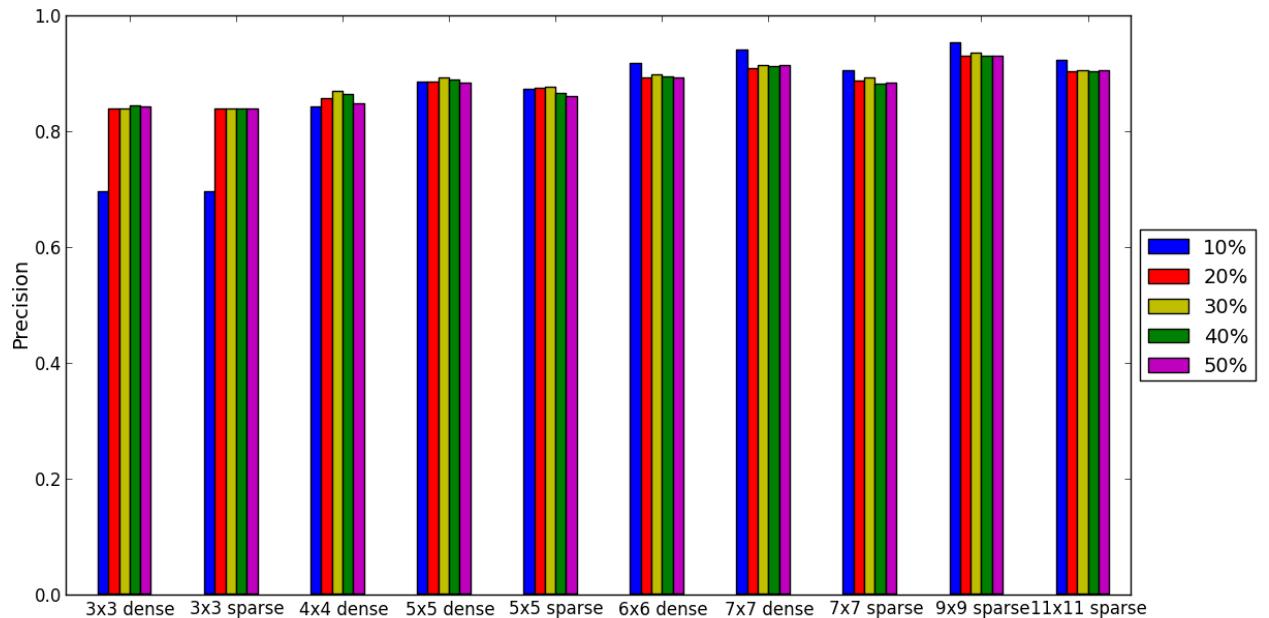


Figura 6: CACM: Classificação de parágrafos

Tabela 6: Média recall na classificação de parágrafos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.9999	0.9537	0.9817	0.9736	0.9718	1.0000	0.9906	0.9937	0.9863	0.9866
20%	0.7796	0.9461	0.9778	0.9686	0.9677	0.7781	0.9894	0.9944	0.9823	0.9829
30%	0.7795	0.9316	0.9740	0.9683	0.9668	0.7781	0.9886	0.9928	0.9829	0.9848
40%	0.7789	0.9433	0.9739	0.9643	0.9641	0.7781	0.9886	0.9928	0.9810	0.9832
50%	0.7796	0.9591	0.9879	0.9802	0.9781	0.7781	0.9939	0.9953	0.9884	0.9896

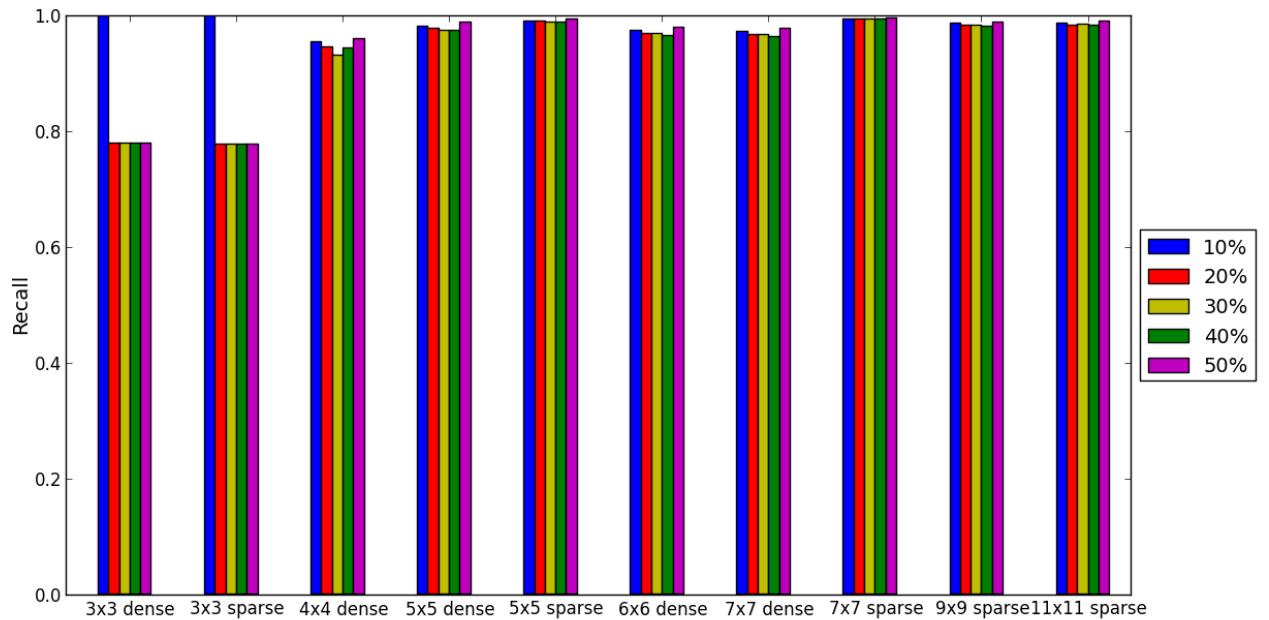


Figura 7: CACM: Classificação de parágrafos

Tabela 7: Média F1 na classificação de parágrafos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.8088	0.8932	0.9304	0.9440	0.9553	0.8087	0.9268	0.9462	0.9691	0.9534
20%	0.8067	0.8979	0.9284	0.9287	0.9365	0.8058	0.9278	0.9366	0.9550	0.9409
30%	0.8068	0.8987	0.9303	0.9315	0.9396	0.8058	0.9279	0.9394	0.9580	0.9428
40%	0.8087	0.9010	0.9288	0.9277	0.9370	0.8058	0.9221	0.9337	0.9542	0.9409
50%	0.8086	0.8984	0.9319	0.9336	0.9440	0.8058	0.9215	0.9354	0.9576	0.9448

Tabela 8: Média MCC na classificação de parágrafos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0212	0.5591	0.7194	0.7746	0.8199	0.0006	0.6999	0.7775	0.8669	0.8100
20%	0.3717	0.5825	0.7123	0.7134	0.7460	0.3691	0.7025	0.7312	0.8123	0.7560
30%	0.3726	0.5940	0.7205	0.7276	0.7621	0.3691	0.7022	0.7434	0.8261	0.7671
40%	0.3815	0.5950	0.7105	0.7087	0.7497	0.3691	0.6803	0.7187	0.8095	0.7568
50%	0.3795	0.5740	0.7246	0.7302	0.7767	0.3691	0.6810	0.7259	0.8235	0.7743

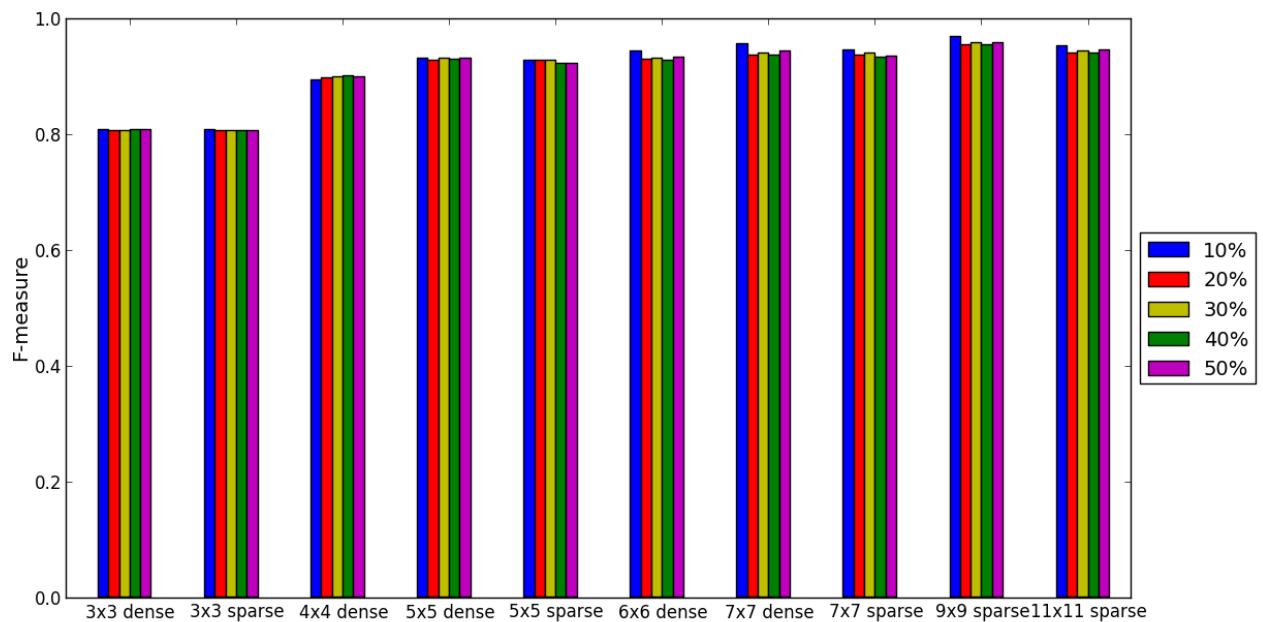


Figura 8: CACM: Classificação de parágrafos

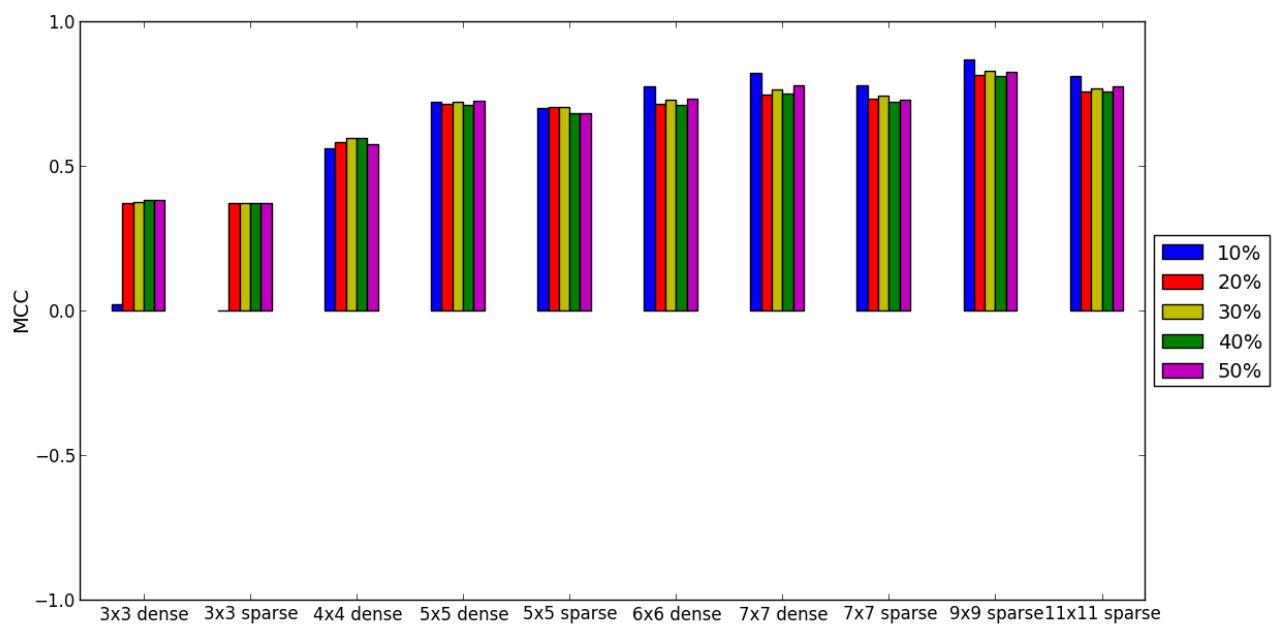


Figura 9: CACM: Classificação de parágrafos

Tabela 9: Média da precisão na classificação de parágrafos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.7725	0.8186	0.8662	0.8856	0.9093	0.7226	0.8536	0.8119	0.8968	0.8095
30%	0.7729	0.8123	0.8699	0.9007	0.9177	0.7226	0.8440	0.8188	0.9032	0.8210
40%	0.7466	0.7997	0.8655	0.9008	0.9211	0.7077	0.7858	0.8155	0.9044	0.8271
50%	0.7493	0.8139	0.8775	0.9087	0.9304	0.7226	0.8364	0.8221	0.9094	0.8272

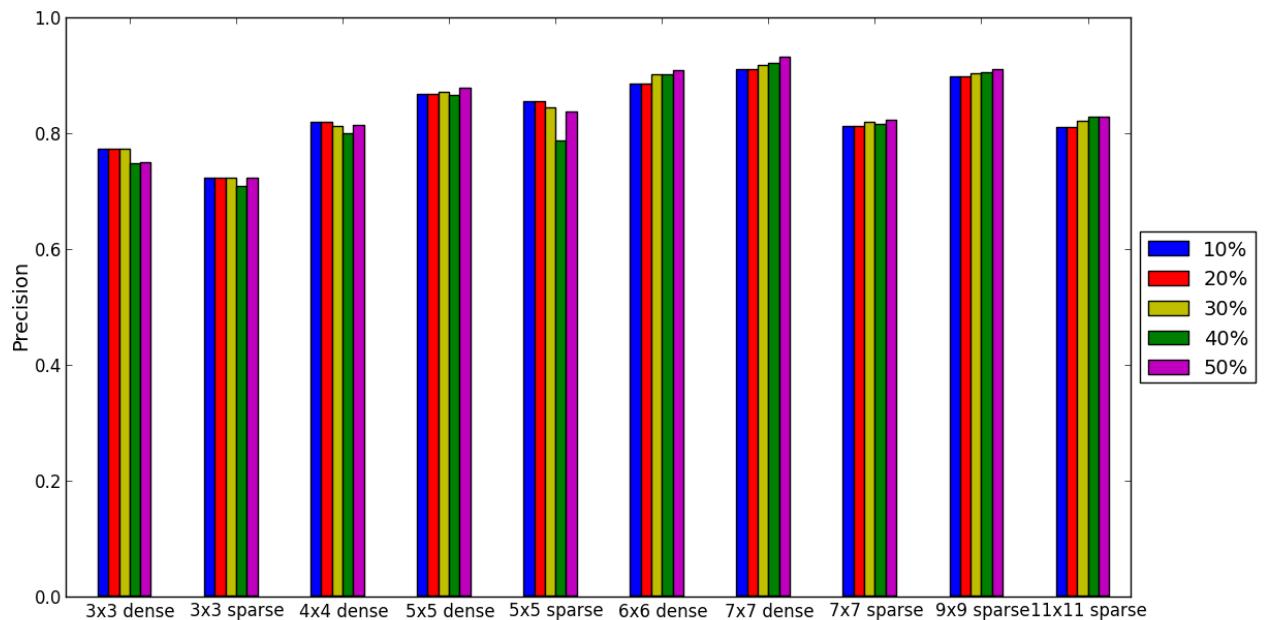


Figura 10: TIME: Classificação de parágrafos

Tabela 10: Média recall na classificação de parágrafos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.6025	0.7503	0.7864	0.7966	0.8039	0.7645	0.6514	0.9461	0.8761	0.9172
30%	0.6011	0.7712	0.7990	0.8170	0.8286	0.7645	0.6825	0.9489	0.8939	0.9263
40%	0.7304	0.8761	0.9166	0.9281	0.9269	0.8087	0.9228	0.9872	0.9640	0.9734
50%	0.7212	0.8436	0.8982	0.9177	0.9251	0.7645	0.7446	0.9790	0.9595	0.9728

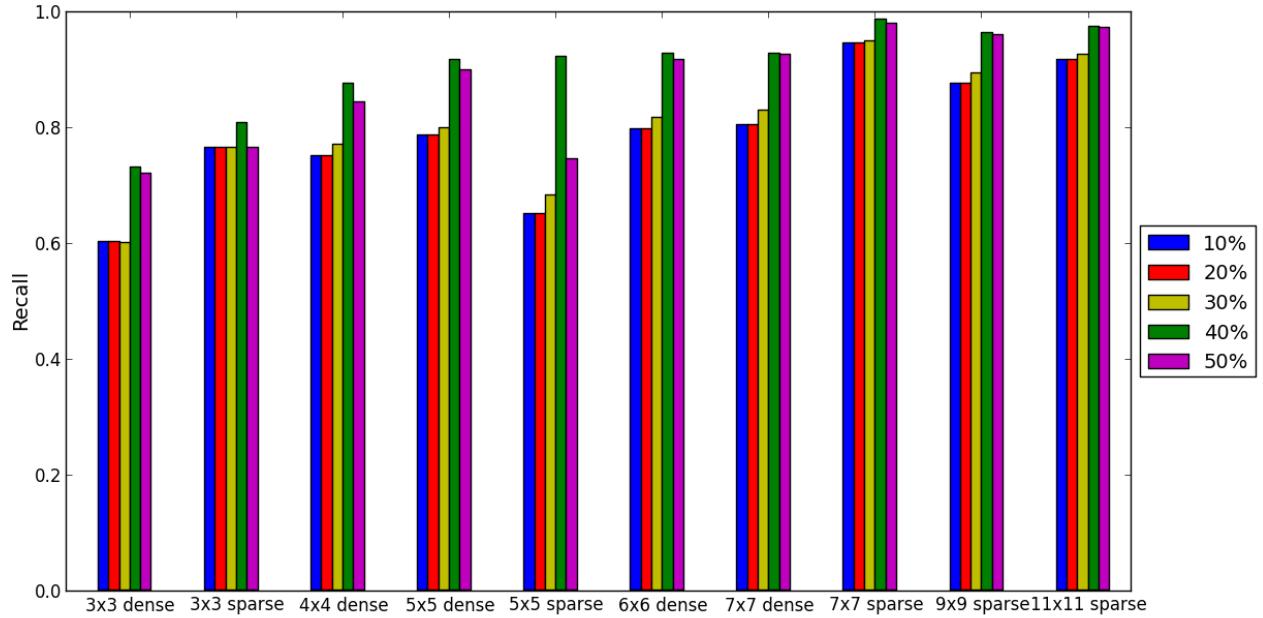


Figura 11: TIME: Classificação de parágrafos

Tabela 11: Média F1 na classificação de parágrafos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.6757	0.7819	0.8230	0.8374	0.8519	0.7418	0.7373	0.8733	0.8857	0.8594
30%	0.6750	0.7902	0.8315	0.8553	0.8694	0.7418	0.7532	0.8785	0.8980	0.8699
40%	0.7372	0.8353	0.8897	0.9138	0.9235	0.7536	0.8479	0.8927	0.9330	0.8940
50%	0.7338	0.8276	0.8868	0.9123	0.9270	0.7418	0.7869	0.8932	0.9335	0.8937

Tabela 12: Média MCC na classificação de parágrafos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.5438	0.6720	0.7350	0.7575	0.7817	0.6004	0.6376	0.8007	0.8224	0.7784
30%	0.5431	0.6801	0.7467	0.7844	0.8059	0.6004	0.6507	0.8091	0.8407	0.7958
40%	0.6010	0.7393	0.8253	0.8642	0.8799	0.6137	0.7601	0.8352	0.8939	0.8362
50%	0.5977	0.7310	0.8227	0.8629	0.8863	0.6004	0.6854	0.8350	0.8947	0.8363

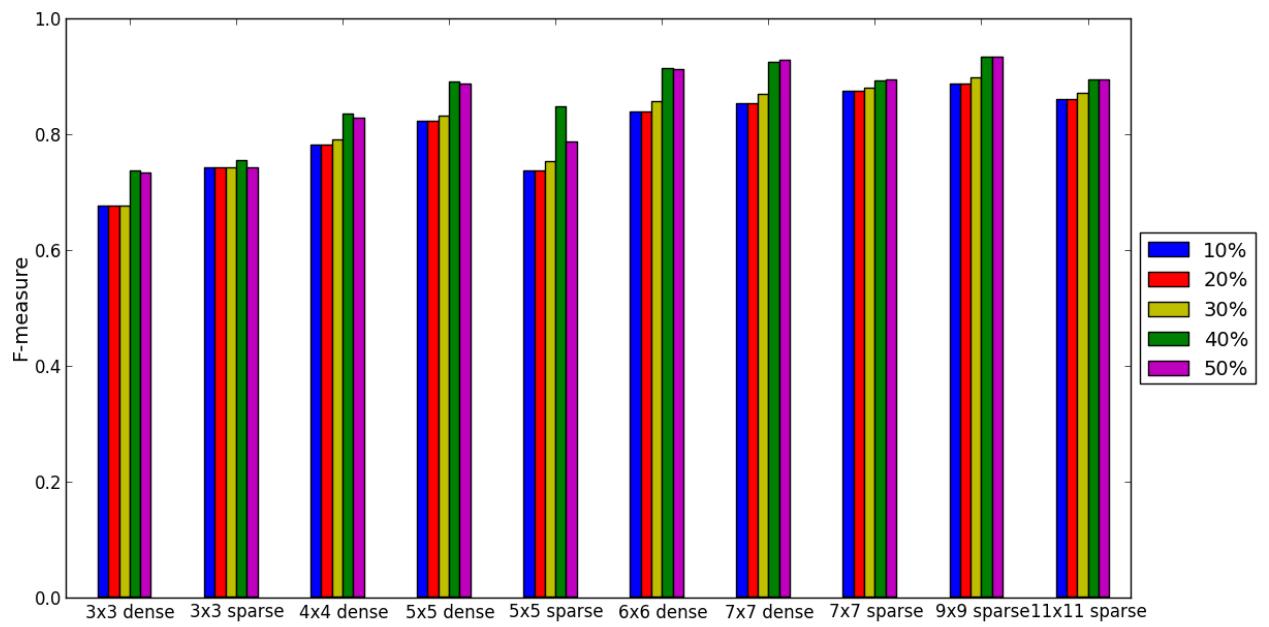


Figura 12: TIME: Classificação de parágrafos

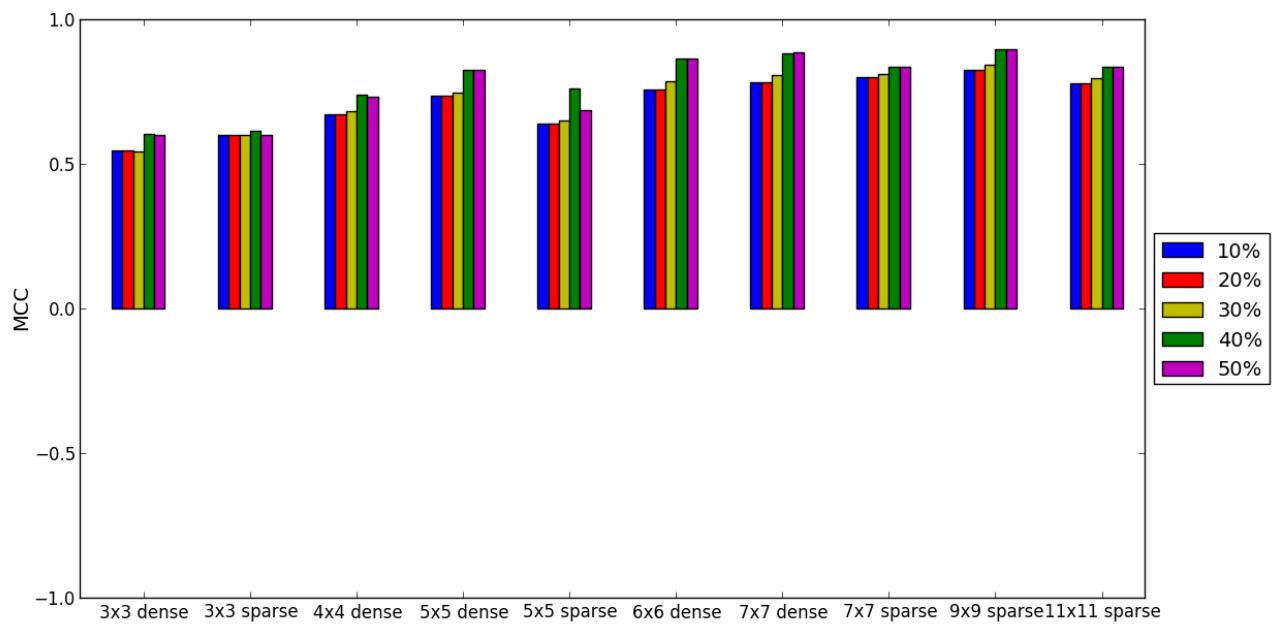


Figura 13: TIME: Classificação de parágrafos

Tabela 13: Média da precisão na classificação de títulos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.2115	0.2198	0.2301	0.2335	0.0467	0.2413	0.0910	0.2802	0.1275
20%	0.0000	0.1650	0.1894	0.2171	0.2609	0.0467	0.2485	0.0878	0.2943	0.1676
30%	0.0000	0.1352	0.1115	0.2115	0.2642	0.0467	0.2000	0.0725	0.3060	0.1495
40%	0.0000	0.1518	0.1381	0.2146	0.2637	0.0467	0.2000	0.1006	0.2945	0.1653
50%	0.0000	0.1864	0.1848	0.2334	0.2842	0.0467	0.0000	0.1069	0.3256	0.1771

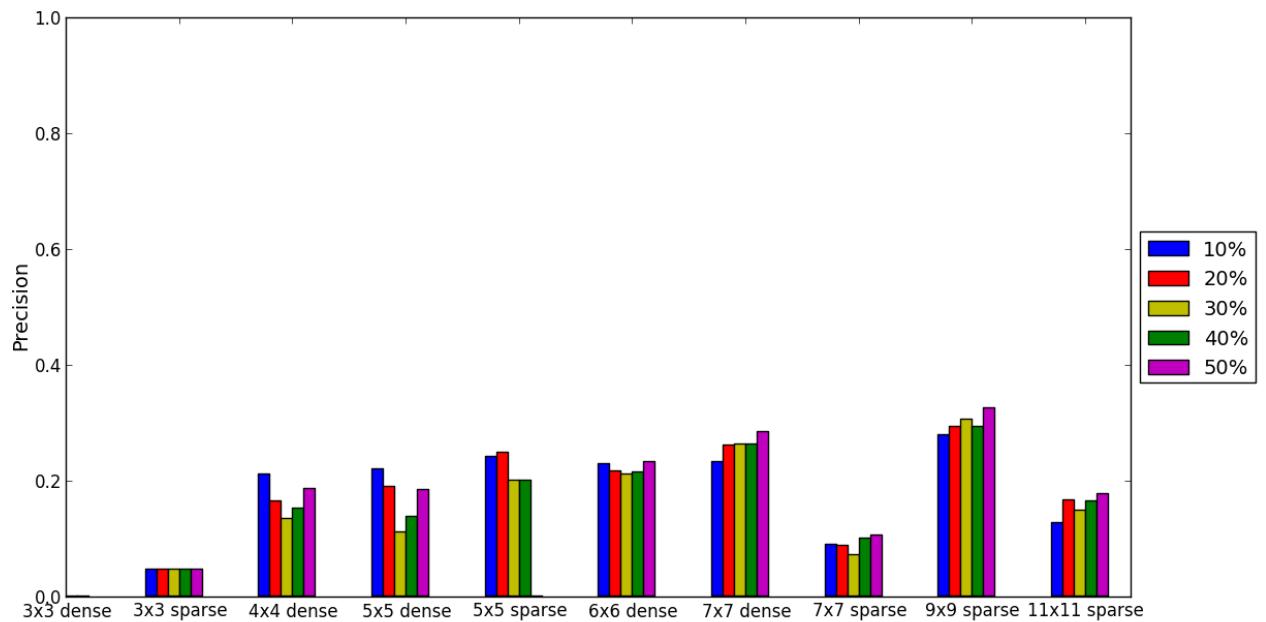


Figura 14: CACM: Classificação de títulos

Tabela 14: Média recall na classificação de títulos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0083	0.0667	0.1420	0.2059	0.1731	0.0130	0.2912	0.2506	0.3712
20%	0.0000	0.0014	0.0382	0.1209	0.2411	0.1731	0.0028	0.2550	0.2931	0.4557
30%	0.0000	0.0003	0.0049	0.0539	0.1683	0.1731	0.0000	0.1883	0.2081	0.3980
40%	0.0000	0.0002	0.0167	0.1091	0.2598	0.1731	0.0000	0.2777	0.2925	0.4640
50%	0.0000	0.0002	0.0269	0.1113	0.2709	0.1731	0.0000	0.2866	0.3531	0.4885

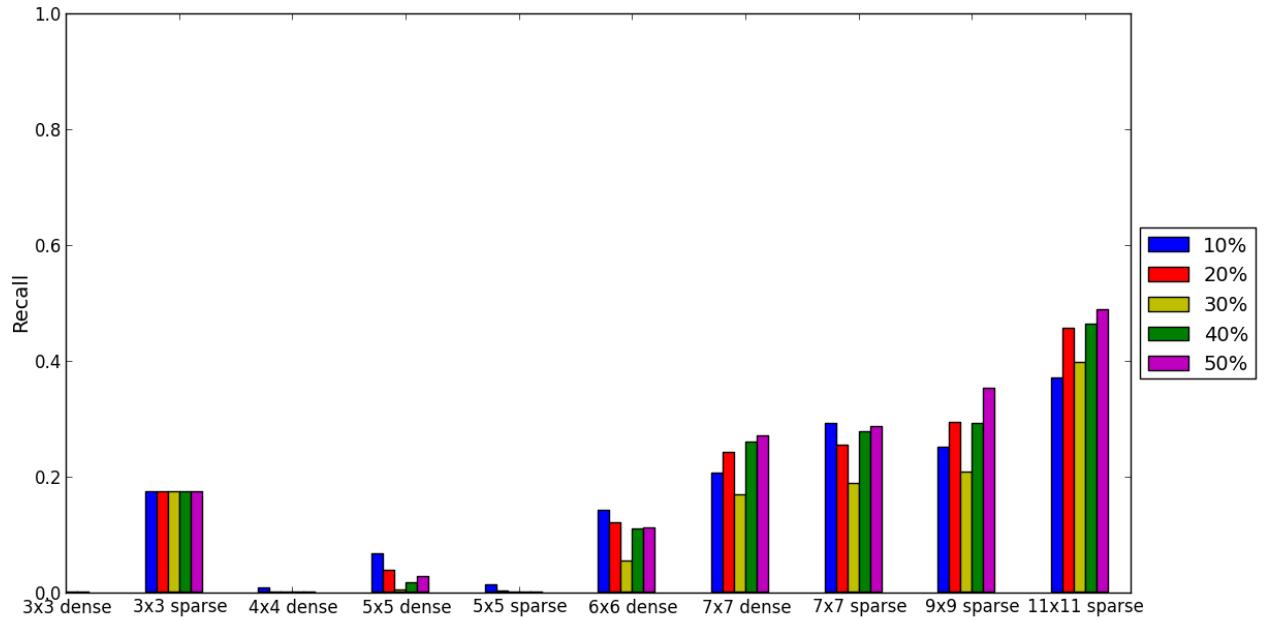


Figura 15: CACM: Classificação de títulos

Tabela 15: Média F1 na classificação de títulos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0152	0.0869	0.1428	0.1766	0.0601	0.0234	0.1119	0.2094	0.1510
20%	0.0000	0.0028	0.0560	0.1311	0.2161	0.0601	0.0054	0.1074	0.2677	0.2116
30%	0.0000	0.0005	0.0088	0.0742	0.1799	0.0601	0.0000	0.0838	0.2308	0.1858
40%	0.0000	0.0004	0.0265	0.1252	0.2319	0.0601	0.0000	0.1234	0.2731	0.2125
50%	0.0000	0.0004	0.0429	0.1331	0.2481	0.0601	0.0000	0.1302	0.3169	0.2259

Tabela 16: Média MCC na classificação de títulos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0243	0.0739	0.1184	0.1484	-0.0879	0.0357	0.0239	0.1958	0.0964
20%	0.0000	0.0078	0.0508	0.1065	0.1849	-0.0879	0.0176	0.0117	0.2342	0.1640
30%	0.0000	0.0020	0.0058	0.0676	0.1551	-0.0879	0.0006	-0.0159	0.2023	0.1264
40%	0.0000	0.0024	0.0203	0.1003	0.1975	-0.0879	0.0008	0.0319	0.2372	0.1637
50%	0.0000	0.0033	0.0416	0.1115	0.2162	-0.0879	0.0000	0.0435	0.2838	0.1851

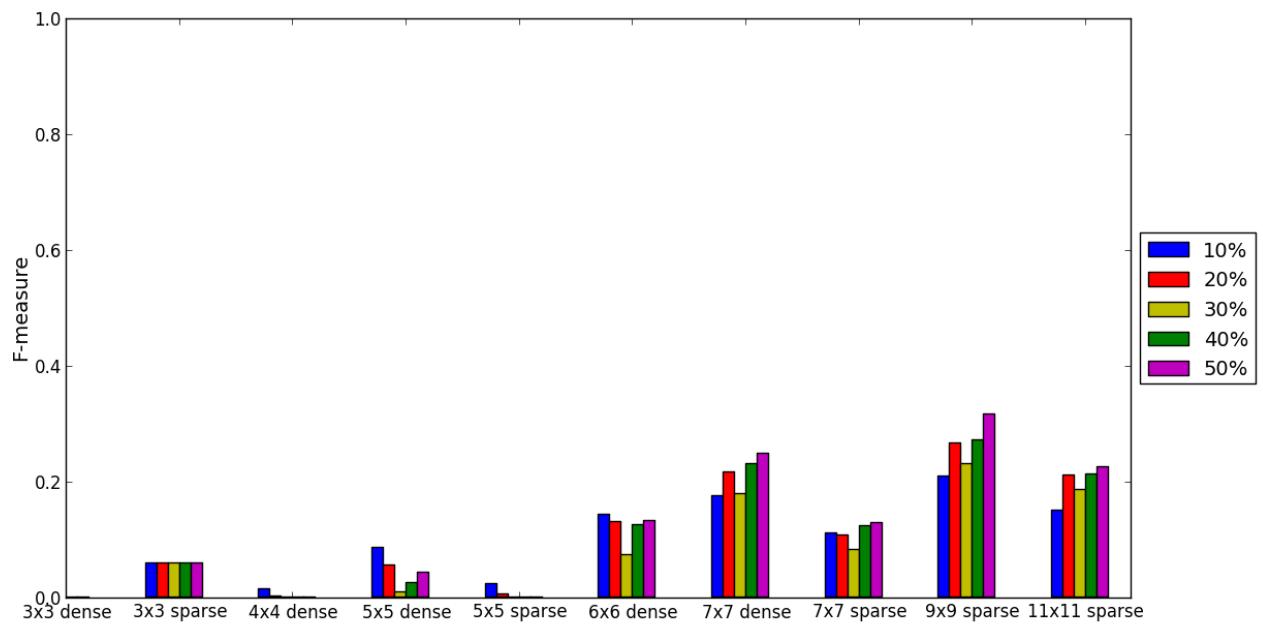


Figura 16: CACM: Classificação de títulos

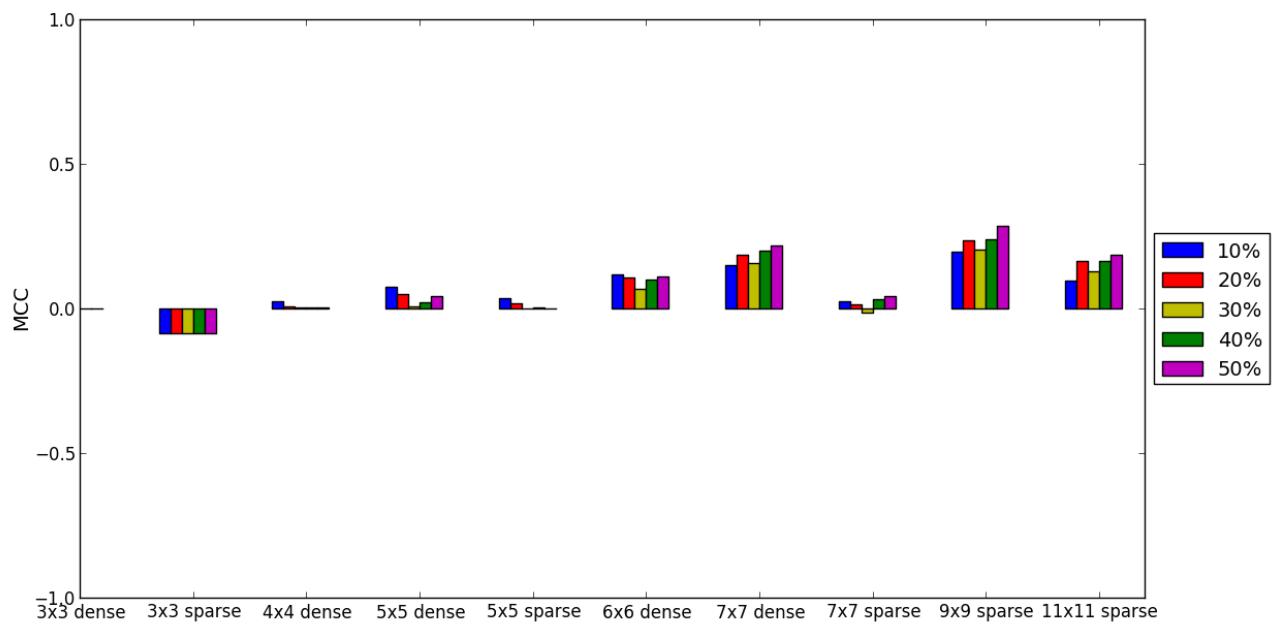


Figura 17: CACM: Classificação de títulos

Tabela 17: Média da precisão na classificação de títulos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0000	0.0020	0.0571	0.0776	0.0189	0.0000	0.0232	0.0510	0.0277
30%	0.0000	0.0000	0.0018	0.0063	0.0137	0.0189	0.0000	0.0227	0.0151	0.0217
40%	0.0000	0.0000	0.0081	0.0203	0.0447	0.0189	0.0000	0.0271	0.0600	0.0297
50%	0.0000	0.0147	0.0122	0.0495	0.0721	0.0189	0.0000	0.0274	0.1172	0.0341

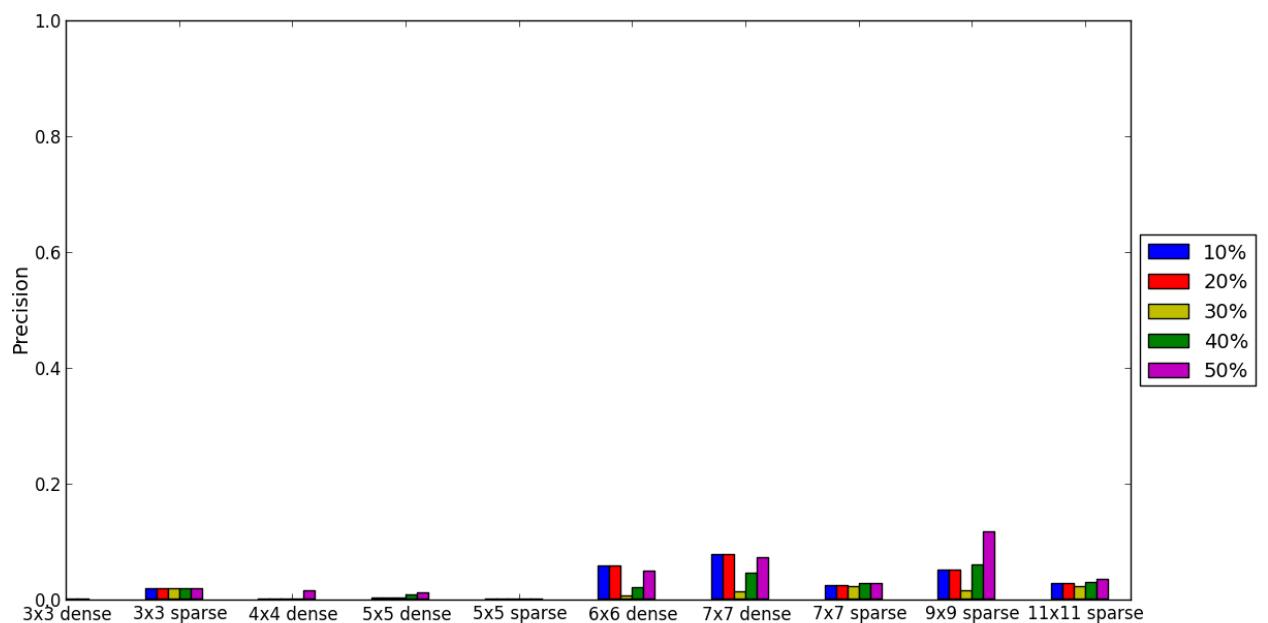


Figura 18: TIME: Classificação de títulos

Tabela 18: Média recall na classificação de títulos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0000	0.0007	0.0610	0.1410	0.1786	0.0000	0.1600	0.0451	0.2089
30%	0.0000	0.0000	0.0004	0.0053	0.0166	0.1786	0.0000	0.1558	0.0108	0.1598
40%	0.0000	0.0000	0.0001	0.0046	0.0210	0.1786	0.0000	0.1626	0.0138	0.1821
50%	0.0000	0.0000	0.0003	0.0056	0.0256	0.1786	0.0000	0.1607	0.0111	0.1943

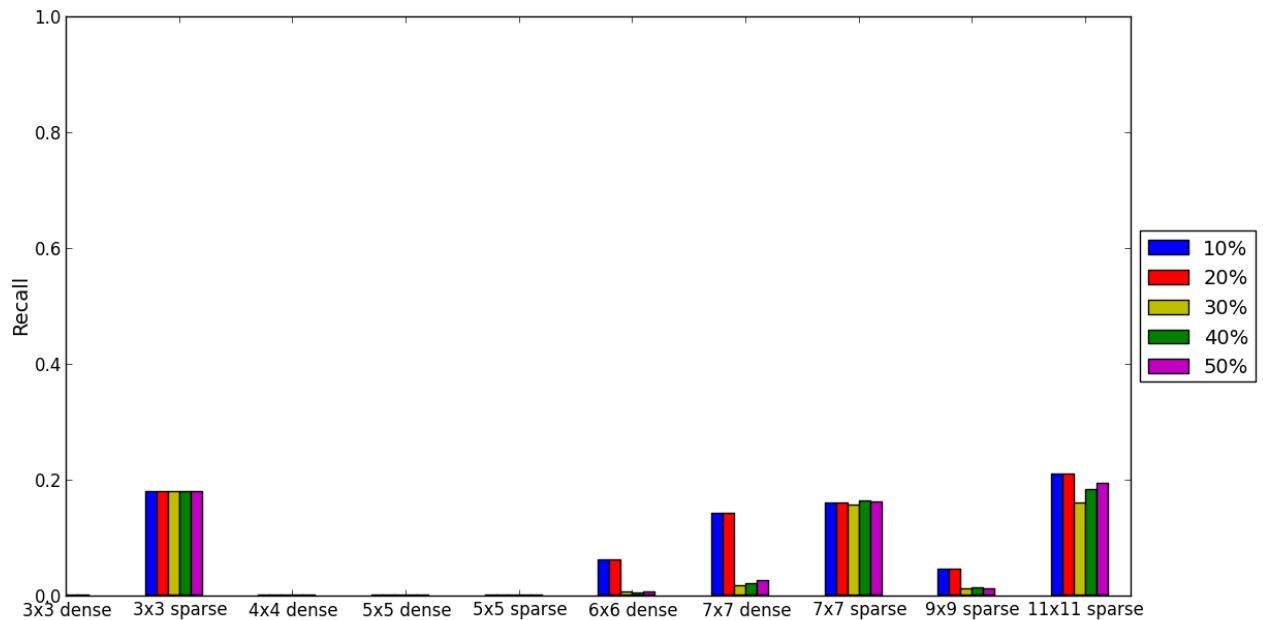


Figura 19: TIME: Classificação de títulos

Tabela 19: Média F1 na classificação de títulos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0000	0.0010	0.0568	0.0966	0.0331	0.0000	0.0392	0.0451	0.0476
30%	0.0000	0.0000	0.0007	0.0056	0.0146	0.0331	0.0000	0.0383	0.0123	0.0371
40%	0.0000	0.0000	0.0003	0.0073	0.0272	0.0331	0.0000	0.0449	0.0214	0.0494
50%	0.0000	0.0000	0.0005	0.0099	0.0367	0.0331	0.0000	0.0452	0.0199	0.0562

Tabela 20: Média MCC na classificação de títulos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	-0.0015	-0.0147	0.0281	0.0649	-0.0371	0.0000	-0.0177	0.0202	-0.0050
30%	0.0000	-0.0014	-0.0122	-0.0186	-0.0148	-0.0371	0.0000	-0.0190	-0.0087	-0.0215
40%	0.0000	0.0000	-0.0024	-0.0027	0.0109	-0.0371	0.0000	-0.0061	0.0151	0.0018
50%	0.0000	-0.0001	-0.0015	0.0069	0.0251	-0.0371	0.0000	-0.0053	0.0263	0.0121

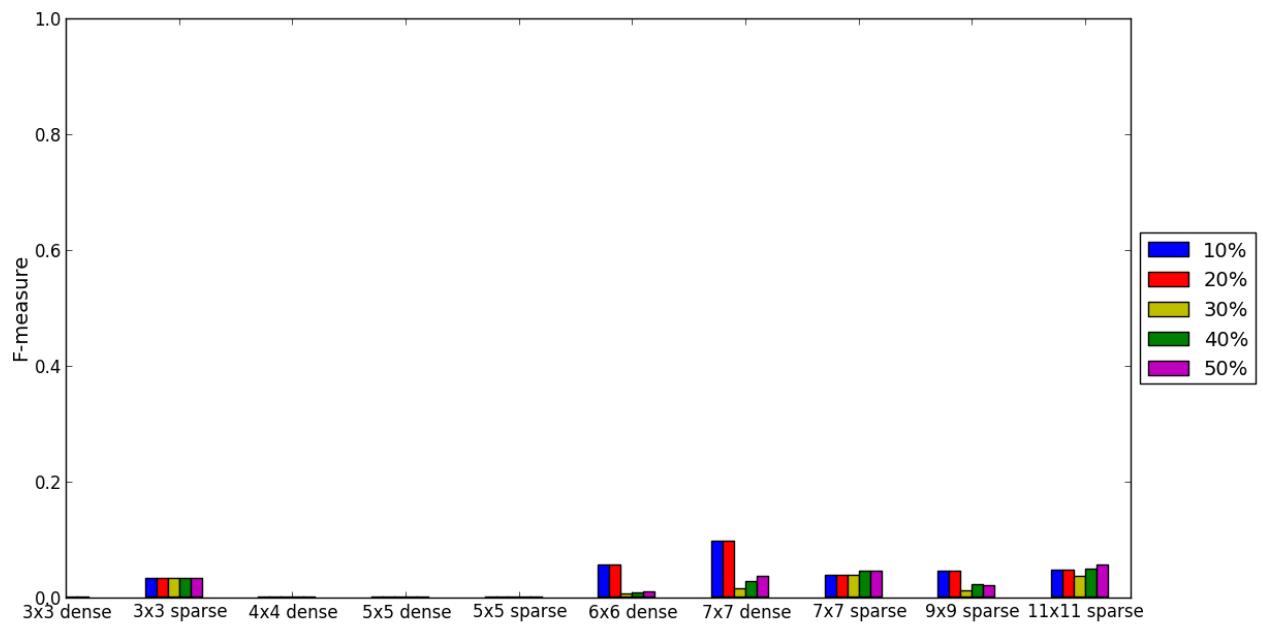


Figura 20: TIME: Classificação de títulos

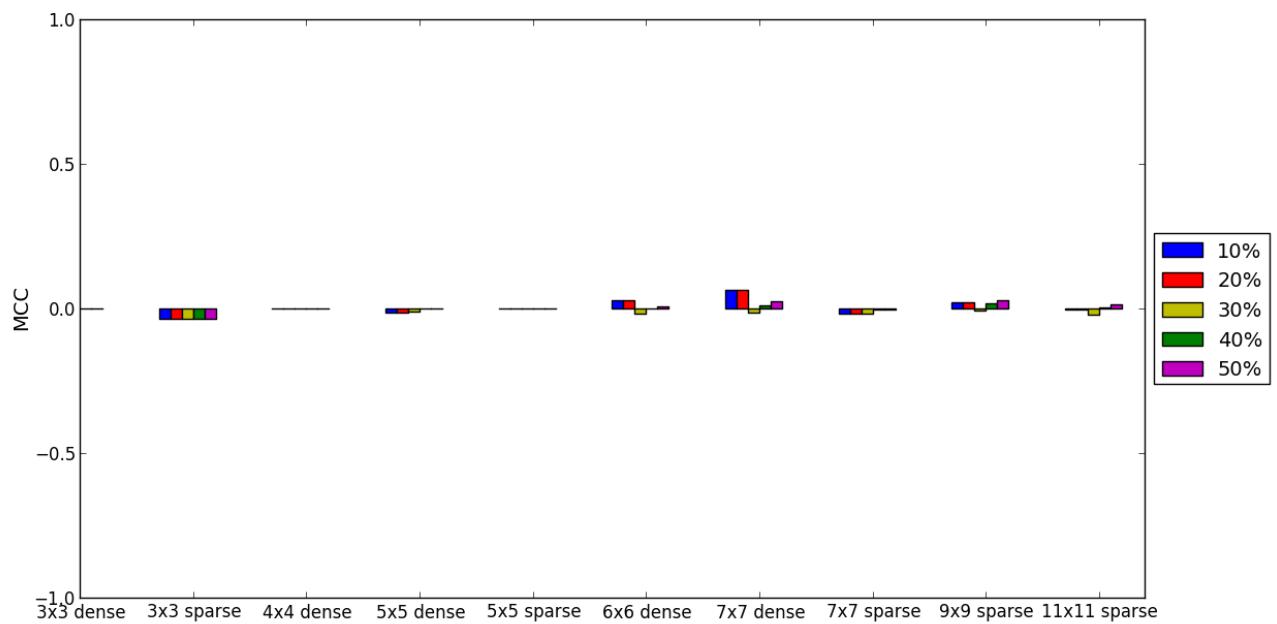


Figura 21: TIME: Classificação de títulos

6.2 Imagens finais

As tabelas 21, 22, 23, 24 e 25 apresentam as duplas de imagens segmentada e ideal para cada uma das imagens do conjunto de teste CACM. As tabelas 26, 27, 28 e 29 contém as imagens da TIME. A imagem segmentada exibida foi obtida utilizando o operador morfológico de melhor F-measure (janela 9x9 esparsa com maior número de exemplos de treinamento). A cor azul representa regiões de parágrafo, verde para títulos e vermelho para marcar o que sobrou.

Tabela 21: Segmentação da imagem 689

Ideal

number of references) follows a geometric distribution with parameter p_s . If references to an object are in fact correlated, then the object's IAT distribution will deviate from the geometric; so to measure correlation we introduce a metric that is very sensitive to this deviation. The coefficient of variation (CV) of a distribution is its standard deviation divided by its mean. To form a metric for a given trace, we take an average of IAT-CV over all references in the trace, as described in [4].

For the geometric distribution with parameter p_s , the CV is

$$\sqrt{1 - p_s}$$

In Web reference streams, in which even the most frequently accessed objects have a low probability of reference (generally much less than 1), the expected IAT-CV in the case of no temporal correlation is very close to one. Therefore, in Web reference streams, CV values close to unity suggest that reference patterns are close to the IRM (contain little temporal

recombined into a systemwide view. The idea is to merge streams coming from different locations on each source of locality we organize our findings in the table here. The results in this table are derived from measurements of Web traces, experimental evaluation of transformations of Web traces, and analytic considerations, as described in [4].

Filtering. Filtering is a query, under commonly used cache replacement policies like LRU and least frequently used (LFU), tends to remove both components of locality. It removes popularity (increases entropy) because highly popular documents are more likely to be found in the cache; and it tends to remove correlation (decreases IAT-CV) because recently referenced objects are more likely to be found near servers.

Aggregation. Aggregation of streams coming from distinct upstream sources tends to increase locality, especially near servers. This occurs because entropy decreases, while IAT-CV changes very little. Entropy decreases because of popularity effects, and the merged stream is generally more skewed than in the component streams being merged. This suggests that the presence of strong temporal locality in the original traces, occurs because globally popular documents tend to occur in each component stream, while unpopularity documents do not. This effect is most pronounced near servers where the largest number of independent streams are being merged.

Disaggregation. Disaggregation into separate streams tends to reduce locality, but its effect is only pronounced near clients. Disaggregation has little effect on entropy; this seems to be because resulting downstream components tend to have a skewed popularity distribution like the original stream. This is consistent with many studies that have shown that even among objects on a single Web server, Zipf's Law is quite pronounced. Disaggregation tends to increase IAT-CV, but this effect is only pronounced when disaggregating streams close to the client. At most other places in the Web system there is little correlation in either incoming or outgoing streams.

Exploring Stream Locality. So far, we have been devoted to decomposing the complexity of Web systems along two axes: the transformations operating on Web streams have been decomposed into filtering, merging, and filtering; and the locality property of request streams has been decomposed into popularity and correlation. Using this decomposition, we can begin to think about the underlying reasons for the various semicentral engineering of the Web. Organizing the various processes of the system along these two axes yields a set of tractable analyses that can ultimately be

Segmentada

number of references) follows a geometric distribution with parameter p_s .

If references to an object are in fact correlated, then the object's IAT distribution will deviate from the geometric; so to measure correlation we introduce a metric that is very sensitive to this deviation. The coefficient of variation (CV) of a distribution is its standard deviation divided by its mean. To form a metric for a given trace, we take an average of IAT-CV over all references in the trace, as described in [4].

For the geometric distribution with parameter p_s , the CV is

$$\sqrt{1 - p_s}$$

In Web reference streams, in which even the most frequently accessed objects have a low probability of reference (generally much less than 1), the expected IAT-CV in the case of no temporal correlation is very close to one. Therefore, in Web reference streams, CV values close to unity suggest that reference patterns are close to the IRM (contain little temporal

recombined into a systemwide view. The idea is to consider how each transformation operates on each source of locality we organize our findings in the table here. The results in this table are derived from measurements of Web traces, experimental evaluation of transformations of Web traces, and analytic considerations, as described in [4].

Filtering. Filtering is a query, under commonly used cache replacement policies like LRU and least frequently used (LFU), tends to remove both components of locality. It removes popularity (increases entropy) because highly popular documents are more likely to be found in the cache; and it tends to remove correlation (decreases IAT-CV) because recently referenced objects are more likely to be found near servers.

Aggregation. Aggregation of streams coming from distinct upstream sources tends to increase locality, especially near servers. This occurs because entropy decreases, while IAT-CV changes very little. Entropy decreases because of popularity effects, and the merged stream is generally more skewed than in the component streams being merged. This suggests that the presence of strong temporal locality in the original traces, occurs because globally popular documents tend to occur in each component stream, while unpopularity documents do not. This effect is most pronounced near servers where the largest number of independent streams are being merged.

Disaggregation. Disaggregation into separate streams tends to reduce locality, but its effect is only pronounced near clients. Disaggregation has little effect on entropy; this seems to be because resulting downstream components tend to have a skewed popularity distribution like the original stream. This is consistent with many studies that have shown that even among objects on a single Web server, Zipf's Law is quite pronounced. Disaggregation tends to increase IAT-CV, but this effect is only pronounced when disaggregating streams close to the client. At most other places in the Web system there is little correlation in either incoming or outgoing streams.

Exploring Stream Locality. So far, we have been devoted to decomposing the complexity of Web systems along two axes: the transformations operating on Web streams have been decomposed into filtering, merging, and filtering; and the locality property of request streams has been decomposed into popularity and correlation. Using this decomposition, we can begin to think about the underlying reasons for the various semicentral engineering of the Web. Organizing the various processes of the system along these two axes yields a set of tractable analyses that can ultimately be

Summary of the effects of the different transformations on the temporal locality

Thus, IAT-CV can be used for capturing temporal correlation. In a manner similar to normalized entropy, the importance of correlation in a reference stream can be measured by its effect on a reference IAT-CV metric. When a reference stream (removing correlations) the resulting hit ratio tends to decrease; in [4] we find this decrease is strongly correlated with IAT-CV.

Exploring Stream Locality
So far, we have been devoted to decomposing the complexity of Web systems along two axes: the transformations operating on Web streams along two axes: the complexity of Web systems along two axes: the transformations operating on Web streams have been decomposed into filtering, merging, and filtering; and the locality property of request streams has been decomposed into popularity and correlation. Using this decomposition, we can begin to think about the underlying reasons for the various semicentral engineering of the Web. Organizing the various processes of the system along these two axes yields a set of tractable analyses that can ultimately be

Filtering

Aggregation

Disaggregation

Little effect

Decreases IATCV

Link effect

Decreases IATCV

Link effect

Filtering

Aggregation

Disaggregation

Little effect

Decreases IATCV

Link effect

Decreases IATCV

Link effect

communications of the ACM January 2005/Vol. 48, No. 1 87

communications of the ACM January 2005/Vol. 48, No. 1 87

Tabela 22: Segmentação da imagem 692

Timeliness of Investor Relations Data at Corporate Web Sites

ONLINE DATA FOR INVESTORS IS OFTEN STALE, EVEN WHEN OF HIGH QUALITY. HOW CAN THIS SITUATION BE IMPROVED?

In recent years Securities and Exchange Commission (SEC) regulations have promoted the widespread and speedy dissemination of financial information to all users. The SEC Web site provides the following information regarding EDGAR archives of downloadable company financial reports and other filings (www.sec.gov/edgar/aboutedgar.htm): "Its primary purpose is to increase the efficiency of financial information in the market for the benefit of investors, corporations, and the economy by accelerating the receipt, acceptance, dissemination, and analysis of timely securities information filed with the agency." (emphasis added)

In October 2000, the SEC Regulation FD (fair disclosure) became effective. Although it requires companies to disclose material nonpublic investor relations (IR) information via press releases, Regulation FD specifically encourages them to use the Internet to accomplish broad-based rapid dissemination [1]. The Sarbanes-Oxley Act of 2002, which became law in July 2002, requires larger companies to accelerate the filing of their Forms 10-Q and 10-K. In addition, these firms will be required to disc-

lose in Form 10-K whether they provide these reports at their Web sites "as soon as reasonably practicable" after filing them with the SEC [8].

Corporate IR personnel also are studying ways to use the Internet to rapidly disseminate IR information. The Standards of Practice for Investor Relations [10] states:

"Information should be released in a manner designed to reach the widest possible audience, including individual investors."

Clearly, the rapid and widespread distribution of IR information via technology is desirable, both from the social (SEC) and economic (investor relations) perspectives.

Use of corporate Web sites for this purpose is an important activity with a relatively short history. Several studies have investigated

Timeliness of Investor Relations Data at Corporate Web Sites

ONLINE DATA FOR INVESTORS IS OFTEN STALE, EVEN WHEN OF HIGH QUALITY. HOW CAN THIS SITUATION BE IMPROVED?

In recent years Securities and Exchange Commission (SEC) regulators have promoted the widespread and speedy dissemination of financial information to all users. The SEC Web site provides the following information regarding EDGAR archives of downloadable company financial reports and other filings (www.sec.gov/edgar/aboutedgar.htm): "Its primary purpose is to increase the efficiency of financial information in the market for the benefit of investors, corporations, and the economy by accelerating the receipt, acceptance, dissemination, and analysis of timely securities information filed with the agency." (emphasis added)

In October 2000, the SEC Regulation FD (fair disclosure) became effective. Although it requires companies to disclose material nonpublic investor relations (IR) information via press releases, Regulation FD specifically encourages them to use the Internet to accomplish broad-based rapid dissemination [1]. The Sarbanes-Oxley Act of 2002, which became law in July 2002, requires larger companies to accelerate the filing of their Forms 10-Q and 10-K. In addition, these firms will be required to disc-

lose in Form 10-K whether they provide these reports at their Web sites "as soon as reasonably practicable" after filing them with the SEC [8].

Corporate IR personnel also are studying ways to use the Internet to rapidly disseminate IR information. The Standards of Practice for Investor Relations [10] states:

"Information should be released in a manner designed to reach the widest possible audience, including individual investors."

Clearly, the rapid and widespread distribution of IR information via technology is desirable, both from the social (SEC) and economic (investor relations) perspectives.

Use of corporate Web sites for this purpose is an important activity with a relatively short history. Several studies have investigated

BY MICHAEL ETTREGE AND JOHN GERDES, JR.

communications of the ACM January 2005/Vol. 48, No. 1 95

BY MICHAEL ETTREGE AND JOHN GERDES, JR.

communications of the ACM January 2005/Vol. 48, No. 1 95

Tabela 23: Segmentação da imagem 695

values of "0" and "1." For the Newest Content variable, at least three-fourths of the observations are coded "1" or less. For this reason, and after examining OLS regression residuals, we decided to employ ordinal logistic regression.

Table 3 presents the logistic regression results for dependent variables Newest Content and Oldest Content. Panel A contains results with Newest Content as the dependent variable. Only two of the variables have significant coefficients. The sign of financial health is opposite to expectations. Although firms having higher current year profits might post their financial reports on the Web more often [as in (6)], financially healthy firms do not provide the freshest general IR content at their sites. In fact, we find that the age of the newest IR content at firms' sites is positively associated with financial firm size. A possible explanation is that larger firms have more favorable information at their sites, although that information might not include the annual report (as in [6]). Such favorable information might include, for example, performance predictions of improved future performance, or frequent announcements of major customer orders.

The second significant variable is the number of file types present at the IR site. It is negatively associated with age of newest content, as expected. Our favored explanation is that number of file types is a proxy for commitment to high-quality Web-based IR, which implicitly includes fresh content. An alternative explanation is that larger firms tend to release content faster; the more likely it is that some of the content is very fresh. The third significant coefficient involves the percentage of bad home links at the IR site. If one views bad links as a proxy for quality of content at the IR site, we would expect the percentage of bad home links to be positively associated with the age of newest content. Instead we observe the opposite. Fresher content is associated with a greater percentage of bad home links. We conclude that newer content sometimes disrupts links, and that firms identify such disruptions, and repair them, with a time lag. Thus, as the age of a firm's newest content increases, the number of bad home links tends to decrease.

Panel B of Table 3 indicates that age of oldest content is unaffected by firm size, issuance of stock, or percentage of bad links. Only one variable, N File Types, is significant in both panels, and its sign is irreducible switching across panels. It appears that factors determining the age of oldest content are quite different from those affecting age of newest content. Stale IR content is interesting because it potentially exposes firms to liability risk at the hands of investors. The first significant coefficient in Panel B is number of IR Web

values of "0" and "1." For the Newest Content variable, at least three-fourths of the observations are coded "1" or less. For this reason, and after examining OLS regression residuals, we decided to employ ordinal logistic regression.

Table 3 presents the logistic regression results for dependent variables Newest Content and Oldest Content. Panel A contains results with Newest Content as the dependent variable. Only two of the variables have significant coefficients. The sign of financial health is opposite to expectations. Although firms having higher current year profits might post their financial reports on the Web more often [as in (6)], financially healthy firms do not provide the freshest general IR content at their sites. In fact, we find that the age of the newest IR content at firms' sites is positively associated with financial firm size. A possible explanation is that larger firms have more favorable information at their sites, although that information might not include the annual report (as in [6]). Such favorable information might include, for example, performance predictions of improved future performance, or frequent announcements of major customer orders.

The second significant variable, the number of file types present at the IR site, is negatively associated with age of newest content, as expected. Our favored explanation is that number of file types is a proxy for commitment to high-quality Web-based IR, which implicitly includes fresh content. An alternative explanation is that larger firms tend to release content faster; the more likely it is that some of the content is very fresh. The third significant coefficient involves the percentage of bad home links at the IR site. If one views bad links as a proxy for quality of content at the IR site, we would expect the percentage of bad home links to be positively associated with the age of newest content. Instead we observe the opposite. Fresher content is associated with a greater percentage of bad home links. We conclude that newer content sometimes disrupts links, and that firms identify such disruptions, and repair them, with a time lag. Thus, as the age of a firm's newest content increases, the number of bad home links tends to decrease.

Panel B of Table 3 indicates that age of oldest content is unaffected by firm size, issuance of stock, or percentage of bad links. Only one variable, N File Types, is significant in both panels, and its sign is irreducible switching across panels. It appears that factors determining the age of oldest content are quite different from those affecting age of newest content. Stale IR content is interesting because it potentially exposes firms to liability risk at the hands of investors. The first significant coefficient in Panel B is number of IR Web

Stale IR content is interesting because it potentially exposes firms to liability risk at the hands of investors.

Stale IR content is interesting because it potentially exposes firms to liability risk at the hands of investors.

Tabela 24: Segmentação da imagem 802

THE UNCONSCIOUS ART OF SOFTWARE TESTING | Phillip G. Armour

The Unconscious Art of Software Testing

The subtle psychology of testing.

In *The Art of Software Testing*, Glenford Myers argued that "the most important considerations in software testing are issues of economics and human psychology" [6]. In fact, the most important consideration of any software development practice is (or should be) issues of economics and human psychology. This challenge in my system is that testers are trying to develop a way to find out if they don't know what they don't know. This is equivalent to a group of scientists trying to devise an experiment to reveal something they are not looking for. In fact, as Thomas Kuhn pointed out, the image we have of the scientist is that he is exploring his territory and finding out things we never knew is at odds with the reality [5]. Scientists almost always work with well-defined hypotheses and it is very difficult to determine their behavior in operation unless you execute them in some controlled fashion. Simulation is taking over traditional testing in some of these areas, but a

conscious intellectual process. It is not possible to be wholly deterministic about testing since we don't know what to be deterministic about. Testing, probably more than any other activity in

have found out earlier but didn't due to some negligence on our part. However, exposing things we didn't know we didn't know by dynamically executing the knowledge contained in a system is not that hard. In fact, we have stopped punishing testers for finding defects, though rewarding them for the same has in peril.

Some of the most effective and efficient way to find certain defects is to test for them. This does not in any way substitute for good engineering practices and frequent inspections, much as inspections. Indeed, without such processes in place and working, any attempt at dynamic testing quickly becomes over whelmed and quite ineffective. But there are certain kinds of problems that are very difficult to identify and analyze. For example, I work with perform enormous quantities of integration testing. They must do this because they have to make sure that their territory and finding out things we never knew is at odds with the reality [5]. Scientists almost always work with well-defined hypotheses and it is very difficult to determine their behavior in operation unless you execute them in some controlled fashion. Simulation is taking over traditional testing in some of these areas, but a

THE UNCONSCIOUS ART OF SOFTWARE TESTING | Phillip G. Armour

The Unconscious Art of Software Testing

The subtle psychology of testing.

In *The Art of Software Testing*, Glenford Myers argued that "the most important considerations in software testing are issues of economics and human psychology" [6]. In fact, the most important consideration of any software development practice is (or should be) issues of economics and human psychology. This challenge in my system is that testers are trying to develop a way to find out if they don't know what they don't know. This is equivalent to a group of scientists trying to devise an experiment to reveal something they are not looking for. In fact, as Thomas Kuhn pointed out, the image we have of the scientist is that he is exploring his territory and finding out things we never knew is at odds with the reality [5]. Scientists almost always work with well-defined hypotheses and it is very difficult to determine their behavior in operation unless you execute them in some controlled fashion. Simulation is taking over traditional testing in some of these areas, but a

conscious intellectual process. It is not possible to be wholly deterministic about testing since we don't know what to be deterministic about. Testing, probably more than any other activity in

have found out earlier but didn't due to some negligence on our part. However, exposing things we didn't know we didn't know by dynamically executing the knowledge contained in a system is not that hard. In fact, we have stopped punishing testers for finding defects, though rewarding them for the same has in peril.

Some of the most effective and efficient way to find certain defects is to test for them. This does not in any way substitute for good engineering practices and frequent inspections, much as inspections. Indeed, without such processes in place and working, any attempt at dynamic testing quickly becomes over whelmed and quite ineffective. But there are certain kinds of problems that are very difficult to identify and analyze. For example, I work with perform enormous quantities of integration testing. They must do this because they have to make sure that their territory and finding out things we never knew is at odds with the reality [5]. Scientists almost always work with well-defined hypotheses and it is very difficult to determine their behavior in operation unless you execute them in some controlled fashion. Simulation is taking over traditional testing in some of these areas, but a

Tabela 25: Segmentação da imagem 803

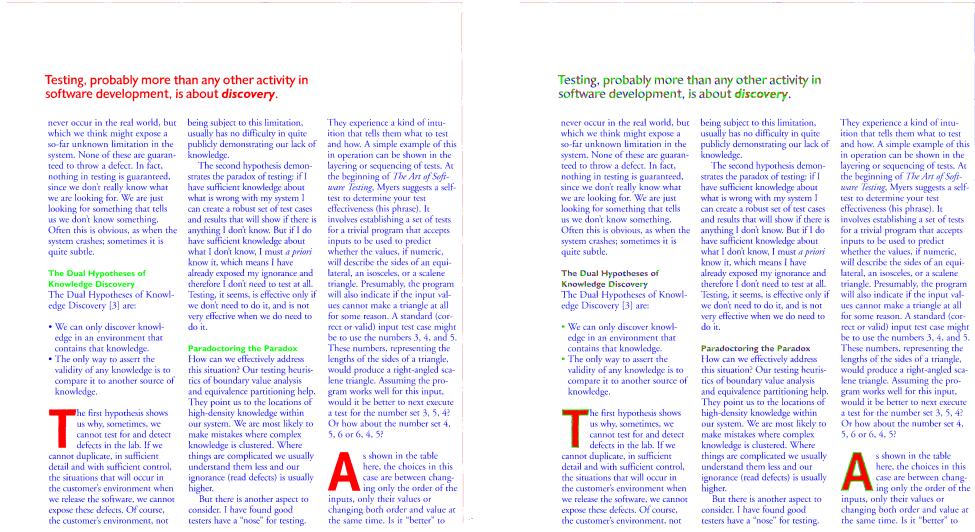


Tabela 26: Segmentação da imagem 783



Tabela 27: Segmentação da imagem 784



Tabela 28: Segmentação da imagem 785

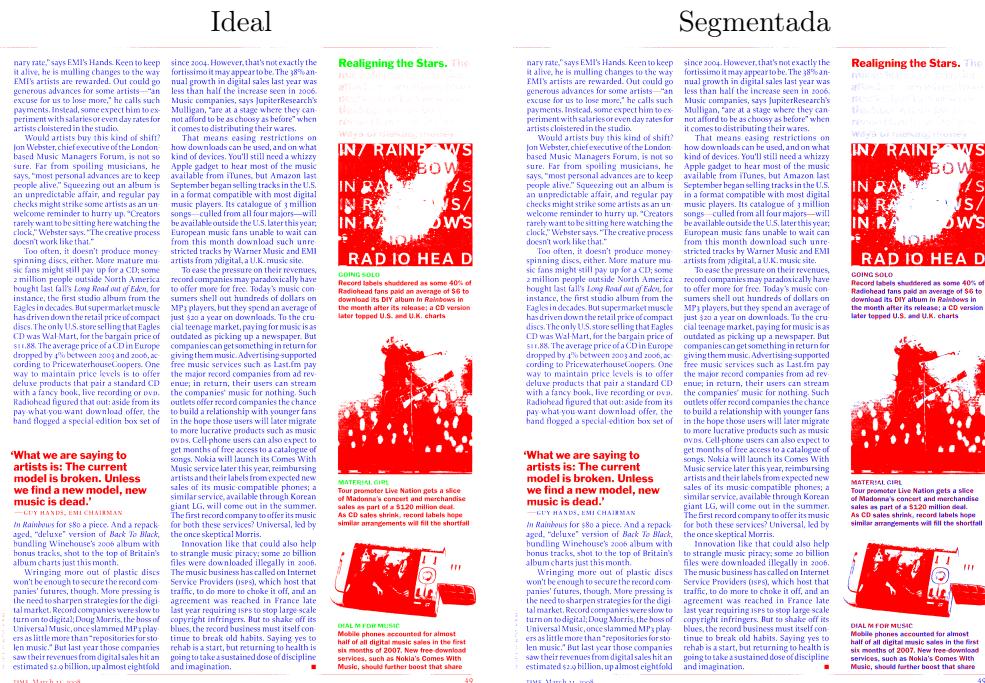


Tabela 29: Segmentação da imagem 786

Ideal



BOOKS
The Jihadi Next Door. What turns a seemingly normal person into a terrorist? By ARVIN BACHER

“**A**NYONE WHO SAWED OFF his testicles was the kind of guy you could have taken home to Thanksgiving dinner,” says Sheik. He once jumped in front of a train in a London tube station, killing himself and his mother. But he also, in the name of the Islamic cause, gloatfully became a hostage with decapitation in 1994. That hostage survived, but Danny Pearl, the Wall Street Journal reporter who was held captive by Sheik is charged with kidnapping in January 2002, did not. The video of Pearl’s beheading, which has been widely distributed (though the identity of the actual knife wielder remains unknown), How does someone become a “gentle person you could meet” according to his brother—turn terrorist?

In his new book, *Leaderless Jihad*, coauthored by the author of 2004’s *Understanding Terror Networks*, forensic psychiatrist Marc Sageman attempts to understand the psychological profile of Islamist terrorists. Like his earlier book, *Leaderless Jihad* discredits conventional wisdom about terrorism, drawing anecdotes and conjecture in favor of hard data and statistics. And statistically, there are few.

“It ceases to view terrorists as alien crea-

tures who exist outside normal patterns of social interaction,” he writes. But he notes that the “softest” members of al Qaeda, those who do not make capable terrorists—they seldom take orders and are rarely willing to sacrifice themselves for the sake of building a better world—“are also younger and less violent than the real threat, waging the ‘leaderless jihad’ of the book’s title chapter.”

Power and lack of opportunity are not necessarily the factors that drive young men to commit violence in al Qaeda’s name. Sheik was middle class and well educated, a private pilot. “They view themselves as warriors willing to sacrifice themselves for the sake of building a better world,” says Sageman, “but this gives meaning to their lives.” They are also younger and less violent, according to the author, than they are given credit for.

Because of security crackdowns, they are unable to reach out to al Qaeda’s original leaders, like Osama bin Laden, through Internet forums for guidance and bomb-making expertise. The Madrid train bombing, the publication of most-wanted lists, and the staging of press conferences that proclaims the identities of the actual knife-wielders remain unknown). How does someone become a “gentle person you could meet” according to his brother—turn terrorist?

In his new book, *Leaderless Jihad*, coauthored by the author of 2004’s *Understanding Terror Networks*, forensic psychiatrist Marc Sageman attempts to understand the psychological profile of Islamist terrorists. Like his earlier book, *Leaderless Jihad* discredits conventional wisdom about terrorism, drawing anecdotes and conjecture in favor of hard data and statistics. And statistically, there are few.

“It ceases to view terrorists as alien crea-

Many terrorists share qualities with ordinary people; they can be cooperative, goal-oriented and intelligent

HORN March 21, 2008

Segmentada



BOOKS
The Jihadi Next Door. What turns a seemingly normal person into a terrorist? By ARVIN BACHER

“**A**NYONE WHO SAWED OFF his testicles was the kind of guy you could have taken home to Thanksgiving dinner,” says Sheik. He once jumped in front of a train in a London tube station, killing himself and his mother. But he also, in the name of the Islamic cause, gloatfully became a hostage with decapitation in 1994. That hostage survived, but Danny Pearl, the Wall Street Journal reporter who was held captive by Sheik is charged with kidnapping in January 2002, did not. The video of Pearl’s beheading, which has been widely distributed (though the identity of the actual knife wielder remains unknown), How does someone become a “gentle person you could meet” according to his brother—turn terrorist?

In his new book, *Leaderless Jihad*, coauthored by the author of 2004’s *Understanding Terror Networks*, forensic psychiatrist Marc Sageman attempts to understand the psychological profile of Islamist terrorists. Like his earlier book, *Leaderless Jihad* discredits conventional wisdom about terrorism, drawing anecdotes and conjecture in favor of hard data and statistics. And statistically, there are few.

“It ceases to view terrorists as alien crea-

Many terrorists share qualities with ordinary people; they can be cooperative, goal-oriented and intelligent

HORN March 21, 2008

Public enemy Sheik’s brother called him “the kindest, most gentle person you could meet”

“**a**l Qaeda Central” made up of those who have sworn an oath of loyalty to Osama bin Laden, is made up of the same people, mobilizing radicalized Islamists around the world without any contact with the real threat, waging the “leaderless jihad” of the book’s title chapter.

Power and lack of opportunity are not necessarily the factors that drive young men to commit violence in al Qaeda’s name. Sheik was middle class and well educated, a private pilot. “They view themselves as warriors willing to sacrifice themselves for the sake of building a better world,” says Sageman, “but this gives meaning to their lives.” They are also younger and less violent, according to the author, than they are given credit for.

Because of security crackdowns, they are unable to reach out to al Qaeda’s original leaders, like Osama bin Laden, through Internet forums for guidance and bomb-making expertise. The Madrid train bombing, the publication of most-wanted lists, and the staging of press conferences that proclaims the identities of the actual knife-wielders remain unknown). How does someone become a “gentle person you could meet” according to his brother—turn terrorist?

In his new book, *Leaderless Jihad*, coauthored by the author of 2004’s *Understanding Terror Networks*, forensic psychiatrist Marc Sageman attempts to understand the psychological profile of Islamist terrorists. Like his earlier book, *Leaderless Jihad* discredits conventional wisdom about terrorism, drawing anecdotes and conjecture in favor of hard data and statistics. And statistically, there are few.

“It ceases to view terrorists as alien crea-

Many terrorists share qualities with ordinary people; they can be cooperative, goal-oriented and intelligent

HORN March 21, 2008

Public enemy Sheik’s brother called him “the kindest, most gentle person you could meet”

“**a**l Qaeda Central” made up of those who have sworn an oath of loyalty to Osama bin Laden, is made up of the same people, mobilizing radicalized Islamists around the world without any contact with the real threat, waging the “leaderless jihad” of the book’s title chapter.

Power and lack of opportunity are not necessarily the factors that drive young men to commit violence in al Qaeda’s name. Sheik was middle class and well educated, a private pilot. “They view themselves as warriors willing to sacrifice themselves for the sake of building a better world,” says Sageman, “but this gives meaning to their lives.” They are also younger and less violent, according to the author, than they are given credit for.

Because of security crackdowns, they are unable to reach out to al Qaeda’s original leaders, like Osama bin Laden, through Internet forums for guidance and bomb-making expertise. The Madrid train bombing, the publication of most-wanted lists, and the staging of press conferences that proclaims the identities of the actual knife-wielders remain unknown). How does someone become a “gentle person you could meet” according to his brother—turn terrorist?

In his new book, *Leaderless Jihad*, coauthored by the author of 2004’s *Understanding Terror Networks*, forensic psychiatrist Marc Sageman attempts to understand the psychological profile of Islamist terrorists. Like his earlier book, *Leaderless Jihad* discredits conventional wisdom about terrorism, drawing anecdotes and conjecture in favor of hard data and statistics. And statistically, there are few.

“It ceases to view terrorists as alien crea-

HORN March 21, 2008

7 Conclusão

Observamos comportamentos diferentes na classificação de parágrafos e títulos. A quantidade de imagens demonstrou-se menos relevante na classificação parágrafos do que na classificação de títulos. Isto se deve a maior quantidade de exemplos por página de texto de parágrafo do que de títulos.

O tamanho da janela é o fator mais sensível, tanto para a qualidade quanto para o tempo de processamento. Janelas maiores porém esparsas apresentaram os melhores resultados, sendo o melhor deles observado para janelas 9x9 esparsas, atingindo F-measure de 0,96 para CACM e 0,93 para a TIME.

Porém nem sempre a maior janela produz o melhor resultado. As janelas 11x11 desempenham pior que as 9x9. Quanto maior a janela, mais exemplos são necessários. Podemos ver que em janelas maiores o tamanho do conjunto de treinamento possui maior influência na qualidade. Com janelas até 5x5 a quantidade de exemplos quase não altera a qualidade ou até piora.

Exemplos ruins também podem comprometer a qualidade do operador. Fontes muito parecidas para títulos e outros tipos de texto podem fornecer exemplos contraditórios, ou seja, hora um certo padrão consta como título hora não. Podemos ver um exemplo deste fenômeno na imagem 23 onde o texto à direita foi classificado como título por possuir características de título. Parte do motivo por este problema ocorrer é decorrente da decisão de testarmos apenas dois tipos de região (parágrafo e título). Possivelmente, caso tivéssemos classificados outros tipos de texto, poderíamos obter resultados melhores na etapa de consensualização.

Uma outra possível melhoria seria trabalhar com imagens redimensionadas (reduzidas), fazendo com que os títulos ficasse menores, cabendo melhor dentro das janelas utilizadas.

Não pudemos testar janelas maiores que 11x11 pois o tempo de processamento cresce muito rapidamente tornando o processo muito custoso.

A comparação com outros métodos foi comprometida pela escolha das regiões a serem segmentadas. No artigo [8] vemos os resultados dos principais métodos para textos em geral e não parágrafos e títulos separadamente. Um outro problema é o método para avaliação, que utiliza polígonos isotéticos, como proposto no artigo [7]. Teríamos que realizar algum tipo de transformação que definiria polígonos em torno dos pixeis de uma certa região, assim como fez o método descrito em [9].

Acreditamos que o método proposto pode ser empregado na extração de texto de documentos. Porém, diversos ajustes de otimização seriam necessários para criar uma ferramenta utilizável em situações práticas. O treinamento dos operadores é muito lento e aplicação também.

8 Apêndice

8.1 Algoritmo de Otsu para binarização

Implementar e avaliar cada algoritmo binarizador vai além do escopo deste trabalho, portanto escolhemos um bom algoritmo segundo os resultados obtidos em [10]. Como o próprio artigo aponta, não foi encontrado um método que apresentasse desempenho superior em todas os cenários de teste realizados.

Os requisitos para a escolha foram o da independência de supervisionamento e parametrização. Caso o algoritmo demandasse ajustes específicos de acordo com a imagem, o seu uso comprometeria a promessa de automatização. Dentre a lista de soluções com esta característica, escolhemos o algoritmo de Otsu [11], por ser de fácil implementação e apresentar resultados satisfatórios nos experimentos realizados.

O algoritmo de Otsu encontra um nível de cinza t tal que a soma ponderada da variância

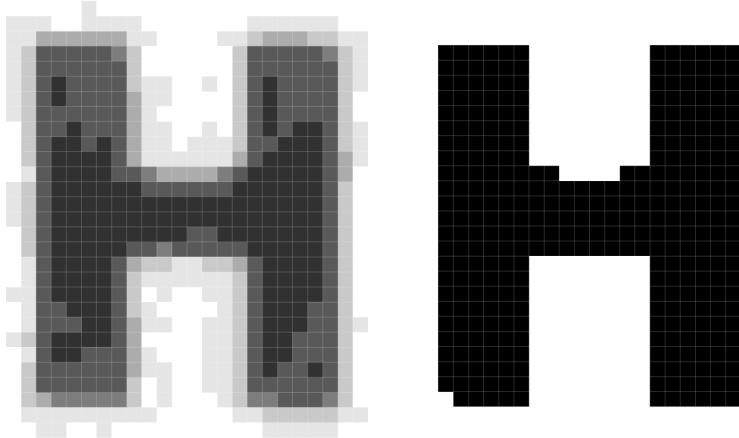
dentro das classes $\mathbb{F} = \{x: f(x) \geq t\}$ (foreground) e $\mathbb{B} = \{x: f(x) < t\}$ (background) seja minimizada, ou seja,

$$t = \operatorname{argmin}\{w_{\mathbb{B}}\sigma_{\mathbb{B}}^2 + w_{\mathbb{F}}\sigma_{\mathbb{F}}^2\} \quad (8)$$

onde $w_{\mathbb{B}} = \frac{|\mathbb{B}|}{|E|}$ e $w_{\mathbb{F}} = \frac{|\mathbb{F}|}{|E|}$ são os pesos respectivamente do background e foreground e $\sigma_{\mathbb{B}}^2$ e $\sigma_{\mathbb{F}}^2$ são as variâncias das classes.

A tabela 31 apresenta um passo a passo do algoritmo aplicado à figura 30.

Tabela 30: Imagem original em escala de cinza e correspondente binarizada com $t = 50\%$.



Como podemos notar, para $t = 50\%$ atingimos o menor valor de $\sigma_w^2 \approx 512693389$. Neste caso o limiar coincide com o único vale no histograma, porém isto nem sempre será válido. Algumas imagens não possuem vales bem definidos. Este algoritmo não se baseia no formato do histograma mas sim na coesão intra classe e na separabilidade das classes.

Uma desvantagem de utilizar este algoritmo é a influência da média de todo os pixels da imagem. Isto pode fazer com que o limiar ótimo para a página toda não seja o mesmo que o de dentro de uma janela.

8.2 Implementação

Todos os experimentos foram realizados utilizando scripts ruby e python para automatizar a tarefa. Também utilizamos a biblioteca ImageMagick e bibliotecas para manipulação de XML.

O processo todo consistiu em baixar a base de dados do PRImA, processá-las gerando as entradas para o TRIOS, construir os operadores, aplicá-los, consensualizar e extrair métricas.

Este trabalho tem como objetivo provar um conceito: avaliar a aplicabilidade de operadores morfológicos à segmentação de página. Logo o código gerado não é destino a aplicações comerciais.

- `sample_generator.rb` produz todos os conjuntos de imagens para treinamento e teste, ou seja, ele transforma todas as imagens e XMLs da base de dados original em entradas apropriadas para as ferramentas que utilizamos.

Tabela 31: Estágios da execução do algoritmo de Otsu.

original	25%	37,5%	50%	62,5%	75%	87,5%
w_B	0.1931	0.4015	0.4179	0.4532	0.5479	0.6969
μ_B	34.00	51.64	53.61	61.37	83.08	112.56
σ_B^2	7.27	288.58	372.94	1054.08	3128.03	5656.37
w_F	0.80	0.59	0.58	0.54	0.45	0.30
μ_F	184.87	225.55	229.03	233.95	243.79	255.0
σ_F^2	5799.89	1409.01	1006.11	673.10	255.43	0.0
σ_w^2	21495165144	967521582	512693389	595067475	4269882800	17660993341

- imgset_training_gen.rb gera e aplica todos os 200 operadores utilizados nos experimentos. Utiliza os executáveis trios_build e trios_apply fornecidos pelo TRIOS.
- measures.py extrai todas as estatísticas apresentadas na seção 6.
- consensus.py faz a união das segmentações realizadas pelo apply_all.rb
- merge.py cria imagens segmentadas coloridas para visualização.

Referências

- [1] Rangachar Kasturi, Lawrence O’Gorman, and Venu Govindaraju. Document image analysis: A primer. *Sadhana*, 27(1):3–22, 2002.
- [2] Nina Sumiko Tomita. Programação automática de máquinas morfológicas binárias baseada em aprendizado pac. Master’s thesis, Departamento de Ciência da Computação, Instituto de Matemática e Estatística, April 1996. This work has been supported by ProTeM-CC/CNPq throught the AnIMoMat project, contract 680067/94-9.
- [3] Igor dos Santos Montagner, Roberto Hirata Jr, and Nina ST Hirata. Trios-an open source toolbox for training image operators from samples.
- [4] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher. A realistic dataset for performance evaluation of document layout analysis, 2009.
- [5] Stefan Pletschacher and Apostolos Antonacopoulos. The page (page analysis and ground-truth elements) format framework. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR2010), Istanbul, Turkey, August 23-26, 2010*, pages 257–260. IEEE-CS, 2010.
- [6] C Clausner, S Pletschacher, and A Antonacopoulos. Aletheia-an advanced document layout and text ground-truthing system for production environments. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 48–52. IEEE, 2011.
- [7] A. Antonacopoulos and D. Bridson. Performance analysis framework for layout analysis methods. *Document Analysis and Recognition, International Conference on*, 2:1258–1262, 2007.
- [8] 10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009. IEEE Computer Society, 2009.

- [9] Michael A. Moll and Henry S. Baird. Document content inventory and retrieval. In *In Proc., IAPR 9th Int'l Conf. on Document Analysis and Recognition (ICDAR2007*, 2007.
- [10] Mehmet Sezgin and Bülent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168, January 2004.
- [11] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979.