Revista Abstração

Revista Abstração

PET Computação UFSC

Abril - 2018

Editorial

Abstrair é olhar através de outro ponto de vista, ver o mundo / ideias e conceitos sob um certo aspecto ou de algum ângulo em particular. Todas as ciências são diferenciadas pela sua abstração.

Como conceito fundamental na Ciência da Computação, abstração é a ideia de encapsulamento ou ocultamento de detalhes. É uma operação intelectual em que um objeto de reflexão é isolado de fatores que comumente lhe estão relacionados na realidade.

A revista Abstração é uma publicação semestral do Departamento de Informática e Estatística (INE) organizada pelo Programa de Educação Tutorial da Computação (PET Computação). O objetivo da Revista Abstração é ser um espaço de apresentação de ideias e resultados de pesquisas, ambos no âmbito dos Trabalhos de Conclusão de Curso (TCCs) e Iniciação Científica (IC) desenvolvidos neste departamento.

Entre as motivações para sua publicação estão a necessidade de publicização das pesquisas realizadas pelos acadêmicos dos cursos de graduação (Ciências da Computação e Sistemas de Informação) e para dar visibilidade ao panorama de pesquisas realizadas por professores e alunos nos diferentes laboratórios e projetos de pesquisa do INE. Serve também como um norte para os estudantes que buscam compreender melhor determinados temas de pesquisa.

Esta edição da revista abstração é composta por uma seleção dos artigos dos TCCs realizados nos cursos de Ciências da Computação e Sistemas de Informação da UFSC durante 2015 e 2016, visando incluir pelo menos um artigo de cada uma das seguintes linhas de pesquisa: bancos de dados, computação paralela e distribuída, engenharia de software, inteligência computacional, redes de computadores, segurança em sistemas computacionais e sistemas embarcados. Os artigos que tinham o formato SBC (formato solicitado pelos cursos para a submissão dos artigos de TCC) e contiam entre 5 e 15 páginas foram enviados para professores do departamento que trabalham na respectiva área de pesquisa relacionada ao artigo. Agradecemos a todos os professores que participaram deste processo de seleção.

Nos empenhamos em estabelecer uma avaliação nos mesmos moldes dos eventos nacionais, elencando itens de avaliação como: originalidade, qualidade do texto, relevância e mérito técnico do artigo, a confiança do revisor na área de pesquisa do artigo, e se recomendaria a inserção do artigo na revista.

Tais critérios foram apresentados aos professores revisores com opções entre muito ruim, ruim, regular, bom ou muito bom; o critério de confiança na avaliação foi respondido com baixa, regular ou alta, e o critério de recomendação foi dividade em rejeição forte, rejeição fraca, neutro, aceitação fraca ou aceitação forte.

Boa leitura a todos!

Sumário

Algoritmo de Eliminações Sucessivas baseado em Soma das Diferenças Transformadas Absolutas (Luiz Henrique De L. Cancellier) - Propõe dois critérios de eliminação de cálculos da métrica de similaridade SATD.

Arquiteturas Energeticamente Eficientes para SATD com Tamanho de Bloco Variável no HEVC (André Beims Bräscher) - Apresenta duas alternativas de arquiteturas para o cálculo de SATD usando Linear Buffer, comparadas a uma arquitetura com Transpose Buffer.

Proposta de uma Plataforma de Sistemas Multiagentes para Suportar a Gerência Autônoma de Recursos em Ambientes de Computação em Nuvem (Alexandre de Limas Santana, Lucas Berri Cristofolini) - se propõe a adaptar uma ferramenta de orquestração existente, de modo a utilizar uma plataforma de sistemas multiagentes para possibilitar que agentes inteligentes realizem a analise, o planeja-mento e a gerência em ambientes de computação em nuvem de forma independente e autônoma.

Um Sistema Para Geração Automática de Questões (Luís Gustavo T. Cordeiro) - Descreve brevemente algumas técnicas básicas de processamento de linguagem natural que podem ser utilizadas em sistemas para a análise de texto e geração de questões através de sentenças em determinada linguagem.

tCALC- Agrupamento de Currículos Lattes por Afinidade de Áreas de Conhecimento Considerando Temporalidade (Jaime Mendes da Silva) - Estende trabalhos realizados anteriormente ao analisar o impacto de qualidade e performance causado pela consideração do fator tempo no processo de agrupamento dos currículos.

Utilização de QoC para melhorar o cenário experimental de sensores biomédicos para suporte às aplicações móveis distribuídas (Pedro José Campos, Mario Antonio Ribeiro Dantas, Eduardo Camilo Inacio) - Descreve como utilizar Qualidade de Contexto (QoC) em aplicações móveis distribuídas, mais especificamente em Ambient Assisted Living (AAL), e a partir de parâmetros de contexto encontrar problemas em determinadas situações e com isso poder melhorar cenário dos sensores envolvidos com o ambiente.

Redes Neurais Convolucionais de Profundidade para Reconhecimento de Textos em Imagens de CAPTCHA (Vitor Arins Pinto) - O objetivo do trabalho proposto neste artigo é realizar o reconhecimento de texto em imagens de CAPTCHA atraves da aplicação de redes neurais convolucionais.

Algoritmo de Eliminações Sucessivas baseado em Soma das Diferenças Transformadas Absolutas

Luiz Henrique De L. Cancellier[1]

[1]Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

l.h.cancellier@grad.ufsc.br

Resumo. *A Estimação de Movimento - Motion Estimation (ME) é a etapa computacionalmente mais intensiva da codificação de vídeo digital. Ela consiste na busca por um bloco que minimize uma métrica de similaridade para ser tomado como referência. Baseando-se no SEA, este trabalho propõe dois critérios de eliminação de cálculos da métrica de similaridade SATD. O primeiro critério, chamado de AFD, teve bom resultados na ME inteira, com 23% de candidatos eliminados no pior caso, enquanto segundo critério,chamado de MSATD, eliminou 24% e 69% na ME fracionária e inteira, respectivamente.*

1. Introdução

Um vídeo é uma sequência de imagens, chamadas de quadros, apresentadas rapidamente no tempo. O armazenamento de todos os quadros de um vídeo não codificado é proibitivo devido ao grande volume de dados usados para sua representação [Agostini 2007]. Tal volume torna-se ainda maior conforme as resoluções aumentam. Assim, ao adotarem-se resoluções como 2160p (3840×2160 pixels) e 4320p (7680×4320), a codificação de vídeos se torna mandatória.

Se por um lado a codificação de vídeo é necessária para reduzir o volume de dados, por outro este processo é computacionalmente intensivo [Bossen et al. 2012]. Dessa forma, é necessário otimizar a codificação visando atingir uma taxa de compressão aceitável, controlando-se as perdas na qualidade do vídeo e reduzindo o tempo gasto ao longo do processo.

O fluxo de codificação é apresentado na Figura 1. Cada quadro não codificado, que será chamado de quadro original, é particionado em pequenos blocos, chamados de blocos originais (“Ori”). Para cada bloco original, a etapa de predição realiza uma busca entre os blocos candidatos. O candidato que mais se assemelha ao original será tomado como referência (“Ref”).

A diferença entre o bloco original e a referência resulta no resíduo (“Res”), que será transformado (T) e quantizado (Q). Na etapa de quantização os valores do bloco transformado serão reduzidos de acordo com o Parâmetro de Quantização - *Quantization Parameter* (QP) definido. Quanto maior o QP, mais os coeficientes serão reduzidos, ou até mesmo zerados. Dessa forma, menos informação será usada para representar o bloco e também será pior a qualidade do vídeo codificado. O bloco resultante de todo esse processo ainda será codificado por entropia, onde o dado é comprimido sem perdas.

Um bloco codificado é reconstruído com a quantização inversa (Q^{-1}), transformação inversa (T^{-1}) e o resultado é somado com o bloco de referência. Essa

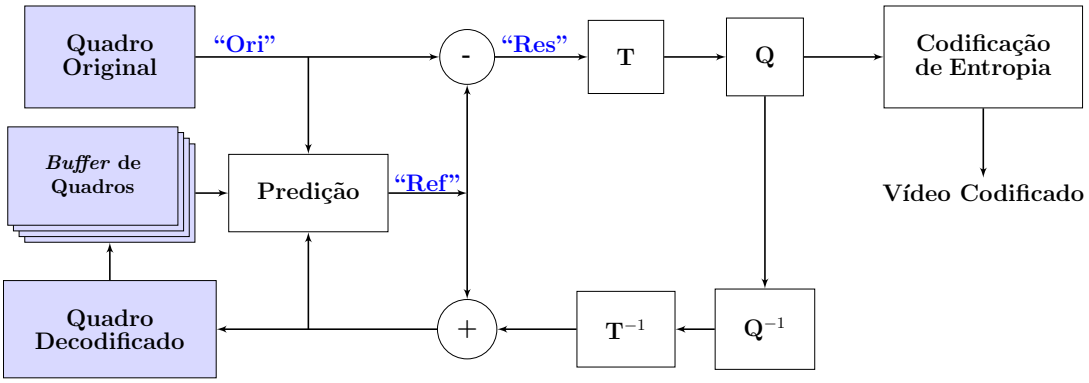


Figura 1. Diagrama simplificado do fluxo de codificação. Adaptado de: [Cancellier et al. 2015].

reconstrução irá realimentar a etapa de predição, adicionando ao *buffer* mais um candidato que será usado para codificar os próximos blocos [Richardson 2002].

A etapa de predição explora a redundância de informação entre blocos. Quando áreas de um mesmo quadro são muito semelhantes, tal característica é conhecida como redundância espacial. Também existe o conceito de redundância temporal, que ocorre quando quadros temporalmente próximos possuem áreas semelhantes [Shi and Sun 1999].

Para explorar a redundância temporal, é executado o processo de ME. O bloco referência é aquele que, dentre os possíveis candidatos em uma área de busca restrita e predeterminada no quadro candidato já codificado, apresenta maior semelhança com o original. Para determinar o quão semelhantes são dois blocos é usado um critério chamado de métrica de similaridade. A ME se resume então à busca pelo candidato que minimize o resultado da métrica de similaridade adotada.

O algoritmo mais conhecido e trivial para ME é a Busca Completa - *Fullsearch* (FS), que aplica a métrica de similaridade para todos os blocos candidatos da janela de busca. A busca intensiva na FS fornece resultado ótimo, porém seu custo computacional é muito elevado. Em função disso, diversos algoritmos rápidos foram propostos com objetivo de reduzir o número de candidatos avaliados, trocando o aumento de desempenho por resultados subótimos [Huang et al. 2006]. Por outro lado, existem algoritmos como o Algoritmo de Eliminações Sucessivas - *Successive Elimination Algorithm* (SEA) [Li and Salari 1995], que utilizam propriedades da métrica de similaridade para eliminar candidatos impossíveis mediante o uso de cálculos mais simples. Um candidato é dito impossível quando sabe-se antes do cálculo da métrica de similaridade que ele não será tomado como referência. Dessa forma, é possível aplicar essas técnicas de eliminação para acelerar o FS e ainda manter os resultados ótimos.

As métricas de similaridade mais usadas em codificação de vídeo são a Soma das Diferenças Absolutas - *Sum of Absolute Differences* (SAD), a Soma das Diferenças Quadráticas - *Sum of Squared Differences* (SSD) e a Soma das Diferenças Transformadas Absolutas - *Sum of Absolute Transformed Differences* (SATD) [Richardson 2003]. Por permitir resultados aceitáveis de qualidade de codificação e ainda ter cálculo bastante simples, a métrica SAD é a mais utilizada.

A SATD, por sua vez, apresenta cálculo mais complexo que a SAD, computando uma transformada Hadamard sobre a matriz de diferenças. Com relação à eficiência de codificação, os resultados obtidos com o uso da SATD são melhores que as demais métricas citadas [Dominges et al. 2011]. Um dos principais fatores que permitem à SATD apresentar melhor eficiência de codificação é que a transformada utilizada em seu cálculo está correlacionada com a Transformada Discreta dos Cossenos - *Discrete Cosine Transform* (DCT) [Zhu and Xiong 2009]. A DCT é aplicada no processo de transformação, que ocorre após a ME, sobre a matriz de diferenças entre o bloco original e o referência. Desta forma, escolher a referência com base numa transformada que se aproxima da DCT reduz o erro gerado [Dominges et al. 2011].

Apesar de apresentar melhores resultados na eficiência de codificação, o uso da SATD em um processo de ME é proibitivo por conta do alto custo para computar a transformada Hadamard. Para reduzir tal custo, serão desenvolvidas técnicas baseadas no SEA para evitar a computação completa da transformada.

O restante deste trabalho está organizado da seguinte forma. Na Sessão 2 será apresentado em detalhes a métrica SATD. Na Sessão 3 serão formalizados os dois critérios de eliminação de candidatos para o cálculo de SATD e os resultados obtidos serão apresentados e avaliados na Sessão 4. Por fim, as conclusões serão apresentadas na Sessão 5.

2. Soma das Diferenças Transformadas Absolutas

Para definir o cálculo da métrica de similaridade SATD, inicialmente é necessário computar a diferença entre os blocos original e candidato, como apresenta a Equação 1. Após computar as diferenças, é preciso aplicar uma transformação sobre ela. Para isso, a SATD usa a transformada Hadamard (Equação 2), que faz uso das matrizes de Hadamard (H).

$$D_{2^n \times 2^n} = Ori_{2^n \times 2^n} - Can_{2^n \times 2^n} \tag{1}$$

$$T(D_{2^n \times 2^n}) = H_{2^n \times 2^n} \times D_{2^n \times 2^n} \times H_{2^n \times 2^n} \tag{2}$$

A matriz Hadamard pode ser obtida recursivamente, como apresenta a Equação 3. O símbolo "$\otimes$" representa o produto de Kronecker [Weisstein 2009], que para esse caso pode ser expandido como mostra a Equação 4. Esta última equação será importante para provar as técnicas de eliminação propostas.

$$H_{2^n \times 2^n} = \begin{cases} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & \text{se } n = 1 \\ \\ \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes H_{2^{n-1} \times 2^{n-1}} & \text{se } n > 1 \end{cases} \tag{3}$$

$$H_{2^n \times 2^n} = \begin{bmatrix} 1 \times H_{2^{n-1} \times 2^{n-1}} & 1 \times H_{2^{n-1} \times 2^{n-1}} \\ 1 \times H_{2^{n-1} \times 2^{n-1}} & -1 \times H_{2^{n-1} \times 2^{n-1}} \end{bmatrix} \tag{4}$$

Por fim, tomando os elementos $t_{i,j} \in T(D_{2^n \times 2^n})$, a SATD pode ser definida como mostra a Equação 5.

$$SATD(Ori_{2^n \times 2^n}, Can_{2^n \times 2^n}) = \frac{1}{2^{n-1}} \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} |t_{i,j}| \tag{5}$$

3. Critérios de Eliminação

O SEA se fundamenta na propriedade de subaditividade do módulo (Equação 6) para propor um critério mais simples de similaridade para eliminar candidatos impossíveis, ou seja, candidatos que garantidamente não serão tomados como referência.

$$|a| + |b| \geq |a + b| \tag{6}$$

Aplicando a propriedade de subaditividade do módulo na SATD, obtém-se a Equação 7. Da forma como foi apresentada, essa última equação mostra um critério de eliminação que ainda necessita da transformação dos elementos, que é justamente a operação que torna a SATD tão custosa. Para evitar a transformação completa, foram propostos dois critérios de eliminação que também baseadas na propriedade de subaditividade do módulo.

$$\frac{1}{2^{n-1}} \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} |t_{i,j}| \geq \frac{1}{2^{n-1}} | \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} t_{i,j}| \tag{7}$$

3.1. Primeira Diferença Absoluta

Partindo da soma dos elementos $t_{i,j}$, deseja-se definir um critério de eliminação que possa ser computado de forma mais eficiente. Para isso, será tomada como hipótese a Equação 8, onde nenhuma transformação precisa ser computada. Para provar esse resultado, será usado o método de prova por indução matemática.

$$\frac{1}{2^{n-1}} | \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} t_{i,j}| = \frac{(2^n)^2}{2^{n-1}} |d_{1,1}| \tag{8}$$

Inicialmente será demonstrado que a propriedade é válida para o passo base. Tomando n, tal que $n = 1$, a transformada é aplicada sobre D como segue:

$$\begin{aligned} T(D_{2\times2}) &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \times \begin{bmatrix} d_{1,1} & d_{1,2} \\ d_{2,1} & d_{2,2} \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} (+d_{1,1} + d_{2,1} + d_{1,2} + d_{2,2}) & (+d_{1,1} + d_{2,1} - d_{1,2} - d_{2,2}) \\ (+d_{1,1} - d_{2,1} + d_{1,2} - d_{2,2}) & (+d_{1,1} - d_{2,1} - d_{1,2} + d_{2,2}) \end{bmatrix} \end{aligned} \tag{9}$$

Ao manipular $T(D_{2\times2})$ na Equação 8, como apresentado em 10, é possível observar que todos os elementos, exceto $d_{1,1}$, se anulam, resultando na Equação 11. Constata-se então que a propriedade definida inicialmente é de fato válida para o caso base.

$$\frac{1}{2^0}|\sum_{i=1}^{2^1}\sum_{j=1}^{2^1} t_{i,j}| = |(+d_{1,1}+d_{2,1}+d_{1,2}+d_{2,2})+(+d_{1,1}+d_{2,1}-d_{1,2}-d_{2,2}) \\ +(+d_{1,1}-d_{2,1}+d_{1,2}-d_{2,2})+(+d_{1,1}-d_{2,1}-d_{1,2}+d_{2,2})| \tag{10}$$

$$|\sum_{i=1}^{2^1}\sum_{j=1}^{2^1} t_{i,j}| = 4 \times |d_{1,1}| \tag{11}$$

No passo indutivo será assumido que, para um valor k arbitrário, a propriedade definida na Equação 8 é válida. Por fim, é necessário provar que a propriedade também se aplica a $n = k + 1$. As matrizes envolvidas no cálculo serão particionadas como mostra a Equação 12 e as multiplicações da transformada serão feitas por particionamento. Essa multiplicação é feita de forma semelhante à usual, porém os fatores são partições das matrizes[Rowland 2006]. Após aplicar a transformação, obtém-se a Equação 13.

$$T(D_{2^n\times 2^n}) = H_{2^{k+1}\times 2^{k+1}} \times D_{2^{k+1}\times 2^{k+1}} \times H_{2^{k+1}\times 2^{k+1}} \\ = \left[\begin{array}{c|c} H_{2^k\times 2^k} & H_{2^k\times 2^k} \\ \hline H_{2^k\times 2^k} & -H_{2^k\times 2^k} \end{array}\right] \times \left[\begin{array}{c|c} (D_{2^k\times 2^k})_{1,1} & (D_{2^k\times 2^k})_{1,2} \\ \hline (D_{2^k\times 2^k})_{2,1} & (D_{2^k\times 2^k})_{2,2} \end{array}\right] \times \left[\begin{array}{c|c} H_{2^k\times 2^k} & H_{2^k\times 2^k} \\ \hline H_{2^k\times 2^k} & -H_{2^k\times 2^k} \end{array}\right] \tag{12}$$

$$\left[\begin{array}{c|c} T(D_{1,1})+T(D_{2,1})+T(D_{1,2})+T(D_{2,2}) & T(D_{1,1})+T(D_{2,1})-T(D_{1,2})-T(D_{2,2}) \\ \hline T(D_{1,1})-T(D_{2,1})+T(D_{1,2})-T(D_{2,2}) & T(D_{1,1})-T(D_{2,1})-T(D_{1,2})+T(D_{2,2}) \end{array}\right] \tag{13}$$

É interessante observar que o padrão de sinais do caso base se repete entre as partições da matriz transformada. Ao somar as partições, todas se anulam exceto $T(D_{1,1})$. O resultado do módulo da soma dos elementos de uma matriz de diferenças transformadas de tamanho $2^k \times 2^k$ foi assumido no início do passo indutivo e será substituído em 7, como mostra a Equação 14. Como $n = k + 1$, essa última equação pode ser reescrita de forma que seja possível concluir que a hipótese é válida.

$$\frac{1}{2^k}|\sum_{i=1}^{2^n}\sum_{j=1}^{2^n} t_{i,j}| = \frac{(2^{k+1})^2}{2^k}|d_{1,1}| \tag{14}$$

Utilizando a Equação 8, cálculos de SATD podem ser poupados usando um critério de seleção baseado apenas no primeiro elemento da matriz de diferenças. Esse critério será chamado de Primeira Diferença Absoluta - *Absolute First Difference* (AFD) e é definido na Equação 15. Como na ME deseja-se encontrar o candidato que minimiza o valor de SATD, se um bloco sendo avaliado apresentar AFD maior ou igual à SATD do melhor candidato selecionado até aquele momento, esse bloco pode ser descartado. Isso é possível pois garantidamente a SATD do bloco descartado também será maior que

$$AFD(Ori_{2^n\times 2^n}, Can_{2^n\times 2^n}) = \frac{(2^n)^2}{2^{n-1}}|d_{1,1}| \tag{15}$$

3.2. Eliminações Sucessivas em Níveis

Uma técnica mais genérica de eliminação faz sucessivos particionamentos da matriz de diferenças para eliminar um número maior de candidatos enquanto se aproxima do cálculo final da métrica de similaridade. Esse critério é definido para computar o Algoritmo de Eliminações Sucessivas em Níveis - *Multilevel Successive Elimination Algorithm* (MSEA) [Gao et al. 2000] e também pode ser aplicado à SATD.

No al ser dividido em l níveis, tal que $0 \leq l < n$ para blocos quadrados de tamanho 2^n. A métrica $MSATD_l$ é definida como mostra a Equação 16, onde (o, p) indexa um elemento da partição (i, j) da matriz das diferenças transformadas.

$$\mathrm{MSATD}_l = \frac{1}{2^{n-1}} \sum_{i=1}^{2^l} \sum_{j=1}^{2^l} | \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \Big(T(D_{i,j}) \Big)_{o,p} | \tag{16}$$

O cálculo do nível $l = 0$ é simplesmente o AFD. Para demonstrar o cálculo do nível 1 será usado como base as diferenças transformadas para blocos de tamanho 2^n, onde $n = k+1$, que é dada pela Equação 13. Neste caso a MSATD_1 é dada pela equação:

$$\begin{aligned}
\mathrm{MSATD}_{l=1} = & \frac{1}{2^k} | \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \Big(+ T(D_{1,1}) + T(D_{2,1}) + T(D_{1,2}) + T(D_{2,2}) \Big)_{o,p} | \\
& + \frac{1}{2^k} | \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \Big(+ T(D_{1,1}) + T(D_{2,1}) - T(D_{1,2}) - T(D_{2,2}) \Big)_{o,p} | \\
& + \frac{1}{2^k} | \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \Big(+ T(D_{1,1}) - T(D_{2,1}) + T(D_{1,2}) - T(D_{2,2}) \Big)_{o,p} | \\
& + \frac{1}{2^k} | \sum_{o=1}^{2^n/2^l} \sum_{p=1}^{2^n/2^l} \Big(+ T(D_{1,1}) - T(D_{2,1}) - T(D_{1,2}) + T(D_{2,2}) \Big)_{o,p} |
\end{aligned} \tag{17}$$

O somatório dos elementos de $T(D_{i,j})$ é conhecido e corresponde ao AFD extraído da matriz $D_{i,j}$. Para fins de simplificação, os elementos $d_{i,j}$ da Equação 18 não serão aqueles referentes a matriz D, mas sim ao elemento $(1, 1)$ relativo à partição (i, j) da matriz D. Reescrevendo a Equação 17 obtém-se:

$$\begin{aligned}
\mathrm{MSATD}_{l=1} = \frac{(2^{n-l})^2}{2^k} (& |(d_{1,1} + d_{2,1} + d_{1,2} + d_{2,2})| + |(d_{1,1} + d_{2,1} - d_{1,2} - d_{2,2})| \\
& + |(d_{1,1} - d_{2,1} + d_{1,2} - d_{2,2})| + |(d_{1,1} - d_{2,1} - d_{1,2} + d_{2,2})|)
\end{aligned} \tag{18}$$

A MSATD_1 é semelhante ao cálculo da $SATD_{2\times2}$ e isso permite que os elementos que compõem as partições do cálculo da MSATD_1 sejam escritas na forma de uma transformada Hadamard, como mostra a Equação 19. Para fins de simplificação, a matriz formada pelos elementos na posição $(1, 1)$ relativa a cada partição será chamada de

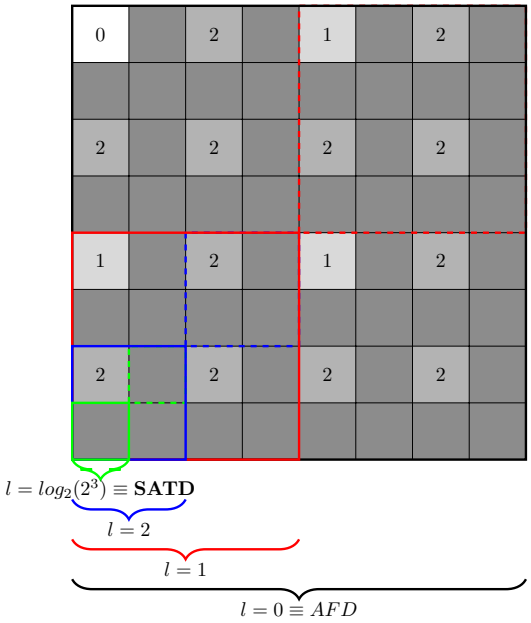


Figura 2. Níveis de particionamento de um bloco 8×8 no algoritmo MSEA-SATD. As posições marcadas com valores menores ou iguais a l compõem a matriz FD_M usada na métrica MSATD_l.

FD_M e as constantes do somatório serão colocadas em evidência. Apesar de ser apresentada apenas para o nível um de particionamento, a Equação 19 é genérica o suficiente para ser aplicada para n e l quaisquer.

$$\mathrm{MSATD}_l = \frac{(2^{n-l})^2}{2^{n-1}} \sum_{i=1}^{2^l} \sum_{j=1}^{2^l} \left| \left(H_{2^l \times 2^l} \times FD\_M_{2^l \times 2^l} \times H_{2^l \times 2^l} \right)_{i,j} \right| \tag{19}$$

A Figura 2 ilustra como é feito o particionamento e seleção de elementos para a transformação em cada nível É importante observar que a computação de MSATD_l tende a se tornar mais precisa e tão complexa quanto a SATD à medida que l aumenta. Entretanto, mesmo computando diretamente o último nível do critério de eliminação, ainda são usadas apenas 25% de operações em comparação com o cálculo completo da SATD.

4. Resultados

As técnicas propostas foram implementadas no código de referência do padrão de Codificação de Vídeo de Alta Eficiência - *High Efficiency Video Coding* (HEVC) [Sullivan et al. 2012, ITU-T 2013]. Os testes foram feitos com base na Condições Comuns de Teste - *Common Test Conditions* (CTC) [Bossen 2012], usando o arquivo de configuração "*Low-delay P-High efficiency*". Exceto por dois vídeos com 10 bits por pixel, todas as outras 22 sequências de vídeo foram executadas usando os quatro QPs indicados (22, 27, 32, 37). Neste artigo, serão apresentados apenas os resultados de pior caso, que ocorreram com o uso do QP 22.

Foram feitos experimentos nas duas etapas da ME. A primeira etapa, chamada de Estimação de Movimento Inteira - *Integer Motion Estimation* (IME), consiste numa busca pelos candidatos de um quadro já codificado. A segunda etapa, chamada de Estimação de Movimento Fracionária - *Fractional Motion Estimation* (FME), consiste em um refinamento onde são gerados e avaliados novos candidatos gerados a partir do candidato selecionado na IME.

4.1. Estimação de Movimento Fracionária

A Figura 4.1 apresenta o percentual acumulado de eliminações em cada nível do algoritmo MSEA-SATD para cada QP. Os gráficos avaliam apenas candidatos que não tiveram sua transformada completa calculada. Aqueles compostos por blocos 4×4 são considerados até o nível 1 enquanto as eliminações no nível 2 consideram apenas candidatos compostos por blocos 8×8.

É possível perceber que, em QPs mais baixos, a técnica AFD (Nível 0) apresenta um baixo percentual de eliminações. À medida que o QP aumenta, a técnica de eliminação se torna mais efetiva e também mais imprevisível, uma vez que o percentual de eliminações varia para cada vídeo avaliado.

O método MSEA-SATD apresenta um bom resultado no último nível de eliminação. No contexto da FME, onde os candidatos possuem valores muito próximos, o último nível tende a ser o único com algum impacto significativo. Aplicando diretamente o nível 2, ainda seriam eliminados, no pior caso, aproximadamente 25% dos candidatos.

Em duas sequências de vídeo o nível 0 apresentou um percentual de eliminação acima de 50%. Tanto o "*SlideEditing*" quanto o "*SlideShow*" são vídeos atípicos, com pouca movimentação em longas sequências de quadros. Pelos resultados obtidos, há um indício de que a métrica AFD é uma técnica bastante efetiva para esse nicho de aplicação.

4.2. Estimação de Movimento Inteira

Na IME, diferente do resultado observado na FME, o uso da AFD (nível 0) apresenta uma boa taxa de eliminação. Isso ocorre pois são avaliados candidatos pouco similares, onde eventualmente um bloco com baixo valor de SATD é encontrado e os outros serão eliminados já nos primeiros níveis. No pior caso, aproximadamente 22,75% dos candidatos foram eliminados, valor considerável para uma métrica que usa apenas três operações aritméticas.

O uso do algoritmo MSEA-SATD também é bastante efetivo. No último nível as sequências convergem para uma taxa de eliminação acima de 65%. Novamente observa-se a possibilidade de computar apenas o terceiro nível como critério de eliminação.

5. Conclusão

Neste trabalho foram apresentadas duas técnicas de eliminação de candidatos na ME que faz uso da métricas SATD na busca pelo bloco de referência. A primeira técnica, inspirada no SEA, deu origem ao critério de eliminação chamado de AFD, que possui cálculo bastante simples e se provou eficiente na IME, com aproximadamente 23% de eliminações no pior cenário.

Foi demonstrado que a técnica usada no MSEA também pode ser aplicada a métrica SATD. Isso resultou na definição do critério de eliminação em níveis MSATD_l.

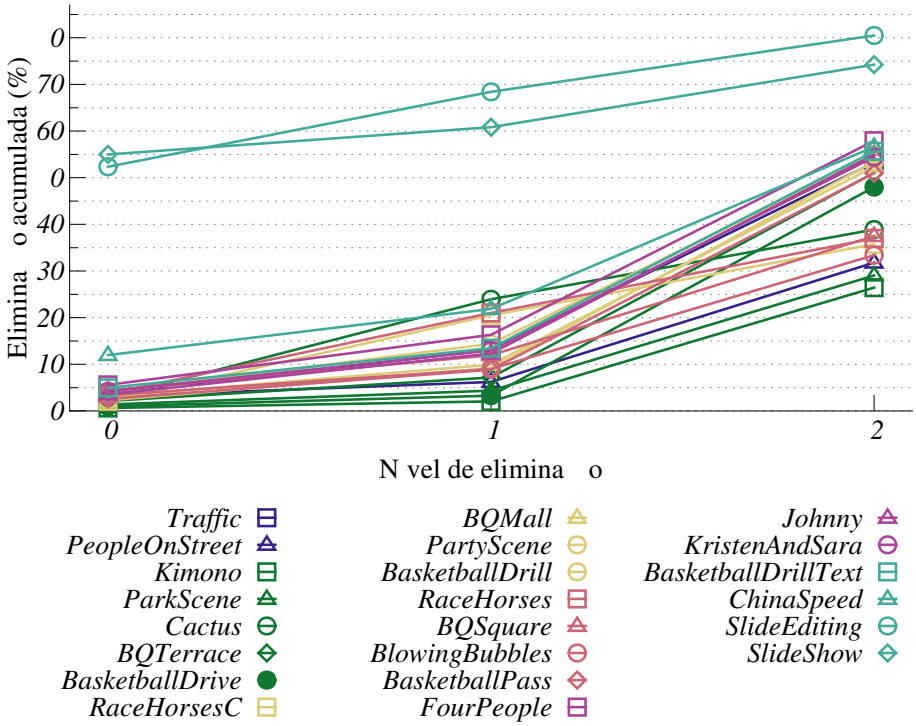


Figura 3. Percentual acumulado de eliminações de candidatos em cada nível na FME usando o QP 22.

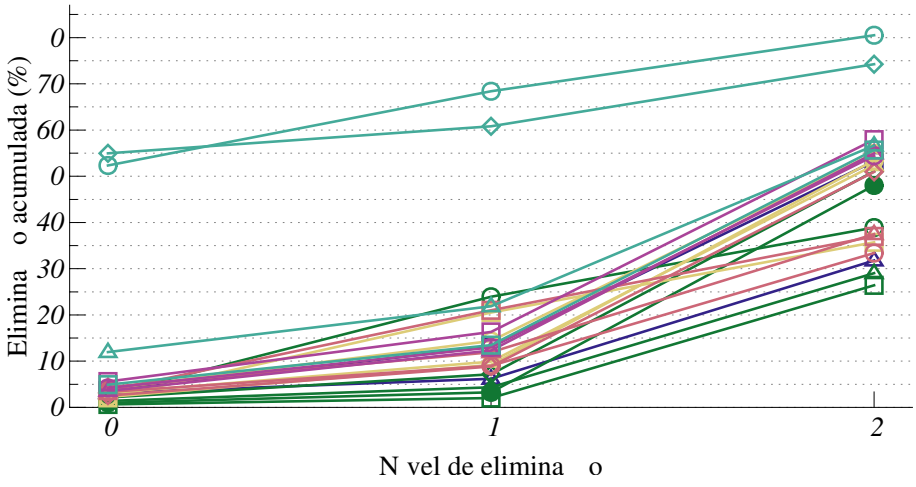


Figura 4. Percentual acumulado de eliminações de candidatos em cada nível na IME usando o QP 22.

Esse critério demonstrou ser bastante eficiente, principalmente no terceiro nível. Ele eliminou, no pior caso, aproximadamente 24% e 69% de candidatos na FME e IME, respectivamente. Além de apresentar uma boa taxa de eliminações no terceiro nível, ele computa apenas 25% das operações necessárias para calcular a SATD.

Referências

Agostini, L. V. (2007). Desenvolvimento de arquiteturas de alto desempenho dedicadas à compressão de vídeo segundo o padrão h.264/avc. Tese de doutorado., UFRJ.

Bossen, F. (2012). Common test conditions and software reference configurations. Document JCTVC-K1100, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Shanghai.

Bossen, F., Bross, B., Suhring, K., and Flynn, D. (2012). Hevc complexity and implementation analysis. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12):1685–1696.

Cancellier, L. H., Bräscher, A. B., Seidel, Ismael, G. J. L., and Agostini, L. (2015). Exploring optimized hadamard methods to design energy-efficient satd architectures. In *Journal of Integrated Circuits and Systems*, JICS, pages 113–112. SBC.

Dominges, Jr, J. S., Possani, V. N., Silveira, D. S., da Rosa Jr, L. S., and Agostini, L. V. (2011). High throughput 4x4 and 8x8 SATD similarity criteria architectures for video coding applications. In *2011 VII Designer Forum (DF)*, page 115. Citeseer.

Gao, X., Duanmu, C., and Zou, C. (2000). A multilevel successive elimination algorithm for block matching motion estimation. *Image Processing, IEEE Transactions on*, 9(3):501–504.

Huang, Y.-W., Chen, C.-Y., Tsai, C.-H., Shen, C.-F., and Chen, L.-G. (2006). Survey on block matching motion estimation algorithms and architectures with new results. *J. VLSI Signal Process. Syst.*, 42(3):297–320.

ITU-T (2013). Recommendation itu-t h.265: High efficiency video coding. Recommendation H.265, International Telecommunication Union, Genebra.

Li, W. and Salari, E. (1995). Successive elimination algorithm for motion estimation. *IEEE Trans. on Image Processing*, 4(1):105–107.

Richardson, I. E. G. (2002). *Video codec design: developing image and video compression systems*. John Wiley and Sons.

Richardson, I. E. G. (2003). *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons Inc.

Rowland, T. (2006). Block matrix.

Shi, Y. and Sun, H. (1999). *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards*. Image Processing Series. Taylor & Francis.

Sullivan, G., Ohm, J., Han, W.-J., and Wiegand, T. (2012). Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668.

Weisstein, E. W. (2009). Kronecker product.

Zhu, C. and Xiong, B. (2009). Transform-exempted calculation of sum of absolute hadamard transformed differences. *IEEE Trans. Circuits Syst. Video Technol.*, 19(8):1183–1188.

Arquiteturas Energeticamente E cientes para SATD com Tamanho de Bloco Variável no HEVC

André Beims Bräscher

Departamento de Informatica e Estatística – Universidade Federal de Santa Catarina
Florianópolis – SC – Brazil
andre.brascher@grad.ufsc.br

Abstract. *The inter-frame prediction is the video coding step that provides the greatest compression. For such step the calculation of a block similarity metric is necessary. A good alternative for such calculation is the SATD, although it presents quadratic energy consumption in relation to the number of processed pixels. A possible alternative is to substitute the usual Transpose Bu er (TB) structure by a Linear Bu er (LB), re-calculating partial results. In this work, two alternatives for SATD calculation using LB are presented and compared to an architecture that uses TB. The proposed architectures achieved great area reduction.*

Resumo. *A predição inter-quadros é a etapa da codi cação de vídeos responsável pela maior compressão. Para tanto, requer-se o cálculo de uma métrica de similaridade entre blocos. Uma métrica utilizada é a SATD porém a energia necessária para tal métrica tende a aumentar quadraticamente ao número de píxeis processados. Uma possível alternativa é a substituição da usual estrutura Transpose Bu er (TB) por Linear Bu er (LB), realizando recálculo de parcelas. Este trabalho apresenta duas alternativas de arquiteturas para o cálculo de SATD usando LB, comparadas a uma arquitetura com TB. As arquiteturas propostas apresentaram grande redução de área.*

1. Introdução

Um vídeo, em formato *raw* (ou seja, sem nenhum tipo de compressão), necessita de um grande volume de dados [RICHARDSON, 2003]. Portanto, a compressão de vídeo torna-se fundamental, em especial para as resoluções atualmente utilizadas (*e.g. Full-HD*, *Quad-HD*). Apesar de não possuir complexidade assintótica alta, a compressão de vídeo é uma atividade computacionalmente intensiva [SEIDEL, 2014]. Além disso, o surgimento do padrão de Codi cação de Vídeo de Alta E ciência - *High E ciency Video Coding* (HEVC) [SULLIVAN et al., 2012] como sucessor do H.264 [ITU-T, 2003], [SULLIVAN, 2005], resultou em melhorias de aproximadamente 50% em e ciência de codi cação [SULLIVAN et al., 2012]. Porém tal melhoria vem a um custo: o Modelo do HEVC - *HEVC Model* (HM) [JCT-VC, 2013] é pelo menos 4 vezes mais

lento em comparação ao código de referência do H.264, Modelo Conjunto - *Joint Model* (JM) [JVT, 2011], em con gurações similares [BOSSEN et al., 2012]. Segundo Delagi (2010), tal complexidade tende a aumentar cerca de 10 vezes entre 2012 e 2020.

Quando feita em Dispositivos Móveis Portáteis - *Portable Mobile Devices* (PMDs) a compressão deve ocorrer em tempo real, tendo em vista as restrições na quantidade de quadros que podem ser armazenados sem compressão. Muitas vezes tal restrição de desempenho pode comprometer o consumo de energia. Logo, a e ciência energética também deve ser considerada, em conjunto com o desempenho. Uma possível abordagem para melhorar o desempenho da codi cação e garantir e ciência energética é realizar tarefas de computação intensiva em *hardware* dedicado [SEIDEL, 2014].

Enquanto responsável por parte signi cativa da compressão, a Estimação de Movimento – *Motion Estimation* (ME) também consome parte signi cativa do tempo de codi cação [BOSSEN et al., 2012]. Tal etapa se tornou ainda mais crítica considerando o HEVC, no qual a ME é responsável por cerca de 40% do tempo de codi cação, mesmo usando o algoritmo de busca rápida [BOSSEN et al., 2012]. A ME consiste, basicamente, em comparar a similaridade entre o bloco sendo codi cado, chamado de original, e diversos blocos candidatos para substituí-lo. Para isso é necessário o uso de uma métrica de similaridade, tais quais a Soma das Diferenças Absolutas - *Sum of Absolute Di erences* (SAD), a Soma das Diferenças Quadráticas - *Sum of Squared Di erences* (SSD) e a Soma das Diferenças Transformadas Absolutas - *Sum of Absolute Transformed Di erences* (SATD) [RICHARDSON, 2003].

A SAD é a métrica mais simples dentre as mencionadas anteriormente, por isso tal métrica é comumente utilizada, especialmente para sistemas embarcados com restrições no consumo de energia. A Equação 1 de ne o cálculo da SAD, onde $d_{i,j}$ é o valor na posição *i, j da* matriz *D* calculada de acordo com a Equação 2. Nota-se que esta métrica consiste apenas em somar todas as diferenças absolutas entre píxeis candidatos e originais.

$$SAD_{N\times M} = \sum_{i=1}^{N}\sum_{j=1}^{M} |d_{i,j}| \quad (1)$$

$$D_{N\times M} = C_{N\times M} - O_{N\times M} \quad (2)$$

A SATD pode ser vista como a SAD, com as diferenças transformadas para o domínio de frequências antes de realizar a operação de absoluto e acumular os valores. Usualmente a transformada escolhida é a Transformada Hadamard – *Hadamard Transform* (HT), devido ao fato de que é uma boa aproximação à Transformada Discreta dos Cosenos - *Discrete Cosine Transform* (DCT) (transformada usada na etapa de transformação), porém mais simples de se calcular. Segundo Zhu e Xiong (2009), a SATD baseada na HT obtém resultados melhores em comparação com a SAD.

O cálculo da SATD baseada na HT (doravante referenciada apenas como SATD) é apresentado na Equação 3, onde *c* é uma constante de dimensionamento, $td_{i,j} \in TD$ é

um elemento da matriz de diferenças transformadas. Tal matriz é obtida como mostrado na Equação 4, onde $T_{N\times N}$ representa a matriz Hadamard (Equação 5 mostra uma Hadamard 4×4).

$$SATD_{N\times N} = c\times\sum_{i=1}^{N}\sum_{j=1}^{N}|td_{i,j}| \quad (3)$$

$$TD_{N\times N} = T_{N\times N}\times D_{N\times N}\times T^{-1}_{N\times N} \quad (4)$$

$$H_{4\times4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (5)$$

Pode-se perceber, pela de nição da SATD, o custo extra decorrente de calcular a transformada. Por este motivo, a SATD tipicamente é usada apenas para poucos casos como uma maneira de realizar ajustes nos, como na Estimação de Movimentos Fracionária – *Fractional Motion Estimation* (FME) e na predição intra.

É importante também considerar que a fase de transformada pode ser dividida em duas partes [PORTO et al., 2005], devido à propriedade da separabilidade. Tais partes são responsáveis pelos resultados de, respectivamente, a primeira e segunda multiplicação de matrizes da Equação 4. Neste cenário de divisão das operações de transformada, as partes são chamadas de transformadas-1D que formam uma transformada-2D.

Padrões de codi cação de vídeo do estado da arte, tais como H.264 e HEVC, usam diferentes tamanhos de blocos. Por isso é preciso levar em consideração como a combinação de arquitetura e métrica vai se comportar ao variar o tamanho dos blocos processados. No caso da SATD, a estrutura necessária para seu cálculo aumenta signi cativamente ao aumentar os tamanhos dos blocos. Este fenômeno se deve à natureza do cálculo da SATD, envolvendo operações com matrizes.

Outro aspecto signi cativo a respeito da SATD, é o fato de que a operação não é bem de nida para matrizes assimétricas. Assim, no caso do HEVC, o HM faz o cálculo desses blocos somando os valores de SATD (compondo o valor desses blocos). Além disso, o HM implementa tamanhos maiores que 8×8 como composição de blocos menores. Blocos de tamanhos não quadrados ($TD_{N\times M}$, sendo $N\neq M$) ou de tamanhos que não são potências de 2 também precisam de composição, uma vez que não existe HT para tais tamanhos. Porém, segundo He et al. (2015), calcular a SATD com blocos maiores, em vez de compor blocos, resulta em ganhos médios de \$0,05 dB\$.

Este trabalho explora alternativas de arquiteturas, buscando maior e ciência energética para o cálculo da SATD na compressão de vídeos de alta resolução (*e.g.*, *Full-HD*). Para isso foi desenvolvida uma arquitetura apresentando duas características principais. São estas:

- O uso e um *Bu er* Linear – *Linear Bu er* (LB)
- A possibilidade de se usar uma técnica de Eliminação Parcial das Distorções - *Partial Distortion Elimination* (PDE). Seidel, Bräscher e Güntzel (2015) apresentam uma arquitetura de SAD que usa tal técnica.

Tal arquitetura foi comparada diretamente à arquitetura com *Bu er* de Transposição – *Transpose Bu er* (TB) de Cancellier et al. (2014).

Na Seção 2 são apresentadas as arquiteturas propostas. A Seção 3 apresenta os resultados para as sínteses das arquiteturas propostas em comparação com uma arquitetura com TB. As conclusões provenientes deste trabalho são apresentadas na Seção 4.

2. Arquitetura Proposta

A Figura 1 mostra a parte comum aos blocos operativos das duas abordagens (TB e LB). Primeiramente as entradas *O* e *C* são armazenadas em registradores e em seguida é feita a diferença de tais entradas. Então, é realizada uma etapa de primeira transformada-1D FHT das diferenças e os resultados são armazenados em um *bu er*. As duas últimas etapas são apresentadas na Figura 1 como "*First Transform and Bu er Speci c Designs*". Os resultados parciais passam então por uma segunda etapa de transformada-1D FHT em *"N-Inputs 1-D H Transform"*. Em seguida faz-se o absoluto dos valores obtidos da segunda transformada-1D. Os valores absolutos são somados em um esquema de árvore de soma e acumulados em *PSATD*. Ao nal, os valores são divididos por dois usando um deslocamento de bits à direita e tem-se o resultado da SATD. No caso da LB-SATD, não é necessário o registrador após o acumulador devido a um estado de *"done"*.

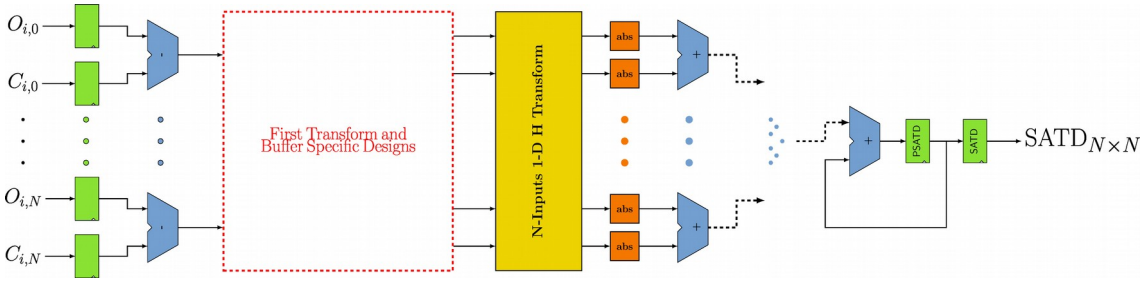


Figura 1. Bloco operativo de ambas as arquiteturas (SATD-LB e SATD-TB). O bloco tracejado pode ser substituído por um dentre dois modelos especí cos para o cálculo da SATD, o TB ou o LB.

Na Figura 2 é apresentada a estrutura especí ca à arquitetura TB-SATD, contendo a estrutura para a realização da primeira transformada-1D, o TB e multiplexadores. A primeira transformada é realizada de maneira a preencher o *bu er* alternando entre preenchimento por linhas e por colunas. O *bu er* é composto por registradores com multiplexadores nas suas entradas para determinar se a escrita será feita por linha ou por coluna. A cada passo de preenchimento de tal estrutura, os dados são deslocados por linha ou por coluna (no mesmo sentido em que a escrita está sendo

realizada). Da mesma forma, as saídas de tal estrutura contém multiplexadores para ler de maneira transposta os dados à medida que eles chegam nas últimas posições. A leitura transposta ocorre pois os dados da primeira execução são lidos durante a segunda execução. Ou seja, se na primeira execução os resultados foram escritos por colunas, na segunda os dados são escritos por linhas e lidos da última linha. Em tal exemplo, na terceira execução o sentido de operação volta a ser por linhas. Assim é mantido um fluxo contínuo após um período inicial, que é interrompido no final da execução para esvaziar o TB.

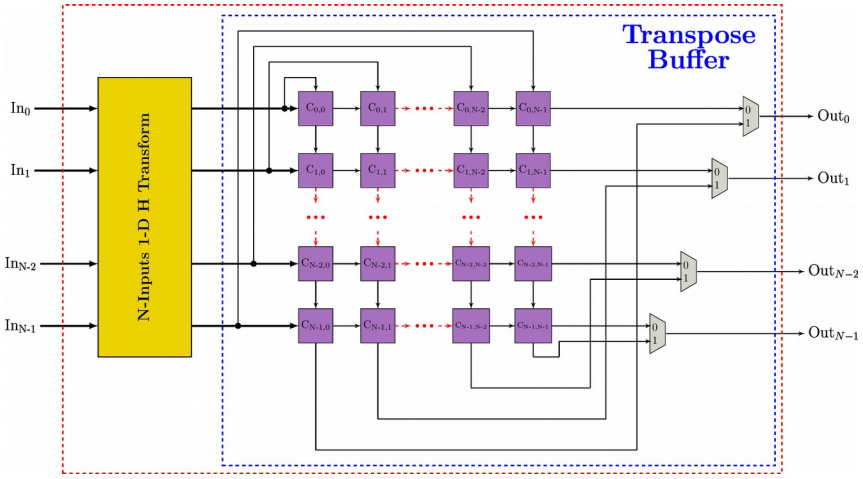


Figura 2. Arquitetura específica à TB-SATD.

O problema de se usar um LB está no fato de que ao realizar a transposição, apenas parte dos resultados parciais ficam armazenados. Com isso, a cada ciclo de cálculos da primeira transformada-1D, $\frac{N-1}{N}$ resultados deixam de ser usados. Assim é preciso repetir a transformada-1D para cada posição da transposição, selecionando o resultado que será usado através de multiplexadores. A estrutura usada para tal esquema de operação é apresentada na Figura 3. Nesta figura, os blocos *"butterfly"* contém a estrutura apresentada anteriormente, na Figura \ref{fig:butterflyStruct}. Percebe-se que a arquitetura apresentada na Figura 3 é significativamente menor que a versão da Figura 2. A redução de área vem do uso de uma transformada-1D reduzida para calcular apenas um resultado por ciclo e um *buffer* com apenas uma linha do TB. Além disso o LB não apresenta conexões entre registradores, reduzindo a quantidade de fios. Tal característica é especialmente importante à medida que se reduz os tamanhos das tecnologias.

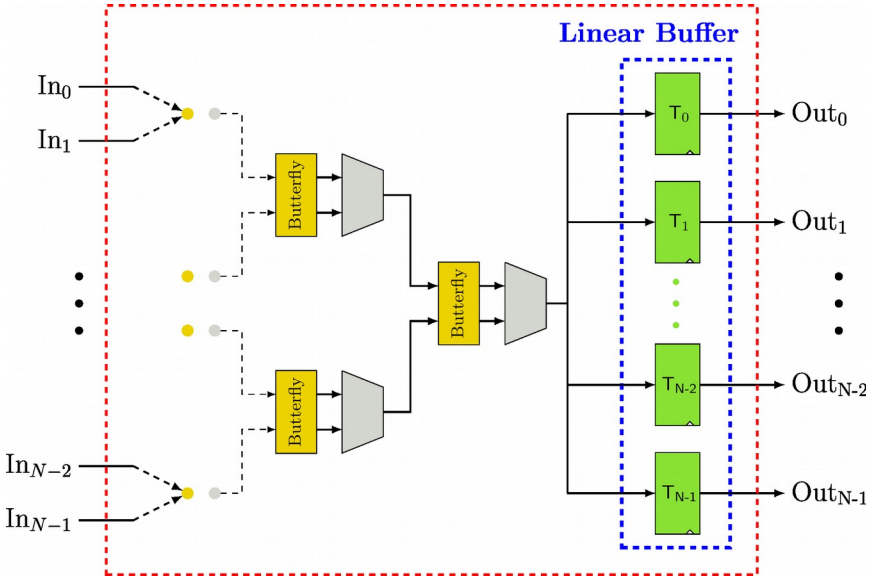


Figura 3. Bloco operativo de ambas as arquiteturas (SATD-LB e SATD-TB). O bloco tracejado pode ser substituído por um dentre dois modelos especí cos para o cálculo da SATD, o TB ou o LB.

Para o controle das arquiteturas com TB e LB foram usadas, respectivamente as FSMs nas Figuras 4 e 5. Ambas as FSMs consideram que a cada ciclo de cálculo os vetores de dados adequados estarão estáveis em suas entradas. A FSM da LB-SATD tem muito menos estados pois foram colocados contadores no bloco operativo, responsáveis por contar colunas e linhas processadas. Em contrapartida, as contagens necessárias à TB-SATD são realizadas através dos estados das FSMs. Além disso, as FSMs para TB-SATD modelam a lógica de carregamento e descarregamento do TB, enquanto que tais funções são realizadas no decorrer da operação da LB-SATD (carrega na primeira etapa de transformada e descarrega enquanto soma as parcelas).

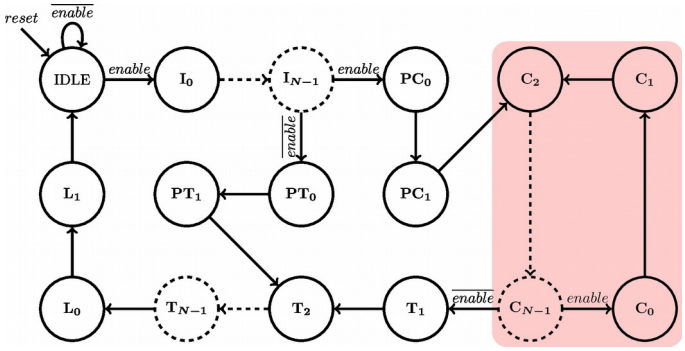


Figura 4. Máquina de Estados Finitos para o controle da arquitetura TB-SATD.

Apesar da LB-SATD ter menos estados, são necessários N^2+N+2 ciclos para se obter uma $SATD_{N\times N}$. Por outro lado apenas N ciclos são necessários para a TB-SATD. Isso ocorre pois a LB-SATD precisa calcular as N colunas (da primeira transformada-1D), para cada uma das N linhas a serem processadas pela segunda transformada-1D. A Tabela 1, mostra com mais detalhes os ciclos necessários para obter uma SATD, considerando diferentes tamanhos para as arquiteturas consideradas. Assim ca aparente a troca realizada entre tamanho do *bu er* (N^2 *vs.* N) por ciclos

necessários para o cálculo (*N vs.* N^2+N+2), respectivamente para TB-SATD e LB-SATD.

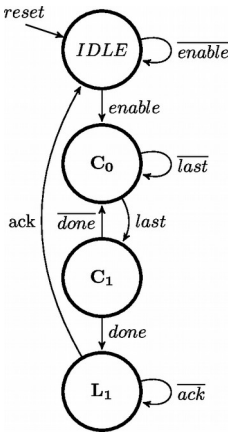


Figura 5. Máquina de Estados Finitos para o controle das arquiteturas LB-SATD e LB-SATD PDE.

Tabela 1. Ciclos/SATD, considerando diferentes tamanhos de blocos, para as arquiteturas TB-SATD e LB-SATD.

	4×4	8×8	16×16	32×32	$N \times N$
TB	4	8	16	32	N
LB	22	74	274	1058	$N^2 + N + 2$

3. Resultados

As arquiteturas foram sintetizadas com a ferramenta *Synopsys Design Compiler* [SYNOPSYS, 2011]. As sínteses usaram uma biblioteca de célula padrão - *standard cell* industrial de 45nm da TSMC, versões Nominal e *Low-Vdd/High-Vt* (LH). LH, consiste em reduzir a tensão de alimentação dos transistores e aumentar a tensão de limiar - *threshold*. Isso tem o objetivo de reduzir o consumo de energia dos transistores, porém os torna mais lentos. Devido aos maiores atrasos de transistores LH, eles só são recomendáveis quando os requisitos de *timing* não são muito restritos. Usar LH com requisitos muito restritos pode obrigar o uso de portas muito grandes (piorando o consumo de energia) ou causar violações de *timing*. Foram usadas as mesmas con gurações usadas por Seidel, Bräscher e Güntzel (2016). Tais con gurações são: atrasos de entrada e saída de 60% do período de relógio e máxima capacitância primária de entrada ajustada para 10 vezes a capacitância de uma porta *And* de duas entradas. Além disso, o *Synopsys Design Compiler* foi executado no modo *Topographical* para estimar capacitâncias parasitas de roteamento [SYNOPSYS, 2009]. As arquiteturas foram sintetizadas a m de manter a mesma quantidade de amostras por unidade de tempo (*throughput*) de trabalhos anteriores, 16 milhões de blocos $4\times4/s$ [WALTER; DINIZ; BAMPI, 2011].

Assim foram obtidos os resultados de área apresentados na Figura 6 e de energia mostrados na Figura 7.

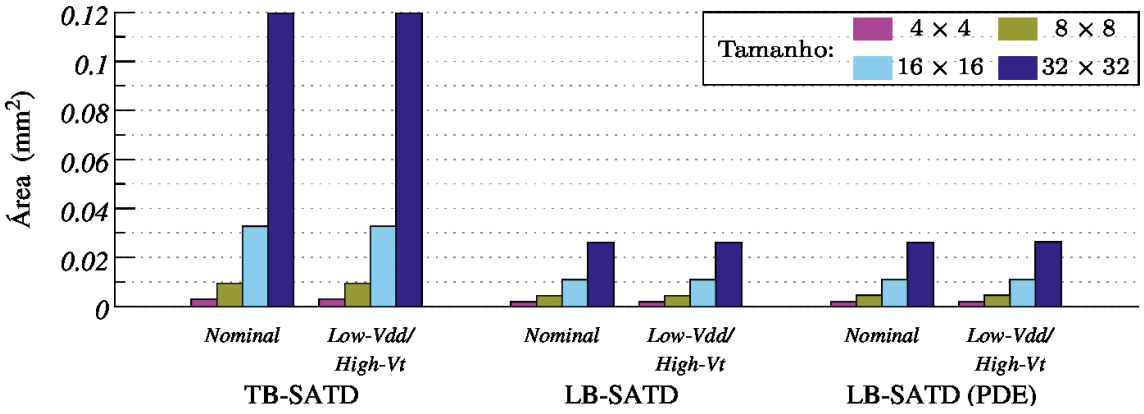


Figura 6. Estimativas de área.

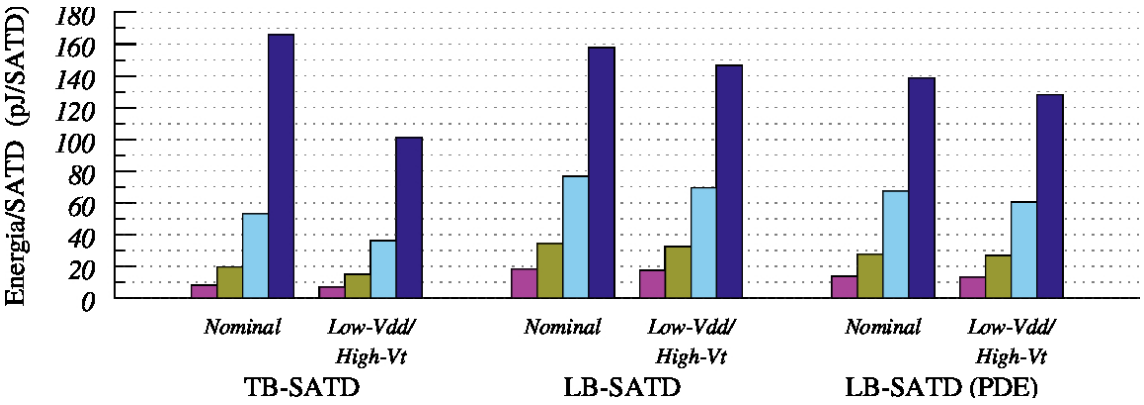


Figura 7. Estimativas de energia por 16 píxeis de SATD.

4. Conclusão

O objetivo deste trabalho foi investigar alternativas que pudessem reduzir o consumo energético do cálculo de SATD. Para isto, foi proposta uma alternativa arquitetural para cálculo de SATD (LB-SATD) para vários tamanhos de blocos de píxeis. Tal arquitetura apresenta uma estrutura do tipo LB, para reduzir o problema da escalabilidade de arquiteturas como a TB-SATD. Os resultados obtidos das arquiteturas LB-SATD foram apresentados por Seidel, Bräscher e Güntzel (2016). Além disso, foram implementadas versões das arquiteturas LB-SATD com PDE, a m de reduzir o grande número de ciclos necessários para se obter uma SATD. A m de mensurar o impacto do uso de PDE, foram feitas simulações na HM conforme as CTC, para obter informações estatísticas de quantos ciclos podem ser poupados ao usar PDE. Considerando as informações de média de ciclos necessários para SATD com PDE, obteve-se melhor consumo de energia, porém não foi possível superar a TB-SATD considerando síntese LH. Notou-se alta taxa de eliminação de candidatos gerados computacionalmente, assim a arquitetura LB-SATD PDE é uma boa alternativa para este tipo de aplicação. Por m, tem-se como possível trabalho futuro a realização de testes com *clock gating* para as arquiteturas avaliadas durante este trabalho. Tal técnica pode ser especialmente bené ca para a LB-SATD, devido à alta frequência de operação aliada ao fato de que a estrutura após o LB ca sub-utilizada durante parte signi cativa da operação da arquitetura.

References

BOSSEN, F. et al. HEVC complexity and implementation analysis. IEEE Transactions on Circuits and Systems for Video Technology, v. 22, n. 12, p. 1685–1696, Dec 2012. ISSN 1051-8215.

CANCELLIER, L. H. et al. Energy-e cient Hadamard-based SATD architectures. In: Proceedings of the 27th Symposium on Integrated Circuits and Systems Design. New York, NY, USA: ACM, 2014. (SBCCI '14), p. 36:1–36:6. ISBN 978-1-4503-3156-2.

DELAGI, G. Harnessing technology to advance the next-generation mobile user-experience. In: 2010 IEEE International Solid-State Circuits Conference - (ISSCC). [S.l.: s.n.], 2010. p. 18–24. ISSN 0193-6530.

HE, G. et al. High-throughput power-e cient vlsi architecture of fractional motion estimation for ultra-hd HEVC video encoding. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, v. 23, n. 12, p. 3138–3142, Dec 2015. ISSN 1063-8210.

ITU-T. H.264 : Advanced video coding for generic audiovisual services. Geneva, May 2003.

JCT-VC. HEVC Test Model. 2013. Dispon ıvel em: <http://hevc.hhi.fraunhofer.de/>.

JVT. JM JOINT VIDEO TEAM Reference Software. 2011. Disponível em: <http://iphome.hhi.de/suehring/tml/>.

MAICH, H. et al. HEVC fractional motion estimation complexity reduction for real-time applications. In: Circuits and Systems (LASCAS), 2014 IEEE 5th Latin American Symposium on. [S.l.: s.n.], 2014. p. 1–4.

NDILI, O.; OGUNFUNMI, T. E cient sub-pixel interpolation and low power vlsi architecture for fractional motion estimation in H.264/AVC. In: Signal Processing and Communication Systems (ICSPCS), 2010 4th International Conference on. [S.l.: s.n.], 2010. p. 1–10.

PEREIRA, F. et al. H.264 8x8 inverse transform architecture optimization. In: Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI. New York, NY, USA: ACM, 2014. (GLSVLSI '14), p. 83–84. ISBN 978-1-4503-2816-6. Disponível em: <http://doi.acm.org/10.1145/2591513.2591564>.

PORTO, M. et al. Design space exploration on the H.264 4 × 4 Hadamard transform. In: NORCHIP Conference, 2005. 23rd. [S.l.: s.n.], 2005. p. 188–191.

RICHARDSON, I. E. G. H. 264 and MPEG-4 video compression: video coding for next-generation multimedia. [S.l.]: John Wiley & Sons Inc, 2003.

SEIDEL, I. Dissertação (mestrado), Análise do impacto de pel decimation na codi cação de vídeos de alta resolução. Florianópolis-SC: [s.n.], 2014.

SEIDEL, I. Redução de Complexidade e Energia em Codi cadores de Vídeo Digital Preservando a E ciência de Codi cação: Exploração de Propriedades das Métricas de Distorção Aplicadas no Casamento de Blocos. Tese (Seminário de Andamento (doutorado)) — UFSC, Florianópolis-SC, 2015.

SEIDEL, I.; BRÄSCHER, A. B.; GÜNTZEL, J. L. Combining pel decimation with partial distortion elimination to increase sad energy e ciency. In: . [S.l.: s.n.], 2015. No prelo.

SEIDEL, I.; BRÄSCHER, A. B.; GÜNTZEL, J. L. Energy-e cient SATD for beyond HEVC. In: Proceedings of the 2016 IEEE International Symposium on Circuits and Systems. [S.l.: s.n.], 2016.

SINANGIL, M. E. et al. Cost and coding e cient motion estimation design considerations for high e ciency video coding (HEVC) standard. IEEE Journal of Selected Topics in Signal Processing, v. 7, n. 6, p. 1017–1028, Dec 2013. ISSN 1932-4553.

SULLIVAN, G. J. Overview of international video coding standards (preceding H.264/AVC). In: . [S.l.]: Presented at Workshop on Video and Image Coding and Applications (VICA), 2005.

SULLIVAN, G. J. et al. Overview of the high e ciency video coding (HEVC) standard. IEEE Transactions on Circuits and Systems for Video Technology, v. 22, n. 12, p. 1649–1668, Dec 2012. ISSN 1051-8215.

SYNOPSYS. Synopsys's Design Compiler User Guide, Version C-2009.06. 2009.

SYNOPSYS. Synopsys Design Compiler, Version F-2011.09-SP5-2. 2011.

TANG, X. l.; DAI, S. k.; CAI, C. h. An analysis of tzsearch algorithm in jmvc. In: Green Circuits and Systems (ICGCS), 2010 International Conference on. [S.l.: s.n.], 2010. p. 516–520.

WALTER, F. L.; DINIZ, C. M.; BAMPI, S. Synthesis and comparison of low-power high-throughput architectures for SAD calculation. In: 2011 IEEE Second Latin American Symposium on Circuits and Systems (LASCAS). [S.l.]: IEEE, 2011. p. 1–4. ISBN 978-1-4244-9484-2.

ZHU, C.; XIONG, B. Transform-exempted calculation of Sum of Absolute Hadamard Transformed Di erences. Circuits and Systems for Video Technology, IEEE Transactions on, v. 19, n. 8, p. 1183–1188, Aug 2009. ISSN 1051-8215.

ZHU, J. et al. Fast prediction mode decision with Hadamard transform based rate-distortion cost estimation for HEVC intra coding. In: 2013 IEEE International Conference on Image Processing. [S.l.: s.n.], 2013. p. 1977–1981. ISSN 1522-4880.

Proposta de uma Plataforma de Sistemas Multiagentes para Suportar a Gerência Autônoma de Recursos em Ambientes de Computação em Nuvem

Alexandre de Limas Santana[1], **Lucas Berri Cristofolini**[1]

[1]Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

alexandre.limas.santana@gmail.com, lucas.cristofolini@grad.ufsc.br

Abstract. *Considering the complexity, heterogeneity and dynamism presented on cloud computing environments, performing an optimization in such environments turns into a challenging task. This work is driven by the similarities between cloud computing and multiagent systems paradigms and aims to propose the addition of a multi-agent systems platform into an existing orchestration tool, thereby enabling autonomous agents to handle the analysis, planning and management of cloud computing environments.*

Resumo. *Dada a complexidade, heterogeneidade e dinamismo crescentes presentes nos ambientes de computação em nuvem, otimizar a utilização de recursos nestes ambientes torna-se uma tarefa desafiadora. Motivado pela semelhança entre entre os paradigmas de computação em nuvem e sistemas multiagentes, este trabalho se propõe a adaptar uma ferramenta de orquestração existente, de modo a utilizar uma plataforma de sistemas multiagentes para possibilitar que agentes inteligentes realizem a análise, o planejamento e a gerência em ambientes de computação em nuvem de forma independente e autônoma.*

1. Introdução

Computação em nuvem (CN) é um paradigma tecnológico que vem chamando a atenção dos provedores de serviços, por propor uma mudança na forma de disponibilizar seus produtos [Armbrust et al. 2010]. A grande aceitação aos serviços de CN vista recentemente se deve, entre outros motivos, à não necessidade de um grande investimento inicial, sendo que os clientes que buscam esse serviço pagam apenas pelo que utilizam [Ibrahim et al. 2011].

Os benefícios da CN estão altamente ligados com a *Quality of Service* (QoS - Qualidade de Serviço) percebida pelos usuários. A necessidade de entregar QoS aos consumidores exige que recursos sejam mantidos ociosos a espera de cargas estocásticas, consumindo mais energia e influenciando diretamente nos custos de manutenção de um ambiente na nuvem [Weingärtner et al. 2015]

Uma alternativa para reduzir este consumo desnecessário consiste em agrupar as máquinas virtuais ativas no menor número de servidores físicos, possibilitando desligar os recursos ociosos e ligá-los quando a demanda torná-los necessários [Awada et al. 2014]. Entretanto, dada a falta de suporte a medidas de gerenciamento energético nas ferramentas

de orquestração de ambientes de CN disponíveis atualmente, [Bräscher 2015] propôs a adoção de um *framework* para a consolidação de recursos em ambientes de CN, visando uma melhor eficiência energética do ambiente orquestrado.

Problemas como este, onde não se conhece uma fórmula para obter a melhor solução e onde a informação está distribuída pelo ambiente, podem ser tratados através de uma abordadem de sistemas multiagentes (SMA). Dessa forma, é possível trabalhar sob diferentes pontos de vista acerca do problema, trazendo o conhecimento de especialistas para os agentes do ambiente, que por sua vez atuarão para alcançar seus objetivos [Silveira 2006].

2. Proposta

Devido ao crescimento da popularidade e da oferta de serviços de computação em nuvem, seja a nível de SaaS, PaaS ou Iaas, a gerência dos ambientes utilizados para fornecer estes serviços tem se tornado um tópico ativo de pesquisa, buscando-se métodos de manter e distribuir recursos computacionais em ambientes de CN de forma mais eficiente [Whitney and Delforge 2014]. Recentemente pode-se encontrar trabalhos que propõem modelos de sociedades de agentes para realizar tal função de maneira autônoma e distribuida tais como [Hou et al. 2014] e [De la Prieta et al. 2013], constatando assim o interesse de pesquisas no casamento dos paradigmas de SMA e CN.

Este trabalho busca expandir a solução implementada por [Bräscher 2015], adaptando a arquitetura inicialmente proposta introduzindo uma plataforma SMA, que por sua vez estará ligada a ferramenta de orquestração. Assim, também abre-se a possibilidade de que a ferramenta de orquestração possa futuramente fazer uso de agentes alheios aos processos propostos em [Bräscher 2015].

Analisando o modelo de componentes de uma ferramenta de orquestração proposto em [Bräscher 2015] na Figura 1, nota-se, em verde, as modificações realizadas por seu trabalho. Estas mudanças são a adição de um novo componente chamado **gerente de consolidação** e a modificação do módulo já existente **gerente de alocação**. Tais ajustes são oriundos das adições de seus gerentes autônomos cujos objetivos são a consolidação e re-alocação de máquinas dentro do ambiente de CN. Para que tais gerentes possam ser modelados na forma de agentes inteligentes é necessário que seja providenciado um local onde esses possam residir dentro do ambiente. Através do uso de uma plataforma de SMA para alocar os agentes, consegue-se realizar a desacoplação dos gerentes da ferramenta de orquestração, implicando em um melhor processo de escalabilidade à solução existente.

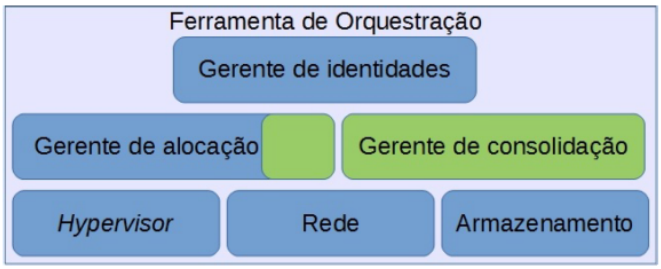


Figura 1. Componentes de uma ferramenta de orquestração expandida pelo *framework*, retirado de [Bräscher 2015].

Os agentes inteligentes usados por uma ferramenta de orquestração podem ser requisitados em vários segmentos desta. A partir da inclusão da plataforma de SMA, pode-se unificar o acesso a eles através da criação de uma interface visível à ferramenta de orquestração, que por sua vez se conecta à plataforma SMA e seus agentes. Uma abordagem desse tipo adiciona um componente a mais na arquitetura da ferramenta de orquestração, porém mantem a lógica dos agentes desacoplada. Para alcançar esse objetivo, este trabalho propõe uma mudança nos componentes da ferramenta de orquestração afim de adicionar um novo módulo, o gerente de SMA. Este novo módulo tem o objetivo de servir de fachada para a plataforma de SMA, garantindo acesso para os nodos de controle da ferramenta de orquestração.

2.1. Arquitetura do Gerente de SMA

Segundo a Figura 1, o uso do *framework* proposto por [Bräscher 2015] incrementa o número de elementos básicos de uma ferramenta de orquestração de 5 para 6. A primeira mudança proposta por este trabalho é renomear o gerente de consolidação para agente de gerência, devido ao fato de que o gerente de consolidação em [Bräscher 2015] não se restringe apenas a executar a consolidação, mas depende apenas das heurísticas apontadas a ele. Sendo assim, este agente trabalha de forma a realizar essas atribuições dentro de um *cluster*, tal como distribuição de cargas entre servidores.

Existe também a necessidade de um novo módulo com o propósito de manter a comunicação da ferramenta de orquestração com a plataforma de SMA. Este módulo, chamado de gerente de SMA pode ser visto na Figura 2, onde estão representados os elementos da ferramenta de orquestração proposto por este trabalho, destacando em um gradiente mais escuro os módulos adicionados e modificados.

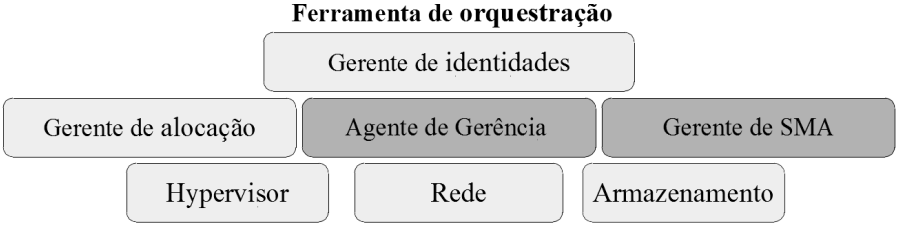


Figura 2. Componentes de uma ferramenta de orquestração incrementados pelo gerente de SMA.

A atribuição do novo componente é gerenciar a plataforma de SMA, garantido o ciclo de vida de todos os componentes distribuídos que compõem a plataforma de SMA. Desta forma, a ferramenta de orquestração pode manter-se alheia a esses processos e assumir que enquanto ela própria estiver operando, a plataforma de SMA está instanciada. Para fazer esta garantia, são adicionadas duas abstrações ao componente de gerência de SMA:

1. Plataforma de SMA: é a abstração dos estados, componentes ativos e aspectos da plataforma de SMA do ambiente;
2. Gerente de plataforma: uma entidade que responsabiliza-se por garantir a atividade, acessibilidade e tolerância a faltas da plataforma de SMA.

Sendo o gerente de plataforma uma parte integrante da ferramenta, este deve servir para comunicar a plataforma de SMA com o restante do ambiente. Dessa maneira, a ferramenta de orquestração e a plataforma de SMA podem manter-se desconexas, de forma que a ferramenta de orquestração não saiba que há agentes operando em seus recursos. A escolha de não acoplar a plataforma diretamente é uma maneira de garantir a modularidade do gerente de SMA. Sendo assim, quando outros segmentos da ferramenta de orquestração sentirem a necessidade de sociedades de agentes, estes poderão ser adicionados a plataforma sem grandes mudanças na ferramenta de orquestração. Caso também seja constatado que há a necessidade de trocar a própria tecnologia da plataforma de SMA, esta mudança refletirá em alterações apenas no gerente de SMA. A Figura 3 expande a visão deste componente, ilustrando a conexão de suas entidades com o restante da ferramenta de orquestração.

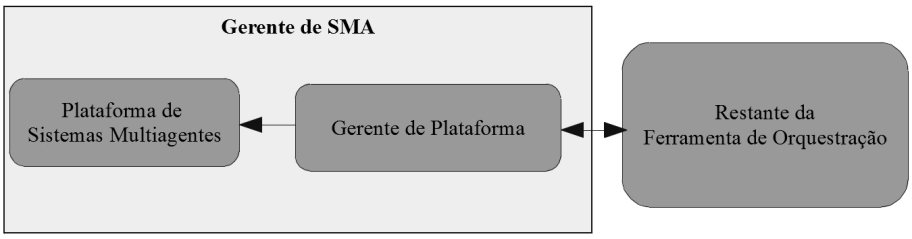


Figura 3. Relações entre componentes do gerente de SMA e elementos de uma ferramenta de orquestração.

2.1.1. Gerente de Plataforma

A plataforma multiagente usada nesse trabalho é a ferramenta JADE, escolhida por implementar o padrão FIPA, permitir que a plataforma seja distribuida e por oferecer tolerância a faltas através de redundância de seus nodos, além do fato de permitir a instancia de agentes reativos e BDI. A plataforma JADE funciona como uma central de comando para os agentes, sendo os processos de criação, comunicação e gerência das atividades desses realizado pela própria API da plataforma.

O JADE é composto por *containers* que precisam ser instanciados em uma JVM. Para facilitar a integração com a ferramenta de orquestração, os *containers* JADE são instanciados em uma *Virtual Machine* (VM - Máquina Virtual) dedicada a funções de sistema dentro do ambiente de CN sob o controle de alguma ferramenta de orquestração. A instalação do JADE, assim como de todas suas dependências nessa VM deve ocorrer de forma automática, sem ser necessária a interferência de um administrador. Para isso, são utilizadas funções implementadas na ferramenta de orquestração extendida em [Bräscher 2015] para instanciar uma máquina virtual contendo uma *Java Virtual Machine*, necessária para o funcionamento do JADE.

O gerente de plataforma é a entidade responsável por fazer os pedidos de recursos que são necessários para executar o JADE, implementando desde a obtenção de VMs de sistema até a instalação dos requisitos do JADE. Para que a criação de um *container* seja justificada o gerente de plataforma deve autonomamente perceber o estado atual do ambiente afim de decidir sua próxima ação. Para que isto seja possível, esse módulo deve

comunicar-se com os *containers* JADE periodicamente. Dependendo de quais *containers* estão ativos, pode ser percebebido a necessidade da criação de novos nodos.

Durante o processo de percepção do ambiente de nuvem, o gerente de plataforma deve encontrar *containers* que não estão funcionando adequadamente. Cabe a esse módulo detectar inconsistências na plataforma e corrigi-las. Através da destruição de nodos, as máquinas virtuais associadas a estes ficam sem propósito. Sendo assim, estas VMs são desalocadas através da ferramenta de orquestração. Pode-se argumentar que tais VMs podem ser usadas para receber novas instâncias de *containers* da plataforma. Sendo assim o ciclo de vida do gerente de plataforma pode ser visualizado na Figura 4.

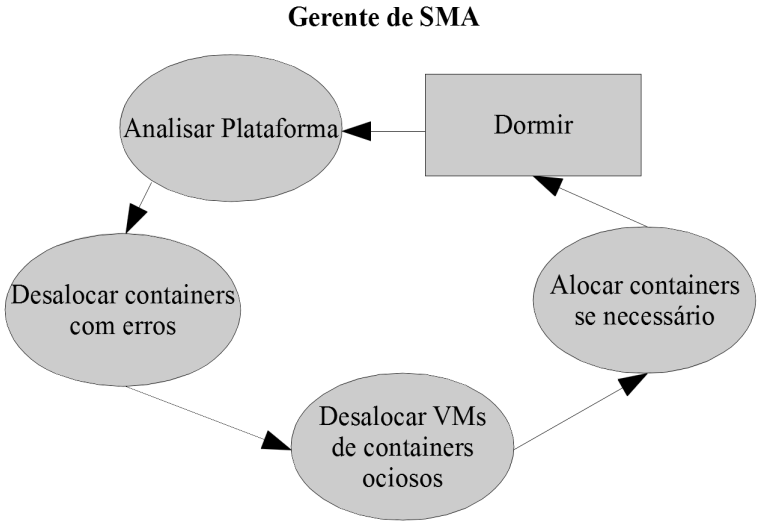


Figura 4. Ciclo de funcionamento do gerente de plataforma.

Com a gerência da plataforma garantida através dos componentes do gerente de SMA, é obtida a estabilidade e acessibilidade a plataforma de SMA. Sendo assim, resta apenas o controle do ciclo de vida dos agentes e modos para criação destes. Uma vez que a ferramenta de orquestração não deve perceber os agentes, esta não há de instanciar nenhum. A responsabilidade por detectar que agentes são necessários dentro do âmbito da ferramenta de orquestração, criá-los e destruí-los cabe a um agente que reside dentro da plataforma SMA, nomeado de agente inspetor.

2.2. Agentes do Sistema de Gerência

Uma vez que o funcionamento da plataforma de SMA está garantido pelo componente de gerência, os agentes modelados na arquitetura correspondente já podem ser instanciados e começar a atuar no ambiente. A ferramenta de orquestração permite o sensoriamento do seu ambiente para seus nodos de gerência, fato este que garante aos agentes a capacidade de observar os recursos do ambiente.

Embora seja garantido aos agentes um ambiente para atuarem e uma arquitetura para estarem contidos, estes ainda precisam ser instanciados. Esse processo, na tecnologia JADE usada neste trabalho, deve ser feito em tempo de execução ou através de parâmetros na inicialização de um dado *container*. No âmbito desta proposta ambos sistemas de inicialização dos agentes são empregados para garantir que tanto o componente

de gerência de agentes quanto o restante da ferramenta de orquestração não precisem ativamente preocupar-se com a inicialização dos agentes.

A modelagem dos agentes elaborados neste trabalho foi realizada utilizando a *Prometheus Modeling Tool* que utiliza a notação apresentada na Figura 5:

Figura 5. Legenda relativa aos diagramas dos modelos de agentes.

- Ação: Mecanismo que permite que o agente interaja com o ambiente em que se econtra. Pode representar funções atuadoras ou interações com artefatos;
- Capacidade: Abstração de uma funcionalidade do sistema. É composta por planos, eventos do ambiente e dados;
- Dados: Representação de um item ou estrutura de dados utilizado pelo sistema.;
- Mensagem: Mensagem trocada entre agentes;
- Plano: Sequência de eventos e ações que resultam na execução de um objetivo;
- Percepção: Informação oriunda de algum ponto do ambiente que pode ser interpretada pelo agente.

2.2.1. Agente Inspetor da Plataforma

O agente inspetor da plataforma deve existir enquanto houver uma instância da arquitetura de SMA ativa. Dada essa característica, este inspetor é inicializado junto aos *main containers* da plataforma através de parâmetros na inicialização dessa. Deste modo, o agente inspetor possui seu próprio ciclo de vida garantido pelo fato de que os nodos principais da plataforma contém redundância, mantendo-se ativos com cópias para atuar como nodo principal na falha do *main container* corrente. O inspetor da plataforma tem como responsabilidade tanto instânciar os agentes para efetuarem a gerência do ambiente de computação em nuvem, quanto realizar o processo de gerenciamento destes. Caso seja notado que um determinado agente não seja mais necessário, é responsabilidade do inspetor terminar sua execução.

Sendo esse o único agente de meta gerência, seus objetivos diferem do restante da sociedade em relação ao domínio de sua atuação (único agente que sensoria e tem suas ações direcionadas a própria plataforma de SMA). Para simplificar a visualização do restante da sociedade de agentes, este agente é modelado de forma separada (como um sistema disjunto do restante da sociedade de agentes), de modo que todas as atribuições, percepções e funcionalidades desse sub-sistema competem a ele apenas. Seguindo a metodologia *Prometheus*, baseada em teoria de agentes, podemos decompor este sistema em funcionalidades, percepções, objetivos, ações e protocolos de comunicação [Padgham and Winikoff 2003].

Os eventos trazem informações diretas e completas para o agente, o qual consegue deliberar sobre as implicações diretas de seus acontecimentos. As percepções, no entanto, precisam ser analisadas e ponderadas [Padgham and Winikoff 2003]. Sendo assim os eventos e percepções relacionados com esse agente são os que seguem:

1. Evento da destruição de *container*: ocorre quando um nodo da plataforma é destruido, terminando ações de agentes que lá residiam;
2. Evento de criação de *container*: ocorre quando um novo nodo da plataforma é criado, podendo conter novos agentes;
3. Percepção de mudança nas necessidades do sistema: ocorre ao ser percebido que a ferramenta de orquestração mudou suas heurísticas;
4. Percepção de ócio: percebido quando a tarefa executada por um grupo de agentes está usando mais recursos do que precisa para sua completude;
5. Percepção de subdesempenho: percebido quando uma tarefa executada por um grupo de agentes não está tendo sucesso de se concluir em uma taxa desejada.

As ações do inspetor são listadas a seguir:

1. Destruir um agente: terminar a execução de um agente de algum dado tipo;
2. Criar um agente: inicializar um agente dentro da plataforma de SMA;
3. Interpretar necessidades da ferramenta de orquestração: deriva um modelo de requisitos para ser implementado pelo inspetor sobre a vontade da ferramenta de orquestração;
4. Verificar espaço para novos agentes: confere se há local para criar novos agentes no ambiente ou se a plataforma está lotada;
5. Analisar tempo de execução de tarefa: mede o tempo que uma atividade está demorando para ser realizada;

A Figura 6 demonstra o documento chamado de *system overview* da metodologia *prometheus* para este sistema.

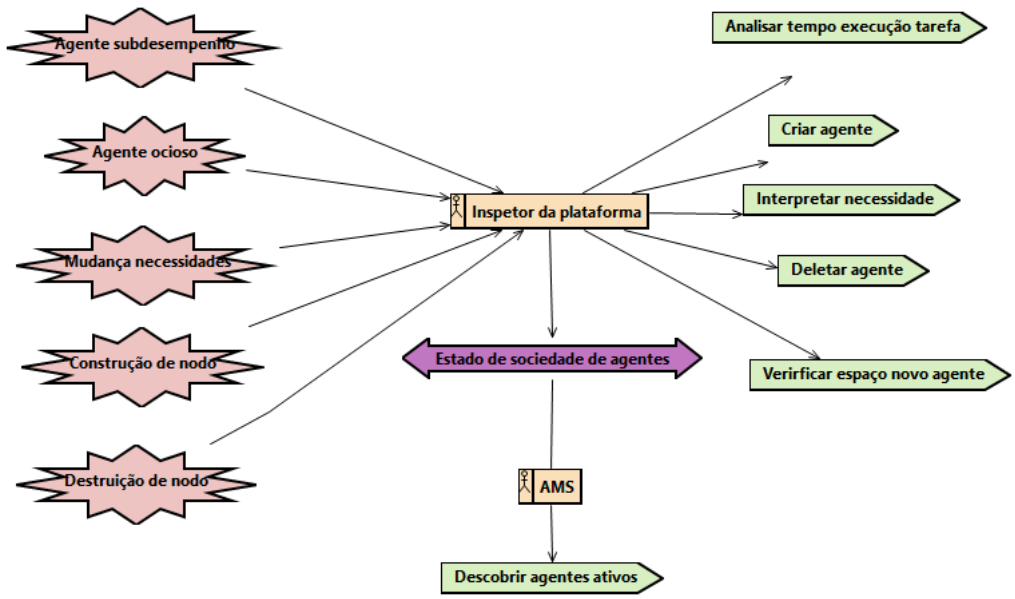


Figura 6. *System overview diagram* referente ao agente inspetor da plataforma.

2.3. Modelo dos Agentes de Gerência do Ambiente de Nuvem

Nesta seção serão descritos os modelos dos três agentes responsáveis por implementar as funções propostas em [Bräscher 2015].

2.3.1. Agente de Busca do Cluster

O agente de busca de *cluster* é um agente reativo, segundo a definição proposta em [Wooldridge et al. 1995], que classifica uma arquitetura de agente como reativa quando este não possui um modelo simbólico do mundo a sua volta e não faz uso de nenhum tipo de raciocínio simbólico. O agente tem o objetivo de monitorar o ambiente a procura de *clusters* que necessitam ser trabalhados, para então apresentá-los de forma que o agente diligente possa trabalhar sobre os *clusters*. Essa listagem é feita através de um *blackboard*, instanciado na plataforma JADE. Como ilustrado na Figura 7, o agente de busca terá conhecimento apenas dos *clusters* ativos do ambiente e, uma vez que um novo *cluster* for descoberto ou algum *cluster* existente tiver sido trabalhado pela última vez a muito tempo, ele deve escrever no *blackboard* a necessidade de trabalhar novamente sobre aquele *cluster*.

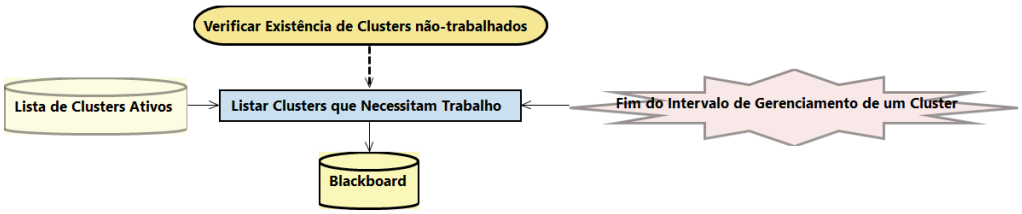


Figura 7. Modelo do agente de busca de cluster.

2.3.2. Agente Diligente do Cluster

O agente diligente do *cluster* é responsável por realizar os processos de gerência nos *clusters* que foram identificados como aptos para receber tais processos. Ao encontrar um *cluster* listado no *blackboard*, o agente deve aplicar as heurísticas definidas pelo administrador do ambiente para determinar a prioridade para a realização do trabalho sobre os seus *hosts*. Uma vez definida a prioridade dos *hosts* a serem trabalhados, o agente diligente solicita, em ordem, que as *VMs* dos *hosts* de menor prioridade sejam mapeados aos *hosts* de maior prioridade, na medida em que estes possuírem recursos suficientes disponíveis. Uma vez que um *host* não possui mais *VMs* mapeadas a ele, este *host* pode ser desativado. Como explicitado na Figura 8, o agente espera o término do trabalho para poder marcar o *cluster* como trabalhado por um tempo determinado pelo administrador e atuar sobre o próximo *cluster*.

2.3.3. Agente de Ativação de Recursos

O agente de ativação de recursos tem como objetivo manter a disponibilidade de recursos suficientes para o bom funcionamento do ambiente, sendo o responsável por verificar periodicamente a necessidade de disponibilizar mais recursos físicos para o ambiente. Ele é o responsável pela ativação de *hosts* que podem vir a serem desativados. Ao ser instanciado, é estabelecido um intervalo de tempo entre verificações, ao final do qual o agente compara a quantidade de recursos sendo utilizados com a quantidade de recursos

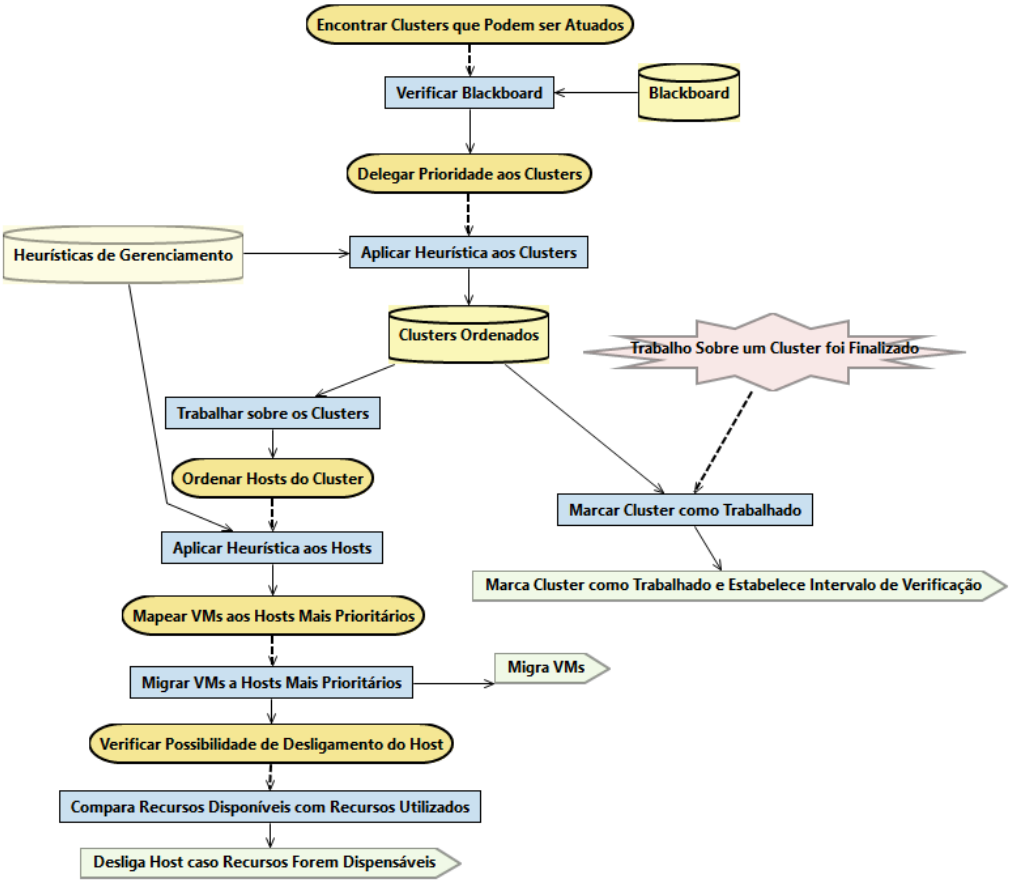


Figura 8. Modelo do agente diligente do cluster.

ativos e, baseado nas heurísticas, decide se devem ser ativados novos recursos e, sendo necessário, a quantidade a ser ativada. A Figura 9 explicita o modelo do agente.

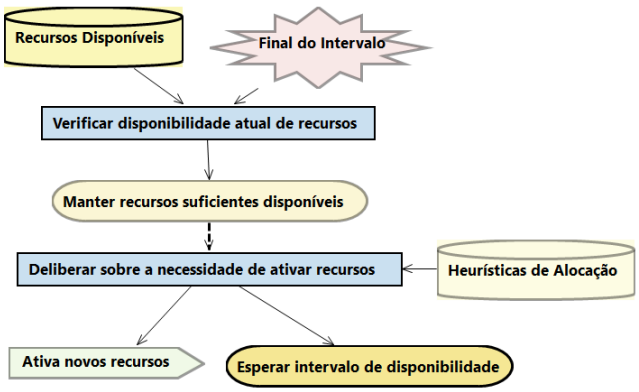


Figura 9. Modelo do agente de alocação.

Vale notar que o agente de ativação, por conta própria, não consegue garantir integralmente a disponibilidade de recursos físicos, já que podem haver casos de uma demanda inesperada por recursos enquanto o intervalo de verificação ainda estiver correndo. Para remediar estas situações, um gerente de recursos é injetado na plataforma

de orquestração do ambiente extendida por [Bräscher 2015], e atua sob demanda para disponibilizar recursos físicos, quando for tentada a alocação de uma *VM* e não forem encontrados recursos suficientes.

3. Desenvolvimento

Este trabalho foi executado sobre a versão 4.6.0 do *Apache Cloudstack*. Todo o processo de desenvolvimento ocorre dentro do escopo criado pelo *framework* de [Bräscher 2015] na versão 1.0.2. O arcabouço criado pelo *framework* é composto por um grupo de *plugins* que formam um módulo adicional no *Apache Cloudstack*, chamado de *autonomiccs platform*.

Para manter a hierarquia criada pela adição do *framework* de [Bräscher 2015] e a sua comunicação com a ferramenta de orquestração, este trabalho teve como objetivos tanto a criação de um módulo para instanciar e gerenciar a plataforma de sistema multiagentes, quanto modificar os *plugins* do *framework* de forma a dissociar os serviços criados que são inerentes ao *Apache Cloudstack* da lógica de tomada de decisões, afim de transforma-las nos agentes definidos e apresentados na seção 2.2.

Para a criação do gerente de SMA, o padrão seguido pelos agentes do *framework* de [Bräscher 2015] foi utilizado, injetando objetos no *Apache Cloudstack* através da ferramenta *Spring*. Assim garantindo a instância e execução do ciclo de percepção da plataforma. Com este gerente ativo, a criação das plataformas SMA e dos agentes nela inseridos pode ocorrer através de funções já existentes no *framework* para criar VMs e acessa-las através de SSH, injetando as dependências do JADE e inicializando-o.

Os agentes residentes na plataforma JADE que compõem a sociedade definida neste artigo foram desenvolvidos utilizando o *framework* de desenvolvimento de agentes JADE e suas funções de atuação envolvem o consumo de APIs disponibilizadas pela ferramenta contida no *framework* de [Bräscher 2015].

4. Conclusão

Embora não tenham sido realizados testes nas mudanças feitas no *framework* proposto por [Bräscher 2015], pode-se observar os efeitos que as modificações surtiram na organização e relação dos módulos do *autonomiccs plataform*. As vantagens dessa reorganização abrangem a modularização do código, a compatibilidade com o padrão FIPA para comunicação entre agentes, fazendo com que outras plataforma de SMA possam comunicar-se com o módulo de gerência da ferramenta de orquestração.

A comunicação criada entre JADE e *Apache Cloudstack* permite que desenvolvedores e pesquisadores possam ter um arcabouço para criar agentes capazes de acessar os recursos administrativos de uma ferramenta de orquestração. Além de fazer com que plataformas externas de SMA possam interoperar com os agentes do *Apache Cloudstack*. Acredita-se ainda que pela capacidade de adicionar múltiplos agentes para aplicar as heurísticas do ambiente na nuvem, sem adições extras no código, tornam essa abordagem mais escalável e eficiente. Entretanto, tendo em vista a ausência de testes para essa estensão, não há condições científicas de afirmar a eficiência e eficácia desta abordagem com sistemas multiagentes.

5. Trabalhos Futuros

Sendo assim, como trabalhos futuros, têm-se:

- Avaliar a eficiência e eficácia da abordagem de sistemas multiagentes contra a aplicação de serviços acoplados na ferramenta de orquestração proposta em [Bräscher 2015];
- Verificar a capacidade de escalabilidade dessa abordagem em comparação com outras atuais, como a proposta em [Bräscher 2015];
- Testar a adaptabilidade do SMA a heurísticas que não sejam a consolidação do ambiente em nuvem;
- Escrever um artigo científico demonstranto a solução proposta e seus resultados para ser publicado em uma conferência a ser definida.

Referências

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al. (2010). A view of cloud computing. *Communications of the ACM*, 53(4):50–58.

Awada, U., Li, K., and Shen, Y. (2014). Energy consumption in cloud computing data centers. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, 3(3):145–162.

Bräscher, G. B. (2015). *Proposta de Um Framework Para Consolidação de Recursos em Ambientes de Computação em Nuvem*. UFSC.

De la Prieta, F., Rodriguez, S., Bajo, J., and Corchado, J. (2013). A multiagent system for resource distribution into a cloud computing environment. In Demazeau, Y., Ishida, T., Corchado, J., and Bajo, J., editors, *Advances on Practical Applications of Agents and Multi-Agent Systems*, volume 7879 of *Lecture Notes in Computer Science*, pages 37–48. Springer Berlin Heidelberg.

Hou, F., Mao, X., Wu, W., Liu, L., and Panneerselvam, J. (2014). A Cloud-Oriented Services Self-Management Approach Based on Multi-agent System Technique. *Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on*, pages 261–268.

Ibrahim, S., He, B., and Jin, H. (2011). Towards pay-as-you-consume cloud computing. In *Services Computing (SCC), 2011 IEEE International Conference on*, pages 370–377. IEEE.

Padgham, L. and Winikoff, M. (2003). Prometheus: A methodology for developing intelligent agents. In *Agent-oriented software engineering III*, pages 174–185. Springer.

Silveira, R. A. (2006). Introdução a sistemas multi-agente. *Universidade Federal de Santa Catarina (UFSC)*.

Weingärtner, R., Bräscher, G. B., and Westphall, C. B. (2015). Cloud resource management: A survey on forecasting and profiling models. *Journal of Network and Computer Applications*, 47:99–106.

Whitney, J. and Delforge, P. (2014). Data center efficiency assessment. *Issue paper on NRDC (The Natural Resource Defense Council)*.

Wooldridge, M., Jennings, N. R., et al. (1995). Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2):115–152.

Um Sistema Para Geração Automática de Questões

Luís Gustavo T. Cordeiro

Departamento de Informática e Estatística - Universidade Federal de Santa Catarina
Florianópolis, SC - Brasil

luis_gtc@hotmail.com

Abstract*. This paper briefly describes some basic techniques of natural language processing that can be used in systems for text analysis and question generation through sentences in a given language. This work is based on Portuguese language and, although some information will be useful for other languages, specific techniques for other languages are not presented. The most popular approaches are described and a basic model of question generation will be detailed. Some examples and results are presented for better understanding of the developed model.*

Resumo. Este artigo descreve brevemente algumas técnicas básicas de processamento de linguagem natural que podem ser utilizadas em sistemas para a análise de texto e geração de questões através de sentenças em determinada linguagem. O trabalho é baseado na língua portuguesa e, apesar de algumas informações também serem úteis para outros idiomas, não são apresentadas técnicas específicas de outras linguagens. As abordagens mais conhecidas são descritas e um modelo básico de geração de questões será detalhado. Alguns exemplos são apresentados para melhor entendimento, bem como resultados obtidos pelo modelo desenvolvido.

1. Introdução

A geração de questões é um ramo bem específico da área de processamento de linguagem natural, que por sua vez faz parte do imenso tema da inteligência artificial. Yao (2010) define o termo como uma tarefa que une os esforços das tarefas de entendimento de linguagem natural e geração de linguagem natural. De forma simples, é uma tarefa conhecida como um mapeamento *text-to-text*. Uma das formas de se realizar esse processo de geração de questões pode ser observada no modelo descrito em Cordeiro (2016). Para entendermos como funciona o sistema desenvolvido para geração automática de questões nos formatos Cloze e *WH-Question* no estilo Múltipla Escolha, primeiramente devemos entender os conceitos mais básicos sobre o assunto.

A forma como uma questão é apresentada ao leitor pode ser diferente dependendo do objetivo que se deseja obter e o contexto em que ela se encontra. Os formatos mais comuns em testes escolares são os de múltipla escolha, onde uma pergunta é realizada e são oferecidas diversas opções e o leitor decide qual delas se adequa melhor como a resposta. Smith e Avinesh (2010) apontam como sendo um formato interessante, pois possibilita a avaliação automatizada dos testes.

O formato de múltipla escolha pode ser combinado com outros tipos de questões, pois é apenas um modo de apresentar opções de resposta ao leitor. O sistema desenvolvido

em Cordeiro (2016) utiliza desse formato em conjunto com as *WH-Questions* e questões de tipo Cloze. Este último também é muito utilizado e bem conhecido como as questões de "preencher o vazio", onde uma sentença é apresentada com uma ou mais expressões removidas para serem completadas. Já as WH-Questions são perguntas que usamos naturalmente no dia a dia para solicitar uma informação específica. O termo é derivado do inglês e representa o conjunto de perguntas que iniciam com as expressões "quem", "quando", "onde", "o que", "de quem", "por que" e "qual". Além dessas, existem diversas outras formas de se apresentar uma questão, como as de ligação de conceitos; verdadeiro ou falso; sim ou não; e somatório; dentre outras das quais este artigo não apresentará em mais detalhes.

Um dos conceitos mais utilizados atualmente em trabalhos da área de processamento de linguagem natural é o corpus. Segundo a definição de Sinclair (2004), um corpus é uma coleção de textos de uma determinada linguagem em formato eletrônico, selecionados de acordo com um critério externo para representar uma linguagem como fonte de dados para pesquisa linguística. Em geral, quanto maior um corpus, maior a sua representatividade, seguindo a lógica probabilística de que uma quantidade maior de exemplares aumenta a chance de palavras e expressões mais raras serem encontradas no conjunto.

Diversos autores como Gasperin e Lima (2001) e Souza e Felippo (2010) apresentam suas ideias sobre características específicas que devem ser abrangidas pelos corpora para que possam ser considerados adequados para estudo, como padronização de um tamanho mínimo aceitável e outros conceitos de diversidade de temas. Estes temas não fazem parte do foco deste artigo e não serão abordados.

As técnicas utilizadas para processamento de texto estão diretamente relacionadas aos tipos de análise textual descritos por em Müller (2003). Segundo o autor, "um sistema de processamento de linguagem natural é abordado do ponto de vista da análise do conhecimento morfológico, sintático e pragmático". Sendo assim, surgiram técnicas como a tokenização, o *stemming* e a lematização, que tratam da identificação das palavras e suas formações morfológicas. As técnicas de etiquetagem morfossintática e da resolução de referências abrangem a análise sintática, identificando o tipo de cada palavra e sua relação com outras partes do texto. Ambas as técnicas são de grande utilidade para evitar erros básicos na geração de alternativas para as questões de múltipla escolha, como será visto mais à frente. Por último apresentamos o reconhecimento de entidades mencionadas: um processo de identificação de alto nível de abstração capaz de classificar os sujeitos e objetos das sentenças em um grupo como "pessoa", "local" e "tempo", utilizado como base para o desenvolvimento do sistema em Cordeiro (2016).

2. Abordagens Para a Geração de Questões

Le et al. (2014) cita três abordagens utilizadas para o processo de geração de questões: sintática, semântica e baseada em *template*. As duas primeiras abordagens seguem um fluxo de processamento muito semelhante, passando pela análise de uma sentença, removendo um conceito alvo, adicionando uma palavra chave da questão e convertendo o verbo para o formato gramaticalmente correto. A diferença entre a abordagem sintática e semântica está na etapa de transformações da sentença e nas regras que definem as modificações necessárias para tais, através da análise textual no nível correspondente.

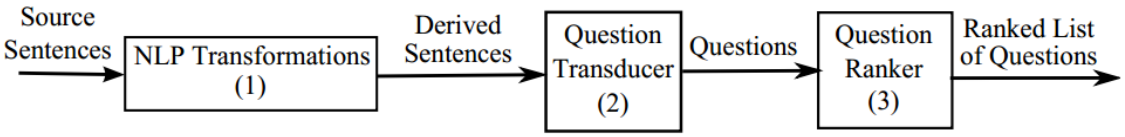
Tanto a abordagem semântica como a baseada em *template* existe uma dependência crescente de um domínio. É preciso ter um conhecimento prévio do tema abrangido pelo texto para definir adequadamente as regras de transformação, limitando seu uso. A utilização de ontologias na abordagem semântica possibilita a geração de questões através de análises mais profundas que se encontram diretamente ligadas ao texto escrito, podendo se utilizar de relações diferentes entre elementos, seus tipos e outras relações.

Os *templates* são modelos de sentenças desenvolvidos manualmente que servem como uma lista de verificações. Se uma sentença encaixar em um modelo, podemos gerar uma questão previamente definida para tal modelo, substituindo variáveis no texto. Tal método possibilita a geração de questões com parafraseamento, ou seja, as questões não refletem exatamente o texto da sentença geradora.

2.1. Processo de QG

O processo de geração de questões pode ser visto de forma genérica em poucos passos superficiais, porém, dependendo da abordagem utilizada, as etapas são ajustadas para realizar alguns detalhes específicos de cada abordagem. Heilman (2011) descreve um gerador de *wh-questions* com um fluxo simplificado de etapas (Figura 1) seguindo um modelo conhecido como *overgenerate-and-rank*, onde as sentenças são simplificadas, transformadas em questões e avaliadas com uma pontuação a fim de remover as inadequadas.

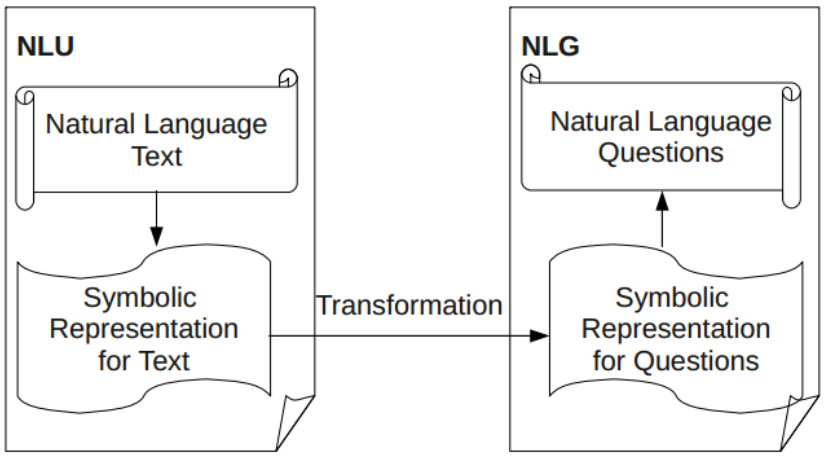
Figura 1 - *Framework* Para Geração de *WH-Questions*



Fonte: Heilman (2011, pg 45)

Em Yao, Bouma e Zahng (2012) foi definido um sistema baseado na abordagem semântica contendo apenas três etapas: transformação de texto em uma representação simbólica, conversão em uma representação simbólica para a questão e a transformação final da questão para linguagem natural, como pode ser visto na Figura 2.

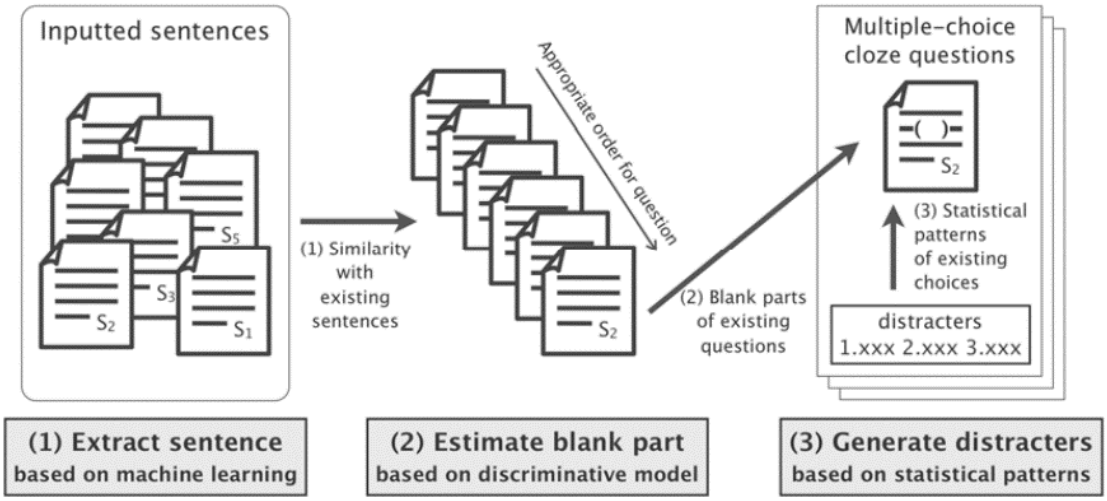
Figura 2 - Relação Entre NLU e NLG no Processo de Geração



Fonte: Yao, Bouma e Zahng (2012, pg 12)

Um processo simplificado para a geração de questões do tipo Cloze múltipla escolha também foi utilizado em Goto et al. (2009) com uma abordagem estatística para seleção dos componentes da questão. O processo também segue três etapas: seleção de uma sentença, estimativa de uma parte da sentença para remoção e geração dos distrativos (alternativas de resposta da questão), demonstrado na Figura 3.

Figura 3 - Processo de Geração de Questões Cloze Múltipla Escolha



Fonte: Goto et al. (2009, pg 2)

3. Sistema de Geração de Questões Cloze e WH em PT-BR

A seguir serão apresentados o modelo de geração de questões utilizado em Cordeiro (2016) e o modo de funcionamento do sistema desenvolvido.

3.1. Modelo

O sistema funciona através de um aplicativo Java de linha de comando com a leitura de um arquivo formato texto (.txt) do qual são extraídas as sentenças e gerando um arquivo texto de saída com uma lista das questões geradas. Existe a opção de escolher entre os tipos Cloze e *wh-question* ou ambos, podendo também ativar ou desativar a geração de uma quantidade de distrativos desejada.

Seguindo a classificação definida por Graesser, Rus e Cai (2008), o sistema tem como propósito o monitoramento do entendimento do tópico apresentado no texto por parte do leitor, com a intenção de possibilitar uma futura adequação para uma plataforma de estudos. O tipo de informação tratado é bastante abrangente e adequa-se em várias classes, abordando definições, especificações e complemento dos conceitos apresentados pelo texto, porém evita questões que requerem comparações e opiniões.

A análise textual é realizada através de um modelo probabilístico gerado com a utilização de algoritmos de aprendizado de máquina na ferramenta Apache OpenNLP com um conjunto de corpora da língua portuguesa do projeto Floresta Sintá(c)tica. A utilização dessa abordagem evita a criação manual de diversas regras para analisar as sentenças e suas estruturas sintáticas. Enquanto para a geração das questões é utilizada uma abordagem mista entre a sintática e semântica, baseando-se principalmente na identificação e classificação de entidades mencionadas no texto. O sistema segue um pouco a ideia de overgenerate-and-rank, porém não foi desenvolvida a parte que realiza a poda dos resultados indesejados.

3.2. Funcionamento

O aplicativo recebe as entradas necessárias na linha de comando e inicia com o *parsing* dos parâmetros. Os principais valores recebidos são o caminho do documento de texto de entrada e o caminho do arquivo de saída, porém também é possível modificar os modelos que serão utilizados para a análise textual, os formatos de questão (Cloze e WH) e a quantidade de distrativos que serão gerados para cada questão.

Com os modelos na memória, o sistema segue com a leitura e análise do arquivo de entrada. O texto é dividido em sentenças, e cada uma delas passa pelo processo de tokenização, dividindo-a em diversos tokens, que em geral são palavras, e cada um é analisado tendo suas informações como a classe morfológica, grau e número armazenados em um objeto. Uma lista desses tokens é guardada para cada sentença, também em uma outra estrutura.

A partir desse ponto, o sistema possui as sentenças em um formato que possam ser passadas para análise com o objetivo de obter uma árvore sintática com a representação de suas partes (sujeito, objeto, adjunto, etc.) e uma lista com as entidades mencionadas (pessoa, tempo, local, organização, evento).

3.2.1. Geração de Questões Cloze

No caso da geração das questões Cloze, apenas as entidades mencionadas são utilizadas. O algoritmo de geração é bem simples, apenas removendo os tokens que fazem parte da entidade da sentença original e substituindo-os por uma parte vazia. Como exemplo vamos utilizar a sentença exemplo número 6 em Cordeiro (2016, pg. 85).

(S.6) Stefano Evodio Assemani trabalhou na Biblioteca Apostólica Vaticana como intérprete de línguas orientais.

Nessa sentença, o sistema identifica duas entidades mencionadas: “Stefano Evodio Assemani” e “Biblioteca Apostólica Vaticana” a partir das quais são geradas as questões A17 e A18, respectivamente, com a omissão dos termos.

(A.17) _____ trabalhou na Biblioteca Apostólica Vaticana como intérprete de línguas orientais.

(A.18) Stefano Evodio Assemani trabalhou na _____ como intérprete de línguas orientais.

3.2.2. Geração de *WH-Questions*

Esse tipo de questão é muito mais complexo de ser gerado, pois requer determinadas regras definidas com intuito de executar a transformação da sentença original em um formato interrogativo. Essas regras são complicadas de serem definidas e implementadas programaticamente, pois dependem do idioma, e do resultado da análise sintática para identificar corretamente as partes que serão removidas, reposicionadas ou adicionadas na sentença.

Atualmente o sistema leva em consideração apenas sujeitos, adjuntos adverbiais e objetos preposicionais para a geração das *wh-questions*. O algoritmo funciona procurando por partes na árvore sintática com uma das três classificações e então verifica se existe alguma entidade mencionada no trecho. Se existir, a wh-word é escolhida dependendo do tipo da entidade, conforme mostra o Quadro 1, caso contrário, a expressão é ignorada.

Quadro 1 - CONDIÇÕES PARA A DEFINIÇÃO DA WH-WORD

TIPO DA ENTIDADE	WH-WORD
Place (local)	Onde?
Person (pessoa)	Quem?
Time (tempo)	Quando?
Numeric (numeral)	Quantos?
Outros	O que?

Fonte: produzido pelo autor

As entidades de tipo numérico deveriam conter um tratamento especial, pois o ideal seria identificar a medida utilizada (tamanho, tempo, etc.) para modificar a *wh-word* e fazer perguntas do tipo "quanto tempo?" ou "quanto pesa?", porém não foi desenvolvido nenhum tratamento do tipo.

Para modificadores das *wh-words* foi implementada a adição de preposições quando a entidade faz parte de um adjunto adverbial ou um objeto preposicional. O algoritmo procura por preposições "por", "para", "com", "de" e suas variações ("pelo", "da", etc.) gerando questões do tipo "por quem?", "com quem?" e "para onde?", dentre outras. Como exemplo temos a sentença exemplo número 4 em Cordeiro (2016, pg. 84).

(S.4) The Magic Cloak of Oz é um filme dirigido por J. Farrell MacDonald.

Nesse exemplo, a entidade "J. Farrell MacDonald" é removida e define-se a wh-word "quem". Porém, a parte "por J. Farrell MacDonald" é classificada como sendo um adjunto adverbial preposicional, sendo assim o sistema identifica a preposição "por" e a adiciona como modificador da wh-word, resultando na questão V4.

(V.4) Por Quem The Magic Cloak of Oz é dirigido?

Um dos pontos levantados com alguns exemplos foi a necessidade de remover o complemento do sujeito mantendo apenas o verbo, pois quando a expressão era mantida a questão gerada não fazia sentido. Esse caso ocorre no exemplo anterior, mas será explicado em um outro caso, com a sentença exemplo número 2 (CORDEIRO, 2016, pg. 83).

(S.2) Arumana no Kiseki é um jogo de videogame lançado pela Konami em 1987.

No exemplo S2, o sistema identifica a entidade Arumana no Kiseki, porém a simples remoção do termo resultaria na questão "O que é um jogo de videogame lançado pela Konami em 1987?". Talvez a sentença ideal seria a substituição da wh-word "o que" por "qual" e a troca do artigo indefinido "um" pelo artigo definido "o", porém apesar de parecer simples nesse exemplo, não é uma tarefa trivial e foi optado pela remoção do complemento, mantando apenas o verbo, resultando na questão A6.

(A.6) O que é lançado pela Konami em 1987?

O último detalhe da geração de uma wh-question é o reposicionamento de adjuntos adverbiais no início da sentença para o final. Esse procedimento foi uma decisão tomada porque normalmente nesses casos a expressão é separada por vírgula e quando transformada a sentença para forma interrogativa, a questão não ficava muito natural. Esse caso pode ser visualizado na sentença exemplo número 7 em Cordeiro (2016, pg. 86) e demonstrado a seguir.

(S.7) Em 1898, Lucy Maud Montgomery voltou para Cavendish para viver com a avó viúva.

Nessa sentença foi identificada a entidade "Lucy Maud Montgomery", definindo a *wh-word* "quem" por sua classificação "pessoa". Porém, se seguida a regra básica de substituição obtemos a questão "Em 1898, quem voltou para Cavendish para viver com a avó viúva?", decidida como não sendo tão adequada quanto a questão A22, o sistema faz um reposicionamento do adjunto adverbial para o final da sentença, removendo a vírgula.

(A.22) Quem voltou para Cavendish para viver com a avó viúva em 1898?

3.2.3. Geração de Distrativos

Como as questões Cloze são baseadas nas entidades mencionadas, a geração de distrativos não é um processo muito complexo. Durante a geração das questões são armazenadas as entidades encontradas, sem repetições, sendo assim possível utilizá-los após a geração para escolher as alternativas para cada questão. Desta forma, o sistema é capaz de gerar os distrativos sem a necessidade da utilização de uma ferramenta externa, pois o próprio texto provê as alternativas. Porém, caso o texto de entrada seja muito pequeno e não possua muitas entidades de um determinado tipo, as alternativas geradas serão muito semelhantes na maioria das questões.

As questões geradas são armazenadas em uma estrutura de dados contendo também a sua resposta, ou seja, a entidade mencionada removida da sentença original, sendo assim possível obter suas informações. Isso é um fato importante, pois podemos selecionar para as alternativas apenas entidades de mesmo tipo, levando a um resultado mais satisfatório, ou seja, com menor chance de erros semânticos. Outro ponto levado em consideração para a seleção são as informações dos tokens da entidade mencionada, evitando erros gramaticais graves de concordância verbal, de gênero e de número.

Para as *wh-questions*, porém, o cenário é um pouco diferente. Como a geração é orientada por partes sintáticas, a remoção do texto em boa parte dos casos vai além da entidade mencionada. Por esse motivo, as alternativas geradas não encaixam adequadamente como sendo uma resposta viável em alguns casos e durante o desenvolvimento optou-se por manter essa forma simplificada nestes casos.

4. Conclusão e Trabalhos Futuros

Este artigo apresentou de forma breve e resumida o trabalho desenvolvido em Cordeiro (2016), desde a definição teórica básica, os conceitos de processamento de linguagem natural, passando por uma explicação mais detalhada do modelo desenvolvido e o funcionamento do sistema implementado para geração de questões. Neste último capítulo serão resumidos os resultados obtidos e os trabalhos futuros citados pelo autor.

O aplicativo obteve sucesso e está funcionando com bons resultados para a geração de questões tipo Cloze abertas, porém quando adicionada a opção de geração de distrativos, algumas alternativas não são adequadas, apesar de exceções, os resultados ainda podem ser considerados satisfatórios, pois cumpriram o objetivo de gerar questões válidas de forma automatizada. A geração de *wh-questions* é muito básica e simplificada, funcionando corretamente apenas em sentenças pequenas e diretas, sem uma estrutura sintática complexa.

Para melhoria dos resultados, o autor sugere a utilização de modelos probabilísticos mais precisos, diminuindo a repercussão de tais falhas para os algoritmos

de transformação de sentenças em questões. Além disso, também seria de grande utilidade a implementação de um processo para simplificação das sentenças, evitando diversos erros na geração de questões factoide. Também ajudaria a gerar resultados mais confiáveis a implementação do ranking dos resultados, possibilitando remover as questões inadequadas e menos naturais da lista final apresentada ao usuário.

As tarefas citadas anteriormente ajudariam na qualidade dos resultados finais, porém também seria interessante a utilização de outros artifícios para a geração de uma variedade maior de questões. Dois pontos levantados pelo autor nesse quesito são a resolução de referências e a utilização de paráfrases, possibilitando a geração de questões com uma apresentação diferente (ordem das palavras, sinônimos, etc.) com o mesmo significado, mas com um formato mais natural para o ser-humano. A falta de resolução das referências no sistema faz com que diversas possíveis questões sejam omitidas, pois pronomes de anáfora não são considerados entidades mencionadas. Outra opção para aumentar a variedade seria a implementação de novos algoritmos para geração de outros tipos de questões, como verdadeiro ou falso e somatórios.

O sistema desenvolvido apresenta bons resultados iniciais para aplicações voltadas a educação, porém ainda é preciso resolver muitos desses problemas para que os resultados se tornem confiáveis e de alta qualidade.

Referências

CORDEIRO, Luís G. T. **Geração Automática de Questões Através de Análise de Texto**. Florianópolis, SC, Brasil. 2016.

GOTO, Takuya; KOJIRI, Tomoko; WATANABE, Toyohide; IWATA, Tomoharu; YAMADA, Takeshi. **An Automatic Generation of Multiple-choice Cloze Questions Based on Statistical Learning**. 2009.

GRAESSER, Arthur C.; RUS, Vasile; CAI, Zhiqiang. **Question classification schemes**. In: Proc. of the Workshop on Question Generation. 2008.

HEILMAN, Michael. **Automatic Factual Question Generation from Text**. School of Computer Science. Carnegie Mellon University. Pittsburgh, PA, Estados Unidos. 2011.

MÜLLER, DANIEL N. **Processamento de Linguagem Natural**. Porto Alegre, Rio Grande do Sul, Brasil. 2003.

SINCLAIR, John. **Corpus and Text: Basic Principles**. Tuscan Word Centre. Developing Linguistic Corpora: a Guide to Good Practice. 2004.

SMITH, Simon; AVINESH P. V. S. **Gap-fill Tests for Language Learners: Corpus-Driven Item Generation**. In: Proceedings of ICON-2010: 8th International Conference on Natural Language Processing. Índia. 2010.

YAO, Xuchen. **Question Generation With Minimal Recursion Semantics**. 2010.

YAO, Xunchen; BOUMA, Gosse; ZHANG, Yi. **Semantics-based Question Generation and Implementation**. 2012.

tCALC: Agrupamento de Currículos Lattes por Afinidade de Áreas de Conhecimento Considerando Temporalidade

Jaime Mendes da Silva

Universidade Federal de Santa Catarina
Florianópolis, BR.

jaimemnds@hotmail.com

Abstract. *Works published before have clustered Lattes curricula from science, technology and innovation's field's professionals through the application of data clustering algorithms [1]. The clusters generated by this process have evidenced information about the field they were working in and which of them were on the same field. The current project extends what have been done by analyzing the impact onto quality and performance caused by the consideration of the temporal aspect of the data in the curricula clustering. The inclusion of time in this application comes from the evidence found in the literature that the expertise retrieval applications have benefit from this inclusion [3]. The effort comes from the fact that professionals that worked in some research field in the past might no longer work on the same subject.*

Resumo. *Trabalhos realizados anteriormente, através de algoritmos de clustering de dados, agruparam currículos Lattes de profissionais da área de ciência, tecnologia e inovação [1]. Os grupos gerados por esse processo evidenciavam informações sobre a área de atuação desses profissionais e quais pertencem a uma mesma área. O presente trabalho estende o que foi realizado ao analisar o impacto de qualidade e performance causado pela consideração do fator tempo no processo de agrupamento dos currículos. A inclusão da temporalidade vem da evidência na literatura de que aplicações de busca por competências se beneficiaram da mesma [3]. A aplicação dá-se pelo fato de que profissionais que atuaram em determinada área do conhecimento no passado podem não ser mais atuantes na mesma.*

1. Introdução

Dados são elementos chave de sistemas computacionais [4]. A importância dos computadores nas atividades humanas, somada ao grande avanço tecnológico dos recursos computacionais e sua consequente redução de custos, trouxe como consequência um estado de geração rápida, variada e massiva de dados [4].

Produzir dados não garante por si só que informação seja adquirida a partir deles e que algum conhecimento seja obtido a partir dessa informação. Para preencher essa lacuna, surgiram diversas disciplinas que se propõem a tratar os dados de forma a torná-los úteis para

o uso humano em suas diversas aplicações. Entre elas, a Análise de Dados e suas diversas técnicas [7].

Uma importante técnica para o escopo deste trabalho é a Mineração de Dados (do inglês *data mining*). Ela consiste do processo de descoberta de padrões interessantes acerca de uma grande quantidade de dados [2]. Uma dos métodos de Mineração é o agrupamento (*clustering*), que procura particionar os dados evidenciando grupos que concentram todas aquelas amostras que demonstram comportamentos similares em relação a determinadas propriedades [2]. O agrupamento é um dos temas centrais deste trabalho.

A fonte de dados utilizada neste trabalho são currículos da plataforma Lattes, que reúne bases de dados de currículos profissionais da área de pesquisa, desenvolvimento e inovação no Brasil. A informação que será buscada a partir deles são as competências dos autores desses currículos.

Este trabalho parte da obra "CALC: Agrupamento de Perfil Científico por Afinidade de Áreas de Conhecimento Utilizando Currículos Lattes", por Bernardo de Farias Esteves (2015). A introdução do aspecto temporal ao problema que já havia sido abordado no projeto CALC dá-se com base em linhas recentes de pesquisa que propõem que o tempo é um fator com alto teor de relevância para avaliar a perícia da qual um profissional dispõe em relação a uma área [3].

2. Descrição do Processo

As atividades deste trabalho seguem as etapas definidas pelo processo de KDD (*Knowledge Discovery in Databases* - Descoberta de Conhecimento em Bancos de Dados). Esse processo foi desenvolvido para servir de ferramenta a auxiliar na extração de informações úteis do grande volume de dados digitais produzidos atualmente e, principalmente, de dados que, se considerados individualmente, possuam baixo teor informativo [6].

O KDD é composto por 5 fases que se fazem presentes no procedimento deste trabalho (seleção, pré-processamento, transformação, mineração de dados, interpretação/avaliação) que, embora propostas nessa sequência, são organizadas de maneira diferente conforme as necessidades do projeto, o que não afeta o resultado.

2.1. Seleção

A primeira etapa que o KDD sugere é a seleção, dentre todo o conjunto de dados disponível, daqueles que são relevantes para a produção do conhecimento. Essa etapa também pode considerar a seleção de dados adicionais de fontes externas [4].

O primeiro aspecto dessa etapa, em relação ao procedimento deste projeto, foi a definição da informação que se pretende extrair do conjunto de dados sobre os quais será trabalhado, ou seja, “o que queremos saber sobre os dados?”.

A resposta a essa pergunta herda seus fundamentos do projeto executado anteriormente por Esteves (2015), onde a resposta era “a área de atuação de um dado profissional” ou “quais profissionais atuam na mesma área”. Mas no caso atual a resposta se expande: queremos saber dos dados quais profissionais atuaram (ou têm competência) na mesma área em um mesmo momento do tempo.

A partir da diferença dessas propostas, foi decidido que o tCALC deveria tratar a etapa de Seleção de maneira diferente, buscando campos que evidenciassem as informações sobre área de atuação e período de atuação em paralelo. Para isso, decidiu-se utiliza a seção referente às publicações do profissional, que trazia essas informações de maneira consistente, mais especificamente os campos 'TITULO-DO-ARTIGO' e 'ANO-DO-ARTIGO'.

Como já se esperava, essa escolha mostrou fragilidades desde os primeiros experimentos que seguiram sua implementação. A pequena quantidade de dados disponível nesses dois atributos impactou diretamente na qualidade do agrupamento. Para melhorar isso, a base de dados *Lattes Expertise Retrieval* (LExR) [8] foi escolhida para a obtenção de dados adicionais acerca dos artigos considerados na entrada.

2.2. Pré-Processamento

É indicado que haja um tratamento adequado dos dados, sejam eles estruturados ou não, antes do processo de mineração [5]. No caso do projeto tCALC as atividades de pré-processamento se limitam à ação conhecida como remoção de *stopwords*. Essa ação é bastante comum em mineração de dados textuais e consiste na remoção daquelas estruturas das linguagens naturais que conectam as ideias dentro de uma frase mas que, por si só, não apresentam muito valor semântico para a mineração. Exemplos comuns desses conectivos são as preposições, artigos e pronomes [10].

2.3. Transformação

Após a fase de pré-processamento garante-se que os dados tenham adquirido uma condição consistente, mas não garante que eles estejam prontos para a mineração. A última ação a ser tomada antes de aplicar os algoritmos de mineração deve ser a transformação dos dados de seu formato original em um que os algoritmos de mineração aceitem como entrada, o que costuma variar entre diferentes implementações [5].

O tCALC executa a transformação através da exportação dos dados referentes ao campos 'TITULO-DO-ARTIGO' e 'ANO-DO-ARTIGO' para arquivos em disco. Após isso, é feita uma consulta no banco de dados referente ao LExR para cada artigo, incluindo no arquivo de texto as palavras-chave e áreas obtidas da base de dados.

2.4. Mineração

Esta é, provavelmente, a etapa mais sofisticada do ponto de vista computacional no KDD. Segundo Fayyad et al (1996), a "mineração de dados é a aplicação de algoritmos específicos para extrair padrões a partir de dados".

A mineração pode ser feita de diversas formas, através de diversas técnicas e ainda, cada técnica pode ser implementada por algoritmos diversos e por vezes bem distintos [4]. A técnica abordada no procedimento deste trabalho foi idealizada por Esteves (2015) e consiste no agrupamento de dados usando os algoritmos *BestStar* e *K-medoids* implementados pelo URSA, um *framework* que oferece algoritmos para cálculo de similaridade e agrupamento de dados e que permite a aplicação dessas funcionalidades sobre diversos tipos de dados [1].

O tCALC, a exemplo do CALC, utiliza o algoritmo *BestStar* inicialmente para estimar uma quantidade ótima de *clusters* a serem gerados para o conjunto de dados de entrada. Essa etapa é uma preparação para a execução do *K-medoids* já que esse segundo algoritmo espera que seja fornecida como entrada a quantidade de *clusters* que serão gerados para os dados, enquanto o *BestStar* não precisa dessa informação [1].

2.5. Análise/Interpretação

A última fase do processo é mais dependente de intervenção humana que as demais. As saídas dos algoritmos da fase anterior são tratadas de forma a evidenciar as informações obtidas através de gráficos ou qualquer outro tipo de representação que seja mais simpática à análise humana [4].

Após a disponibilização visual desses resultados, é necessário que o analista os avalie e compare na tentativa de obter informações evidenciadas pelos padrões formados ou, mediante a devida constatação, decida pela alteração do processo para uma nova tentativa de KDD [4].

Neste projeto, a aplicação gera como saída do agrupamento uma hierarquia de diretórios no sistema de arquivos que revelam os *clusters* formados. Essa árvore de diretórios se organiza com uma raiz chamada *Clusters*, dentro da qual existem pastas referentes a cada período de tempo considerado no agrupamento e denominadas com base no primeiro ano do intervalo considerado (por exemplo, a pasta referente ao triênio 2010-2012 é denominada 2010). Dentro da pasta de cada ano estão diretórios numerados referentes aos *clusters* gerados nos quais estão os currículos Lattes atribuídos a esses grupos.

As análises deste projeto foram feitas com base na saída descrita acima a fim de gerar deduções sobre a qualidade dos *clusters* gerados e da possibilidade de aprimorar o processo para melhorar os resultados.

3. Experimentos

A amostra de currículos utilizada possui 206 arquivos contendo currículos Lattes no formato XML de pesquisadores de programas de pós-graduação da UFSC das áreas: agronomia, aquicultura, bioquímica, ciência da computação, ciências humanas, enfermagem, engenharia de alimentos, engenharia civil, engenharia mecânica, farmácia, farmacologia, filosofia, física, história, literatura e química. Essa amostra foi a mesma utilizada na implementação do CALC e foi escolhida pelo autor daquele projeto de forma a possuir elementos de áreas distintas mas também de áreas com alguma semelhança para analisar efeitos que mudanças nos parâmetros causariam aos resultados do agrupamento [1].

Os currículos da amostra são identificados pelos nomes das áreas aos quais pertencem, seguidos de números para diferenciar uns dos outros (por exemplo, Agronomia 3 ou Aquicultura 1). Alguns outros currículos, pertencentes a profissionais de Ciência da Computação, são denominados com os nomes dos profissionais que retratam. Essas estratégias foram tomadas para permitir que, na etapa de Análise, seja possível distinguir quais *clusters* possuem currículos semelhantes de fato [1].

3.1. Experimentos sem LExR

Nesta fase os experimentos foram elaborados buscando avaliar a qualidade do agrupamento ainda sem a intergração com a base de dados externa e, para isso, utilizou-se o lote de amostras com 206 currículos.

O experimento realizado, considerou triênios para agrupamento. Ou seja, fez um agrupamento para cada triênio contido no período no qual a amostra trouxe artigos publicados.

3.2. Experimentos com LExR

Os experimentos desta etapa foram bastante semelhantes aos anteriores, visto que a única mudança – embora substancial – foi a adição dos dados da base de dados externa LExR no processo. Com os dados adicionais de palavras-chave e áreas dos artigos do Lattes, foi possível aumentar a expressividade dos dados de entrada.

A próxima subseção se preocupa em avaliar a qualidade alcançada nesses experimentos.

3.3. Análise de Resultados

A análise dos resultados deste projeto dá-se de forma puramente argumentativa com base nos experimentos apresentados.

Os currículos fornecidos como entrada da aplicação foram nomeados com as áreas dos profissionais que eles representam com o intuito de facilitar a observação de *clusters* gerados com sucesso ou não. A argumentação presente nestas análises parte da definição que as seguintes áreas presentes no conjunto de entrada são mutuamente próximas no ponto de vista da engenharia de conhecimento:

- Química, Bioquímica, Farmácia e Farmacologia;
- Agronomia e Aquicultura;
- Ciências Humanas, Filosofia, História e Literatura;
- Física e Engenharia Civil;
- Física e Engenharia Mecânica;
- Física e Química;
- Química, Física e Engenharia de Alimentos.

Outras relações não listadas podem ser verdadeiras mas não foram consideradas nas análises a seguir.

As análises foram feitas para todos os *clusters* gerados nos experimentos e levaram em consideração a taxa de agrupamentos bem-sucedidos – ou seja, aqueles que colocam em um mesmo grupo currículos de áreas próximas conforme a lista acima – e, para quando ocorrer, a quantidade de subgrupos formados no *cluster*. Os resultados com ou sem a integração com LExR são comparados ao final de cada experimento.

Os dados abaixo dizem respeito ao triênio entre 2011 a 2013. E foram obtidos pela implementação sem uso do LExR:

- Quantidade total de currículos: 165;
- Quantidade total de *clusters*: 11;
- Quantidade de currículos agrupados corretamente: 127;
- Quantidade de currículos agrupados incorretamente: 38;
- Quantidade de subgrupos: 10 divididos entre 4 grupos;
- Taxa de sucessos: 77%;
- Taxa de fracassos: 23%.

Os dados obtidos pela execução integrada ao LExR são:

- Quantidade total de currículos: 165;
- Quantidade total de *clusters*: 14;
- Quantidade de currículos agrupados corretamente: 142;
- Quantidade de currículos agrupados incorretamente: 23;
- Quantidade de subgrupos: 16 divididos entre 7 grupos;
- Taxa de sucessos: 86%;
- Taxa de fracassos: 14%.

Portanto, com base nas variáveis enumeradas acima, pode-se concluir que há uma melhora com o uso do LExR no experimento. A taxa de sucessos de 86% é considerada satisfatória visto as limitações apresentadas pela base de dados tais como a escassez de termos e campos com informação temporal. A grande quantidade de subgrupos evidencia que, se fosse assumido um número maior de grupos, esses subgrupos se manifestariam como grupos por si só, o que indica que a quantidade ótima de grupos deve ser maior que a escolhida pelo algoritmo. Isso pode abrir espaço para otimizações no processo.

4. Considerações Finais

A aplicação produzida juntamente com esta obra se preocupou, a exemplo de sua antecessora, em oferecer a solução a uma demanda inerente à proposta da plataforma Lattes, o que literatura chama de *expertise retrieval* ou recuperação de especialidades em uma tradução livre. Além disso, o projeto em si buscou apresentar a importância da mineração de dados, da descoberta de conhecimento e da análise de competências.

Até esses requisitos apresentados anteriormente, o projeto tCALC apenas expandiu o que já havia sido abordado no CALC. A diferença fundamental e que permitiu que um novo escopo fosse trabalhado – o aspecto temporal – foi abordada com protagonismo compatível com a relevância dela para o problema apresentado. Ao optar-se por levar em consideração o tempo no *expertise retrieval*, a proposta muda radicalmente tal como os resultados atingidos. E isso foi o que este projeto se preocupou em deixar claro através de seus capítulos de experimentos e análises.

Ainda há muito o que ser produzido pela comunidade científica nas áreas às quais este trabalho pertence. Ainda assim, a expectativa é de que, de alguma forma, esta obra tenha contribuído, seja para gerar conhecimento acerca do assunto ou motivação para que outros projetos avancem cada vez mais o estado da arte.

Referências Bibliográficas

[1] B. F. ESTEVES. *CALC: Agrupamento de Perfil Científico por Afinidade de Áreas de Conhecimento Utilizando Currículos Lattes*. Florianópolis: Universidade Federal de Santa Catarina. 2015.

[2] M.J. ZAKI, and W. MEIRA. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. 1st ed. New York: Cambridge UP, 2014.

[3] Y. LI, and J. TANG. *Expertise Search in a Time-varying Social Network*. Beijing: Tsinghua University, 2008.

[4] U. FAYYAD, G. PIATETSKY-SHAPIRO, P. SMYTH, and R. UTHURUSAMY. *Advances in knowledge discovery and data mining*. American Association for Artificial Intelligence, Menlo Park, CA, USA 1-34. 1996.

[5] S. GARCÍA, J. LUENGO, and F. HERRERA. *Data Preprocessing in Data Mining.* In *Intelligent Systems Reference Library vol. 72.* New York: Springer, 2015.

[6] A. ADHIKARI, and J. ADHIKARI. *Advances in Knowledge Discovery in Databases.* In *Intelligent Systems Reference Library vol. 79.* New York: Springer, 2015.

[7] C.R. SHALIZI. *Advanced Data Analysis from an Elementary Point of View*. New York: Cambridge UP. 2016.

[8] V. MAGARAVITE, R.L.T. SANTOS, I.S. RIBEIRO, M.A. GONÇALVES, A.H.F. LAENDER. *The LExR Collection for Expertise Retrieval in Academia.* Department of Computer Science, Universidade Federal de Minas Gerais. Belo Horizonte, MG, Brazil.

Utilização de QoC para melhorar o cenário experimental de sensores biomédicos para suporte às aplicações móveis distribuídas

Pedro José de Campos[1], Mario Antonio Ribeiro Dantas[2], Eduardo Camilo Inacio[2]

[1]Instituto de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

[2]Laboratório de Pesquisa em Sistemas Distribuídos (LaPeSD)
Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC - Brasil

pedro.campos@grad.ufsc.br, mario.dantas@ufsc.br,
eduardo.camilo@posgrad.ufsc.br

Resumo. *Este artigo descreve como utilizar Qualidade de Contexto (QoC) em aplicações móveis distribuídas, mais especificamente em Ambient Assisted Living (AAL), e a partir de parâmetros de contexto encontrar problemas em determinadas situações e com isso poder melhorar cenário dos sensores envolvidos com o ambiente. Para a análise, foi utilizado dois ambientes: um com sensor de temperatura e outro com sensor de pressão. Além disso foi utilizado também três cenários, cada um com uma determinada configuração de sensor. Os resultados foram obtidos através da aplicação dos três cenários nos dois ambientes. E depois foi feita uma análise dos resultados.*

1. Introdução

Computação ubíqua é um termo usado para descrever a onipresença de um sistema computacional no cotidiano. O termo foi definido pelo cientista Mark Weiser (1991), para se referir a dispositivos conectados em todos os lugares de maneira transparente para o ser humano. Em outras palavras, são dispositivos, portáteis que fazem parte do nosso dia a dia, capturando e processando diversas informações.

Ambient Assisted Living (AAL) compreende conceitos interoperáveis, produtos e serviços, que combinam novas informações de comunicação e com o objetivo de melhorar e aumentar a qualidade de vida das pessoas. Tais produtos e serviços podem ser esses dispositivos pequenos, baratos, robustos e sem fio. Dessa forma, é possível, com um baixo custo, fazer um bom monitoramento, dados a capacidade de processamento, recursos de comunicação e o armazenamento de dados, de um determinado ambiente previamente conhecido.

O conceito de interoperabilidade é caracterizado por uma "atuação de pedido", ou seja, uma entidade manda um pedido ou uma resposta para a entidade que a solicitou [Chen e Doumeingts 2003][Cheng 2005].

Qualidade de contexto é o termo referente à qualidade da informação e depende do contexto no qual está inserido. Em outras palavras, significa o quão boa a informação

é. Assim, qualidade de contexto refere-se à informação e não ao processo, nem ao componente de hardware que fornece as informações[Nazário 2015].

Devido a tais importâncias dessas aplicações, é necessário um cenário experimental que melhore o desempenho dos sensores, essas configurações podem ser obtidas através de avaliações da qualidade de contexto.

2. **Computação Ubíqua e AAL**

2.1 **Computação Ubíqua**

A ideia de computação ubíqua surgiu recentemente. Há um pouco mais de uma década atrás Mark Weiser, considerado por muitos o pai da computação ubíqua, dissertava sobre o tema, dizendo que a computação sairia do âmbito do trabalho e dos PCs pessoais, e iria migrar para objetos mais comuns no cotidiano, de maneira imperceptível ao usuário como por exemplo, etiquetas de roupas, xícaras de café, interruptores de luz, canetas e etc. Nesse novo mundo proposto por Weiser, devemos aprender a conviver com computadores e não apenas a interagir com eles [Weiser 1991].

Daí pode-se observar o termo utilizado, computação ubíqua ou computação pervasiva. Pervasiva significa infiltrada, espalhada; já ubíqua significa onipresente. Somando-se as duas ideias temos que computação ubíqua é onde os computadores estão distribuídos pelo ambiente de maneira onipresente, ou seja, está em todo lugar, mas de forma imperceptível aos usuários.

A computação ubíqua é um paradigma caracterizado pela presença de dispositivos portáteis, que estão cada vez mais fazendo parte do dia-a-dia das pessoas. Estes dispositivos possuem uma considerável capacidade de processamento, recursos de comunicação sem fio e armazenamento de dados. Possuem funcionalidades diversificadas e interfaces como GPS, rádio e TV, tocadores de áudio, câmeras digitais entre outros, sendo utilizados em aplicações de diversas áreas como: indústria, comércio, turismo, saúde, entretenimento. Este tipo de computação possui forte ligação com as características do mundo físico, bem como aquelas apresentadas pelos perfis de seus usuários. Tais informações são chamadas de contextos e representam o elemento de entrada para a computação ciente ou sensível ao contexto [Loureiro 2009].

2.2 **AAL**

Ambientes monitorados ou Ambient Assisted Living (AAL) se baseia na interoperabilidade de conceitos, produtos e serviços, que combinados geram novas tecnologias de informação e comunicação (TIC) em ambientes sociais, com o objetivo de melhorar e aumentar a qualidade de vida das pessoas em todas as fases do ciclo de vida [Pieper; Antona e Cortés 2011].

AAL então nada mais é do que, um ambiente monitorado por computadores que possuem um sistema capaz de obter dados do ambiente e, a partir desses dados, oferece suporte aos usuários, que no caso usufruem deste ambiente. Com o conceito apresentado de computação ubíqua, imagina-se AAL com utilização de dispositivos pervasivos (sensores) para o usuário.

Com o aumento da idade chegam os novos desafios à população idosa devido ao declínio de suas funções cognitivas, doenças crônicas relacionadas à idade, bem como, limitações nas atividades físicas, visão e audição. As tecnologias de ambientes inteligentes podem ser usadas para monitorar o estado de saúde de idosos com doenças crônicas e fornecer um cuidado com assistência para indivíduos com limitações físicas ou mentais [Acampora 2013].

3. Qualidade de Contexto

A definição de QoC para [Krause e Hochstatter 2005] é qualquer informação inerente que descreve informação de contexto e pode ser usada para determinar o valor da informação para uma aplicação específica. Isso inclui informações sobre o processo de provisionamento que a informação foi submetida (“histórico”, “idade”), mas não tratam de estimativas sobre os passos de provisionamentos futuros.

No trabalho [Buchholz; Kupper e Schiffers 2003] a Qualidade de contexto (QoC) descreve a qualidade da informação que é usada como de contexto. Assim, QoC refere-se à informação e não ao processo, nem ao componente de hardware que fornece as informações.

Ainda relacionado à qualidade da informação, no trabalho de [Buchholz; Kupper e Schiffers 2003] é feito uma relação entre as dimensões de QI e parâmetros de QoC; os autores apresentam algumas justificativas para a necessidade de QoC. Uma delas é que QoC é um indicador valioso para selecionar um provedor de contexto apropriado. O Provedor de CAS pode selecionar um provedor de contexto adequado com base na QoC oferecida e no preço da informação de contexto. Outra justificativa é que QoC permite especificar as políticas de acesso de uma forma mais refinada. Sem QoC o proprietário do contexto só poderia determinar quem tem permissão para acessar parte de seu contexto. Com QoC, por exemplo, um proprietário de contexto pode conceder a permissão de que um determinado grupo pode acessar sua localização atual, mas apenas com uma precisão de 10 quilômetros e com um atraso de algumas horas. Assim, QoC permite políticas de privacidade mais sofisticadas.

4. Proposta

Utilizar o Siafu, que é um simulador de contexto open source. E a partir dos ambientes e cenários propostos neste trabalho, dentro desses ambientes realizar uma quantificação e uma avaliação dos parâmetros de QoC, analisando assim cada cenário, conforme a Figura 1.

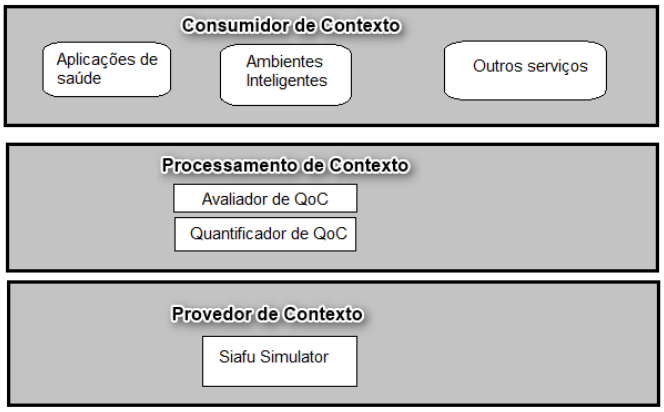


Figura 1. Método proposto. Adaptado de [Acampora 2013].

4.1. Parâmetros propostos

O trabalho [Kim e Lee 2006] mostra uma relação entre parâmetros de contexto e dimensões de qualidade de informação. E obtém os seguintes parâmetros: Trust-worthiness, Up-to-dateness, Accuracy.

Com base nesse trabalho foi proposto a utilização destes parâmetros em conjunto com o Completeness.

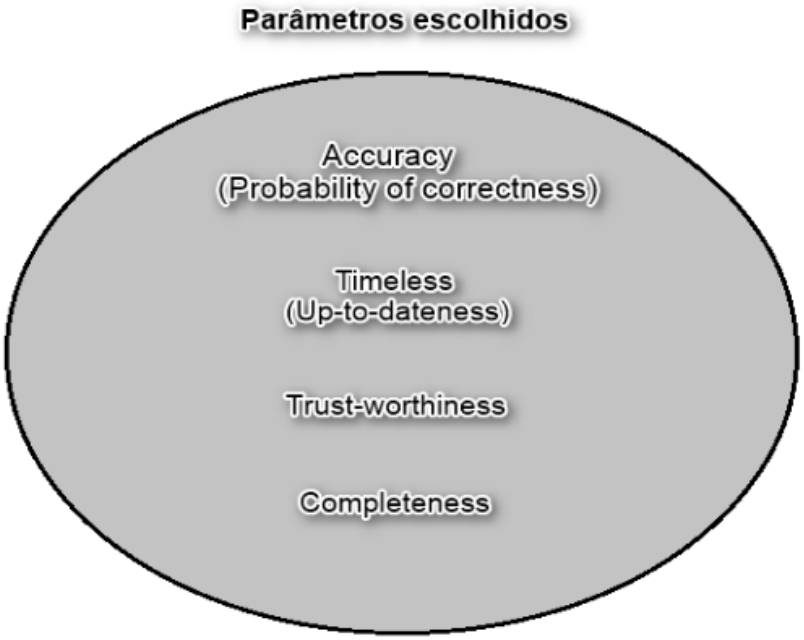


Figura 2. Parâmetros propostos

1. **Accuracy**: uma medida dos dados serem corretos e confiáveis; probabilidade de uma parte da informação de contexto estar correta [Kim e Lee 2006] (outros autores utilizam Probability of correctness com este significado);
2. **Timeliness**: é a faixa de erro em termos de tempo de alguns fenômenos [Gray e Salber 2001]; para os autores de [Ren e Seung 2009] está relacionado com a idade das

informações recebidas, onde informações mais recentes geralmente são mais relevantes em relação às mais velhas;

3. **Trustworthiness**: descreve a probabilidade da informação fornecida ser correta. É utilizado pelo provedor de contexto para avaliar a qualidade do agente a partir do qual o prestador de contexto originalmente recebe informação de contexto [Buchholz; Kupper e Schiffers 2003];

4. **Completeness**: é o grau em que as informações de contexto estão disponíveis, suficientes e não ausentes[Kim e Lee 2006].

4.2 Quantificação do QoC

Segundo o estudo de trabalho correlatos foi possível designar formulas matemáticas para quantifica cada um dos parâmetros de QoC.

Accuracy = PorcentagemDeAcerto ÷ MinPorcentagemDeAcerto

Onde, *PorcentagemDeAcerto* representa a porcentagem de acerto do provedor contexto (sensor) e *MinPorcentagemDeAcerto* representa a porcentagem mínima de acerto definida pelo usuário. Se a razão é maior do que 1, a acurácia pode ser boa[51].

Up-to-Dateness (Timeliness) [Manzoor; Truong e Dustdar 2008]:

$$idade = |tempo\ da\ informacao - tempo\ atual|$$

Se $idade < tempo\ de\ vida$

$$U(O) = 1 - \frac{idade}{tempo\ de\ vida}$$

Senão

$$U(O) = 0$$

A variável *tempo de vida* é definida com um valor em que a informação se torna desatualizada, obsoleta, por exemplo:

tempo_de_vida = 10;

Exemplos, aplicando a fórmula:

idade = 0 → U = 1;

idade = 5 → U = 0.5;

idade = 10 → U = 0;

Trust-worthiness:

O [Manzoor; Truong e Dustdar 2008] define também como calcular o trust-worthiness de um objeto de contexto, dado como T(O).

Se $d(S,\mathcal{E}) < dmax$

$$T(O) = \left(1 - \frac{d(S,\mathcal{E})}{dmax}\right) * \delta$$

Senão

$$T(O) = 0$$

Onde $d(S,\mathcal{E})$é a distância entre o sensor e a entidade para onde o sensor envia os dados. E $dmax$ é a distância máxima em que se pode confiar nos dados do sensor. O δ é a acurácia do sensor. Dessa forma, dependendo da distância do sensor os dados serão mais ou menos confiáveis.

Completeness:

De acordo com [Manzoor; Truong e Dustdar 2008], esta medida de qualidade indica a quantidade de informação provida por um objeto de contexto. É a relação entre o número de atributos disponíveis e o total de atributos de um objeto de contexto, neste caso um sensor. O cálculo leva em consideração os atributos disponíveis e o peso de cada um dos atributos e está representado a seguir

$$CM(O) = \frac{\sum(Todos\ atributos\ disponiveis)}{\sum(Peso\ de\ todos\ atributos)}$$

QoC geral:

Neste trabalho o cálculo do QoC geral vai ser baseado no trabalho [Nazário 2015], tirando a média dos valores calculados citados acima:

$$QoC = \frac{Ac + U + T + Cm}{4}$$

4.3 Avaliação do QoC

O valor geral de QoC quantificado deve indicar se a qualidade das informações obtidas é adequada, neste caso o contexto é utilizado provendo uma adaptação mais precisa [Nazário 2015].

Quando um problema de qualidade é detectado, ou seja, o valor de QoC geral não está adequado, espera-se que o conjunto de parâmetros utilizado possibilite uma análise para a identificação do problema ocorrido. Por exemplo, se os valores não estiverem dentro de uma faixa esperada é possível que a acurácia (parâmetro Accuracy) esteja com um valor baixo. Então possivelmente existe algum problema no sensor e/ou o sensor pode estar tão longe que a informação recebida pode não ser confiável (Trust-worthiness). Informações não disponíveis (Completeness) podem indicar que um determinado sensor caiu da pessoa ou até mesmo parou de funcionar. desatualizadas (Timeless) podem indicar uma falha na comunicação do sensor.

5. Resultados

Previamente foi feito três cenários de possíveis configurações de sensores, como mostra a Tabela 1:

Tabela 1. Cenários prévios

	Cenário 1	Cenário 2	Cenário 3
Precisão do sensor	75%	90%	90%
Alcance (unidades)	10	30	30
Prob. de atualizar	40%	70%	15%
Prob. de ler zero	30%	5%	80%

Conforme a tabela acima, o cenário 2 é o que apresenta uma melhor configuração de sensor. Com uma precisão, alcance e chance de atualizar altos, e probabilidade de ler zero (falhar) pequena. Com isso espera-se uma melhor qualidade nos dados do cenário 2.

Foram feitos simulações de 2 horas para cada cenário, onde a cada 30 segundos o sensor de pressão e temperatura liam um novo valor; bem como a cada 30 segundos eram recalculados os valores de cada parâmetro de contexto e o valor geral do QoC.

Fazendo uma média, com arredondamento de três casas decimais, de cada um dos parâmetros e do valor geral do QoC nessas 2 horas simulados obteve-se os seguintes gráficos mostrados na Tabela 2:

Tabela 2. Resultados obtidos

	Média Ac	Média T	Média Up	Média Cm	Média QoC
Cenário 1 temperatura	0.273	0.317	0.982	0.346	0.489
Cenário 2 temperatura	0.7	0.699	0.994	0.737	0.783
Cenário 3 temperatura	0.098	0.668	0.559	0.108	0.358
Cenário 1 pressão sistólica	0.589	0.327	0.981	0.718	0.641
Cenário 1 pressão diastólica	0.558		0.986		
Cenário 2 pressão sistólica	0.901	0.655	0.987	0.946	0.858
Cenário 2 pressão diastólica	0.891		0.969		

Observando a tabela pode-se afirmar que o cenário 2 possui um QoC geral maior, conforme o esperado. Analisando os parâmetros, podemos perceber que accuracy foi baixa no cenário 1 e 3, detectando assim falhas no sensor, no cenário 1 a precisão do sensor era menor do que os demais cenários. Porém no cenário 3 a precisão era alta, mas a chance de falha era de 80%, ou seja, em 80% das leituras o valor era zero, o que diminui muito a accuracy. Além da accuracy, a alta chance de falha também deteriorou o valor do Completeness, já que em 80% dos casos o valor estava ausente/incompleto.

Trust-worthiness foi alta nos cenários 2 e 3, pois o alcance deles era o triplo do cenário 1. Up-to-Dataness foi baixa apenas no cenário 3, pois esta tinha uma chance de 15% de atualizar, então, em 75% dos valores lidos pelo sensor estavam desatualizados.

O parâmetro Completeness agrega mais valor para o sensor de pressão, pois para a pressão ele não verifica somente se existe um valor ou não:

- Se (diastólica E sistólica disponíveis) → Cm(pressão) = 1;

- Se (diastólica OU sistólica disponíveis) → Cm(pressão) = 0.5;

- Se (diastólica E sistólica indisponíveis) → Cm(pressão) = 0;

Pode-se perceber que nos dois cenários o parâmetro Completeness foi bem superior para o sensor de pressão, pois o parâmetro em si faz muito mais sentido, agrega mais valor de informação nesse contexto. Consequentemente o valor geral do QoC foi maior também para os sensores de pressão.

Além disso, é possível perceber também que o parâmetro Accuracy, também sofreu grande alteração entre os cenários. Sendo maior no caso de pressão. No caso do sensor de temperatura, sempre que o Completeness era igual a zero, Accuracy também era zero, pois se a informação não existia, ela estava incompleta e com "zero" de precisão. Portanto, é possível que haja uma certa semelhança entre esses parâmetros.

6. Conclusão e trabalhos futuros

Através do Siafu, foi desenvolvido um cenário de um parque com um agente caminhando no ambiente. O agente possui sensores de temperatura e pressão, lendo a cada 30 segundos um novo valor. Com isso, realizou-se uma pesquisa na base de dados feita pelo trabalho [Nazário 2015] para a escolha dos quatro parâmetros propostos, bem como suas respectivas quantificações e avaliações.

Para realização dos testes, escolheu-se alguns cenários (configurações do sensor) e diante dessa configuração, já era possível imaginar os resultados possíveis. Foi realizado nesses cenários, simulações de duas horas, gerando dados a cada 10 segundos em um arquivo CSV. Trabalhando no arquivo CSV tirou-se os valores médios de cada parâmetro, e em cima desses valores foi feita a avaliação de contexto, consolidando as configurações propostas previamente. Outras considerações feitas em cima desses arquivos CSV foram que a escolha dos parâmetros pode influenciar numa baixa qualidade de contexto.

Para trabalhos futuros pode ser feito uma mineração desses dados CSV, a fim de obter relações entre parâmetros de contextos, ou até mesmo quais parâmetros são mais significantes em determinados ambiente ou em determinados sensores; para efeitos de data mining seriam necessários conjuntos de dados maiores.

Também pode ser direcionado a consolidação do método de quantificação e avaliação de dados utilizando sensores, pessoas e ambientes reais, tendo em vista que o mundo real possui certas limitações que a simulação acaba por ocultar.

Por fim, testar outros parâmetros da literatura, podendo observar outras causas de problemas de QoC.

Referências

WEISER, Mark. The computer for the 21st century. **Scientific american**, v. 265, n. 3, p. 94-104, 1991.

CHEN, David; DOUMEINGTS, Guy. European initiatives to develop interoperability of enterprise applications basic concepts, framework e roadmap. **Annual Reviews in Control**, v. 27, n. 2, p. 153-162, 2003.

CHEN, David. Practices, principles e patterns for interoperability. 2005.

NAZÁRIO, Débora Cabral et al. Cuida: um modelo de conhecimento de qualidade de contexto aplicado aos ambientes ubíquos internos em domicílios assistidos. 2015.

LOUREIRO, Antonio Alfredo Ferreira et al. Computação ubíqua ciente de contexto: Desafios e tendências. **27o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. Anais**, p. 99-149, 2009.

PIEPER, Michael; ANTONA, Margherita; CORTÉS, Ulisses. Introduction to the Special theme Ambient Assisted Living. **Ercim News,** Londres, p.18-19, out. 2011. Mensal. Disponível em: <http://ercim-news.ercim.eu/en87>. Acesso em: 12 nov. 2016.

ACAMPORA, Giovanni et al. A survey on ambient intelligence in healthcare. **Proceedings of the IEEE**, v. 101, n. 12, p. 2470-2494, 2013.

KRAUSE, Michael; HOCHSTATTER, Iris. Challenges in modelling e using quality of context (qoc). In: **International Workshop on Mobile Agents for Telecommunication Applications**. Springer Berlin Heidelberg, 2005. p. 324-333.

BUCHHOLZ, T.; KUPPER, A.; SCHIFFERS, M. Quality of context: what it is e why it need. In: **Proc. of the Workshop of the HP OpenView University Association 2003 (HPOVUA2003)**. 2003.

KIM, Younghee; LEE, Keumsuk. A quality measurement method of context information in ubiquitous environments. In: **2006 International Conference on Hybrid Information Technology**. IEEE, 2006. p. 576-581.

GRAY, Philip; SALBER, Daniel. Modelling e using sensed context information in the design of interactive applications. In: **Engineering for Human-Computer Interaction**. Springer Berlin Heidelberg, 2001. p. 317-335.

REN, Lim Luo; SEUNG, Quah Jon Tong. Towards context information refinement for proximity mobile service using quality of context. In: **Proceedings of the 6th International Conference on Mobile Technology, Application & Systems**. ACM, 2009. p. 39.

MANZOOR, Atif; TRUONG, Hong-Linh; DUSTDAR, Schahram. On the evaluation of quality of context. In: **European Conference on Smart Sensing e Context**. Springer Berlin Heidelberg, 2008. p. 140-153.

Redes Neurais Convolucionais de Profundidade para Reconhecimento de Textos em Imagens de CAPTCHA

Vitor Arins Pinto[1]

[1]Departamento de Informática e Estatística - Universidade Federal de Santa Catarina (UFSC)
Santa Catarina – SC – Brazil

vitor.arins@grad.ufsc.br

Abstract. *Currently many applications on the Internet follow the policy of keeping some data accessible to the public. In order to do this, it's necessary to develop a portal that is robust enough to ensure that all people can access this data. But the requests made to recover public data may not always come from a human. Companies specializing in Big data have a great interest in data from public sources in order to make analysis and forecasts from current data. With this interest, Web Crawlers are implemented. They are responsible for querying data sources thousands of times a day, making several requests to a website. This website may not be prepared for such a great volume of inquiries in a short period of time. In order to prevent queries to be made by computer programs, institutions that keep public data invest in tools called CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). These tools usually deal with images containing text and the user must enter what he or she sees in the image. The objective of the proposed work is to perform the text recognition in CAPTCHA images through the application of convolutional neural networks.*

Resumo. *Atualmente, muitas aplicações na Internet seguem a política de manter alguns dados acessíveis ao público. Para isso é necessário desenvolver um portal que seja robusto o suficiente para garantir que todas as pessoas possam acessá-lo. Porém, as requisições feitas para recuperar dados públicos nem sempre vêm de um ser humano. Empresas especializadas em Big data possuem um grande interesse em fontes de dados públicos para poder fazer análises e previsões a partir de dados atuais. Com esse interesse, Web Crawlers são implementados. Eles são responsáveis por consultar fontes de dados milhares de vezes ao dia, fazendo diversas requisições a um website. Tal website pode não estar preparado para um volume de consultas tão grande em um período tão curto de tempo. Com o intuito de impedir que sejam feitas consultas por programas de computador, as instituições que mantêm dados públicos investem em ferramentas chamadas CAPTCHA (teste de Turing público completamente automatizado, para diferenciação entre computadores e humanos). Essas ferramentas geralmente se tratam de imagens contendo um texto qualquer e o usuário deve digitar o que vê na imagem. O objetivo do trabalho proposto é realizar o reconhecimento de texto em imagens de CAPTCHA através da aplicação de redes neurais convolucionais.*

Introdução

Com o aumento constante na quantidade de informações geradas e computadas atualmente, percebe-se o surgimento de uma necessidade de tornar alguns tipos de dados acessíveis a um público maior. A fim de gerar conhecimento, muitas instituições desenvolvem portais de acesso para consulta de dados relevantes a cada pessoa. Esses portais, em forma de aplicações na Internet, precisam estar preparados para receber diversas requisições e em diferentes volumes ao longo do tempo.

Devido a popularização de ferramentas e aplicações especializadas em Big data, empresas de tecnologia demonstram interesse em recuperar grandes volumes de dados de diferentes fontes públicas. Para a captura de tais dados, *Web crawlers* são geralmente implementados para a realização de várias consultas em aplicações que disponibilizam dados públicos.

Para tentar manter a integridade da aplicação, as organizações que possuem estas informações requisitadas investem em ferramentas chamadas CAPTCHA (teste de Turing público completamente automatizado para diferenciação entre computadores e humanos). Essas ferramentas frequentemente se tratam de imagens contendo um texto qualquer e o usuário precisa digitar o que vê na imagem.

Objetivo

O objetivo geral do artigo é analisar o treinamento e aplicação de redes neurais convolucionais de profundidade para o reconhecimento de texto em imagens de CAPTCHA. Com isso será retratada a ineficiência de algumas ferramentas de CAPTCHA, mostrando como redes neurais convolucionais podem ser aplicadas em imagens a fim de reconhecer o texto contido nestas imagens.

Conceitos teóricos

Classificador Logístico

Um classificador logístico (geralmente chamado de regressão logística[Bengio and Courville 2016]) recebe como entrada uma informação, como por exemplo os pixels de uma imagem, e aplica uma função linear a eles para gerar suas predições. Uma função linear é apenas uma grande multiplicação de matriz. Recebe todas as entradas como um grande vetor que será chamado de "X", e multiplica os valores desse vetor com uma matriz para gerar as predições. Cada predição é como uma **pontuação**, que possui o valor que indica o quanto as entradas se encaixam em uma classe de saída.

$$WX + b = Y \tag{1}$$

Na equação 1, "X" é como chamaremos o vetor das entradas, "W" serão pesos e o termo tendencioso (*bias*) será representado por "b". "Y" corresponde ao vetor de pontuação para cada classe. Os pesos da matriz e o *bias* é onde age o aprendizado de máquina, ou seja, é necessário tentar encontrar valores para os pesos e para o *bias* que terão uma boa performance em fazer predições para as entradas.

Função *Softmax*

Como cada imagem pode ter um e somente um rótulo possível, é necessário transformar as pontuações geradas pelo classificador logístico em probabilidades. É essencial que a probabilidade de ser a classe correta seja muito perto de **1.0** e a probabilidade para todas as outras classes fique perto de **0.0**. Para transformar essas pontuações em probabilidades utiliza-se uma função chamada ***Softmax***[Bengio and Courville 2016].

One-Hot Encoding

Para facilitar o treinamento é preciso representar de forma matemática os rótulos de cada exemplo que iremos alimentar à rede neural. Cada rótulo será representado por um vetor de tamanho igual ao número de classes possíveis, assim como o vetor de probabilidades. No caso dos rótulos, será atribuído o valor de **1.0** para a posição referente a classe correta daquele exemplo e **0.0** para todas as outras posições. Essa tarefa é simples e geralmente chamada de *One-Hot Encoding*. Com isso é possível medir a eficiência do treinamento apenas comparando dois vetores.

Camada convolucional

A camada de uma rede neural convolucional é uma rede que compartilha os seus parâmetros por toda camada. No caso de imagens, cada exemplo possui uma largura, uma altura e uma profundidade que é representada pelos canais de cor (vermelho, verde e azul). Uma convolução consiste em coletar um trecho da imagem de exemplo e aplicar uma pequena rede neural que teria uma quantidade qualquer de saídas (K). Isso é feito deslizando essa pequena rede neural pela imagem sem alterar os pesos e montando as saídas verticalmente em uma coluna de profundidade K. No final será montada uma nova imagem de largura, altura e profundidade diferente. Essa imagem é um conjunto de **mapas de características** da imagem original. Como exemplo, transforma-se 3 mapas de carcterísticas (canais de cores) para uma quantidade K de mapas de características.

Uma rede convolucional[Dumoulin et al. 2016] será basicamente uma rede neural de profundidade. Ao invés de empilhar camadas de multiplicação de matrizes, empilha-se convoluções. No começo haverá uma imagem grande que possui apenas os valores de pixel como informação. Em seguida são aplicadas convoluções que irão "espremer" as dimensões espaciais e aumentar a profundidade. No final é possivel conectar o classificador e ainda lidar apenas com parâmetros que mapeiam o conteúdo da imagem.

ReLU

Modelos lineares são simples e estáveis numericamente, mas podem se tornar ineficientes ao longo do tempo. Portanto, para adicionar mais camadas ao modelo será necessário introduzir alguns cálculos não lineares entre camadas. Em arquiteturas de profundidade, as funções de ativação dos neurônios se chamam *Rectified Linear Units* (ReLUs)[Bengio and Courville 2016], e são capazes de introduzir os cálculos necessários aos modelos que possuem mais de uma camada. Essas são as funções não lineares mais simples que existem. Elas são lineares ($y = x$) se **x** é maior que **0**, senão ficam iguais a **0** ($y = 0$). Isso simplifica o uso de *backpropagation* e evita problemas de saturação, fazendo o aprendizado ficar muito mais rápido.

Max pooling

Após a camada convolucional adiciona-se uma camada de *pooling* que irá receber todas as convoluções e combiná-las da seguinte forma[Dumoulin et al. 2016]. Para cada ponto nos mapas de características a execução desta camada olha para uma pequena vizinhança ao redor deste ponto. Com esses valores em mãos é possível calcular o valor máximo dessa vizinhança.

Dropout

Uma forma de regularização que previne o *overfitting*[1] é o *dropout*[Krizhevsky et al.]. Supondo que temos uma camada conectada à outra em uma rede neural, os valores que vão de uma camada para a próxima podem se chamar de **ativações**. No *dropout*, são coletadas todas as ativações e aleatoriamente, para cada exemplo treinado, é atribuído o valor 0 para metade desses valores. Basicamente metade dos dados que estão fluindo pela rede neural é destruída aleatoriamente.

Camada completamente conectada

De acordo com Krizhevsky[Krizhevsky et al.], uma camada completamente conectada tem conexões com todas as ativações das camadas anteriores, assim como em redes neurais comuns. Suas ativações podem ser calculadas através de uma multiplicação de matrizes seguida da adição do fator *bias*.

Devido a quantidade de componentes presentes na estrutura de redes neurais é aparente a complexidade quanto ao entendimento do funcionamento geral. A figura 1 tenta explicar como esses componentes se conectam e em qual sequência. Os valores são completamente fictícios e não condizem com um cálculo real.

[1] O *overfitting* ocorre quando um modelo de rede neural se encaixa muito bem em um conjunto de dados e acaba memorizando propriedades do conjunto de treinamento que não servem para o conjunto de teste.

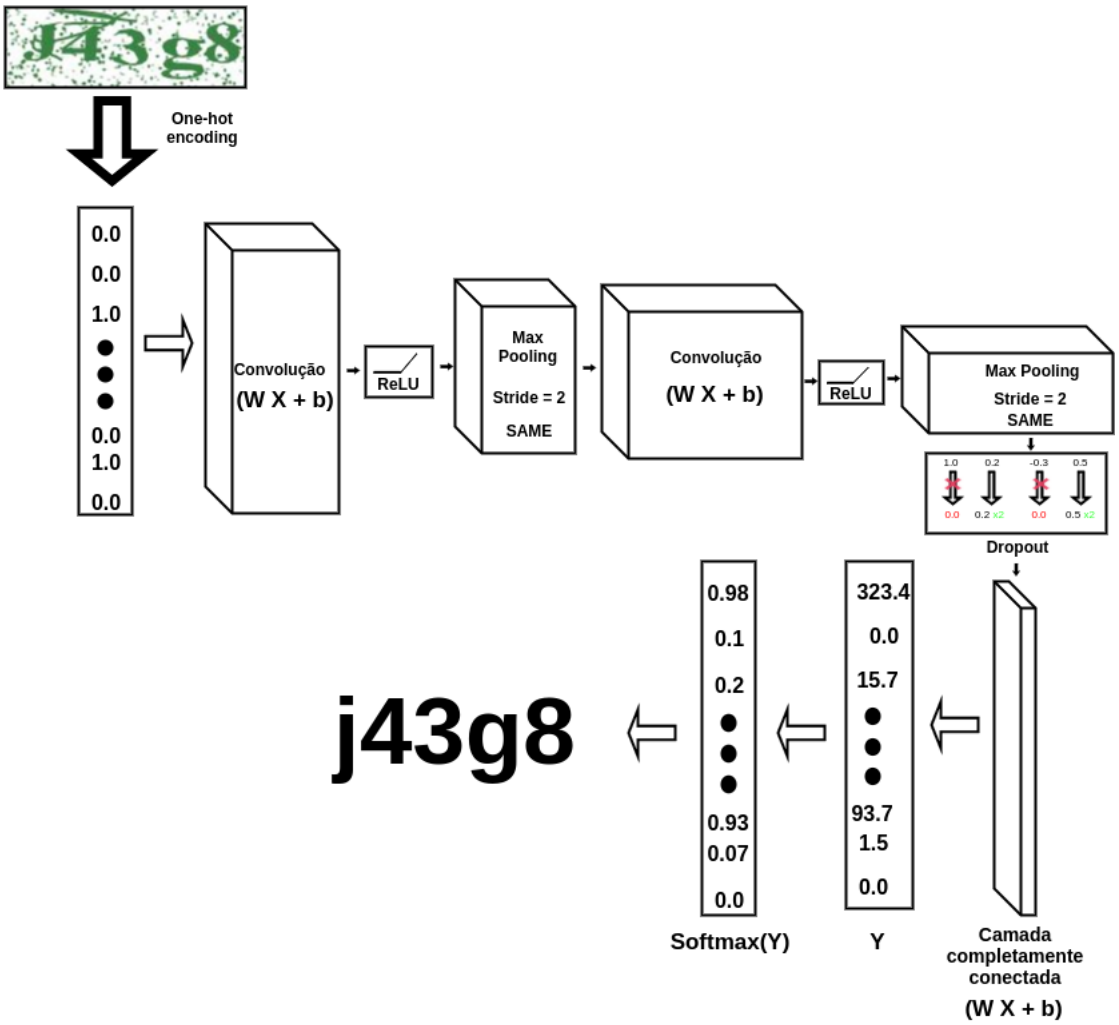


Figura 1. Exemplo da composição de todos os componentes presentes em redes neurais convolucionais de profundidade.

Experimento

Geração do Conjunto de dados

O conjunto de dados (ou *"dataset"*) que alimenta a rede neural é gerado em tempo de execução do treinamento. Cada imagem é lida de seu diretório em disco e carregada na memória como uma matriz de valores de pixel. Ao final deste processo há um vetor em memória com todas imagens existentes já pré-processadas. Isso é feito para o *dataset* de treinamento e de teste. Para o treinamento também será necessário um conjunto separado para teste que não possui nenhuma imagem presente no conjunto de treinamento. O *dataset* de treinamento terá cerca de 96% das imagens, e o *dataset* de testes terá 4% das imagens.

Junto com a geração do conjunto de dados, ocorre o pré-processamento das imagens. Durante o pré-processamento as imagens são transformadas para uma representação em escala de cinza. Por fim as imagens são redimensionadas para um tamanho menor, tornando os cálculos mais eficientes durante o treinamento da rede neural.

Treinamento

Após gerado o conjunto de dados, é possível trabalhar no treinamento do modelo da rede neural. Para isso será usado o *framework* ***TensorFlow***[TensorFlow] destinado à *Deep Learning*. Também será desenvolvido um *script* em *Python* que fará uso das funções disponibilizadas pela biblioteca do *TensorFlow*. Assim realizando o treinamento até atingir um valor aceitável de acerto no conjunto de teste. O resultado do treinamento será um arquivo binário representando o modelo que será utilizado para avaliação posteriormente.

Avaliação de acurácia

Com uma nova amostra de imagens, será feita a execução do teste do modelo que obteve melhor performance durante o treinamento. Ao final da execução será contabilizado o número de acertos e comparado com o número total da amostra de imagens para avaliação. Resultando assim em uma porcentagem que representa a acurácia do modelo gerado.

Desenvolvimento

Para a construção e treinamento da rede neural foi implementado um script em *Python* que possui toda a arquitetura da rede descrita de forma procedural. O *framework TensorFlow* chama a arquitetura dos modelos de *Graph* (ou grafo, em português) e o treinamento da rede neural é feito em uma *Session*.

O projeto é composto por 5 tarefas de implementação:

- Desenvolvimento do leitor e processador do conjunto de dados.
- Desenvolvimento da função que monta a rede neural.
- Configuração da rede neural para otimização dos resultados.
- Desenvolvimento da etapa de treinamento da rede neural.
- Desenvolvimento da etapa de teste e acurácia do modelo da rede neural.

Testes

Treinamento com 200 mil iterações

Inicialmente é realizado um treinamento com 200 mil iterações. A fase de treinamento completa levou **1 hora 23 minutos e 54 segundos** para completar. Deve-se salientar que para o primeiro treinamento há uma espera maior devido ao *caching* dos dados. Isso é feito pelo sistema operacional para otimizar a memória da GPU e do sistema em geral quando os dados são carregados para a memória volátil.

Como é possível observar nos gráficos o valor da acurácia fica em torno de 5% até um momento que começa a subir. Ao final do treinamento foi alcançado um valor máximo de acurácia igual a **84,38%** no conjunto de treinamento, **79,6%** no conjunto de teste.

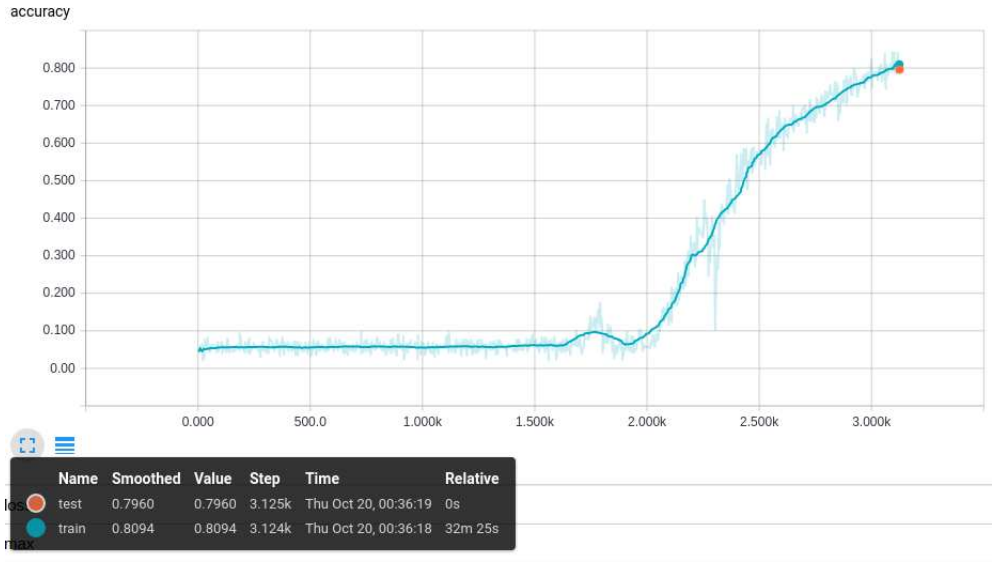


Figura 2. Gráfico da acurácia em relação ao número de passos para o treinamento da rede com 200 mil iterações.

Treinamento com 500 mil iterações

Visto a instabilidade nos valores de gráficos no treinamento anterior, a tentativa seguinte foi aumentar o número de iterações para 500 mil. O tempo total de treinamento foi de **1 hora 18 minutos e 23 segundos**.

Ao final do treinamento foi alcançado o valor máximo de acurácia igual a **98,75%** no conjunto de treinamento, **81,37%** no conjunto de teste.

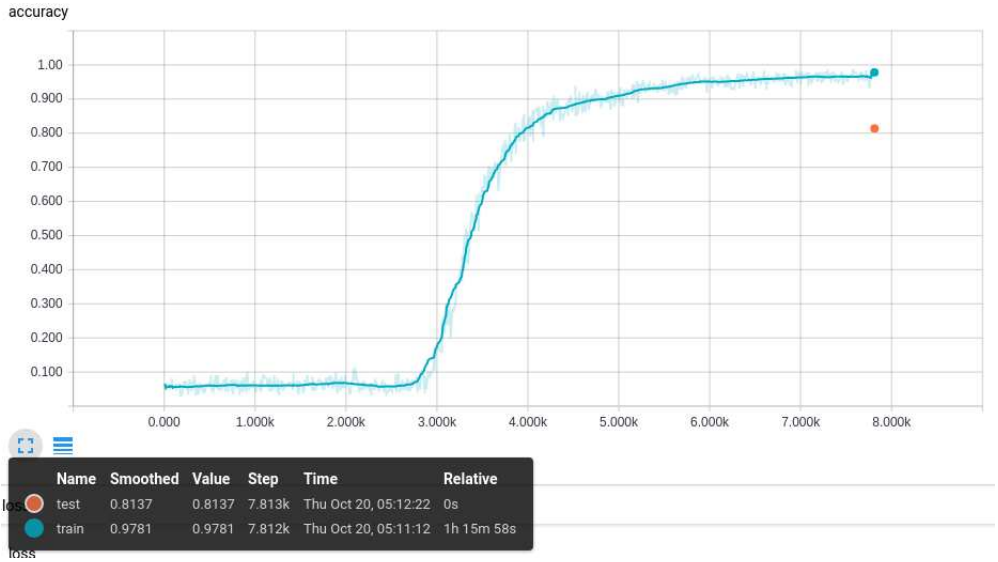


Figura 3. Gráfico da acurácia em relação ao número de passos para o treinamento da rede com 500 mil iterações.

Treinamento com 500 mil iterações e Dropout de 50%

Na tentativa de minimizar os problemas encontrados anteriormente, foi realizado um terceiro treinamento. Foi visto que uma das técnicas de regularização para minimizar o

overfitting é adicionando uma camada de *dropout* ao modelo. Nossa arquitetura já previa uma camada de *dropout*, no entanto o parâmetro de probabilidade de mantimento das ativações estava configurado para 75% (0,75). Para o terceiro treinamento foi configurada a probabilidade do *dropout* para 50% (0,5) e assim analisados os resultados. O tempo total de treinamento foi de **1 hora 18 minutos e 53 segundos**.

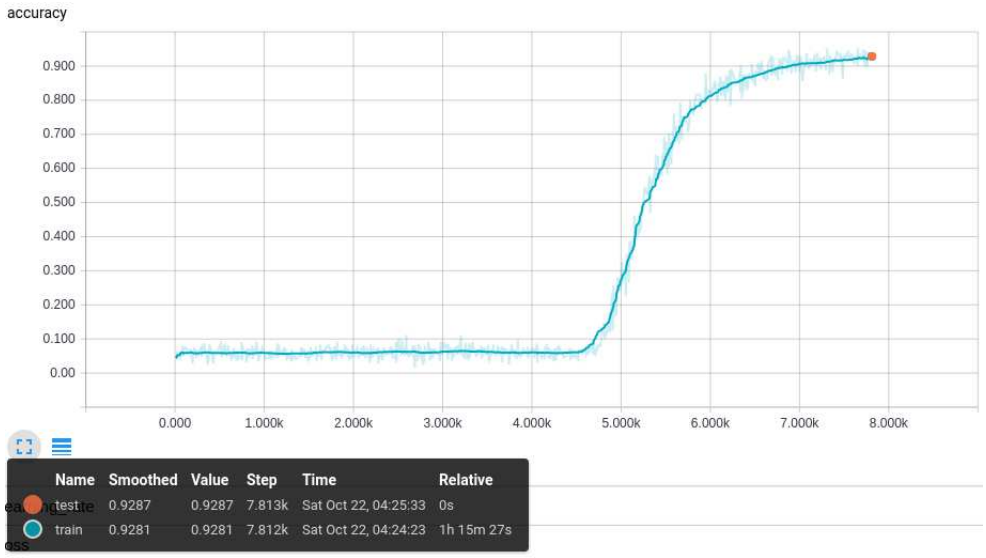


Figura 4. Gráfico da acurácia em relação ao número de passos para o treinamento da rede com 500 mil iterações e probabilidade de *dropout* igual a 50%.

Ao final do treinamento foi alcançado o valor máximo de acurácia igual a **95,31%** no conjunto de treinamento, **92,87%** no conjunto de teste.

Conclusão

Os testes realizados em todos os casos mostraram ser possível atingir um resultado razoável na tarefa de reconhecimento de textos em imagens, isso com poucos ajustes à configuração de treinamento de redes neurais. Atualmente a quantidade de exemplos e tutoriais disponíveis para tarefas de aprendizado de máquina é imenso. Fica claro que é possível implementar classificadores mesmo com poucos recursos.

Diante do objetivo alcançado pelo trabalho, fica aparente que fontes públicas de dados podem estar vulneráveis.

Trabalhos futuros

Como possíveis trabalhos futuros, cita-se:

- Fazer um melhor uso das informações geradas pelo processo de treinamento para gerar heurísticas mais inteligentes. Um exemplo seria utilizar outros tipos de otimizadores para a função da perda.
- Estender o sistema para realizar o reconhecimento de outros tipos de CAPTCHAs.
- Estender o sistema para realizar o reconhecimento de tipos de CAPTCHAs que possuem um tamanho de texto variável.
- Realizar um estudo sobre *Web crawlers* em fontes públicas que utilizam CAPTCHA, executando o sistema proposto neste trabalho.

- Implementar um sistema de reconhecimento de CAPTCHAs mais avançados que solicitam a classificação de uma cena completa ou identificação de objetos em imagens.
- Estudar um artifício mais efetivo para o bloqueio de consultas automatizadas em *websites*.

Considera-se de extrema importância a implementação de projetos desse tipo pois o mesmo auxilia na compreensão e aplicação de Inteligencia Artificial em casos específicos.

Referências

Bengio, I. G. Y. and Courville, A. (2016). Deep learning. Book in preparation for MIT Press.

Dumoulin, V., Visin, F., and Box, G. E. P. (2016). A guide to convolution arithmetic for deep learning.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. Acessado: 21/10/2016.

TensorFlow. TensorFlow — an Open Source Software Library for Machine Intelligence. Acessado: 03/08/2016.