Detecção e Extração de Templates em Páginas Web

Karane Vieira[1], **Altigran Soares da Silva (Orientador)**[1]

[1]Departamento de Ciência da Computação – Universidade Federal do Amazonas (UFAM)
Manaus-AM, Brasil (Dept. onde dissertação foi aprovada)
{karane,alti}@dcc.ufam.edu.br

Abstract. *The widespread use of templates on the Web is considered harmful for two main reasons. Not only do they compromise the relevance judgment of many web IR and web mining methods, but they also negatively impact the performance and resource usage of tools that process web pages. In this paper we present two new algorithms based on tree mappings that efficiently and accurately removes templates found in collections of web pages by just inspecting a few sample pages. We show that our algorithms are effective for identifying terms occurring in templates - obtaining F-measure values around 0.9, and that they also boost the accuracy of web page clustering and classification methods.*

Resumo. *O difundido uso de templates na Web é considerado prejudicial por duas razões principais. Não só eles comprometem o julgamento de relevância de muitos métodos de RI e mineração para a Web, mas também influenciam negativamente o uso de recursos por ferramentas que processam páginas web. Neste artigo, apresentamos dois novos algoritmos baseados em mapeamentos de árvores que de forma eficiente e acurada removem templates encontrados em coleções de páginas web inspecionando apenas poucas páginas exemplo. Mostramos que nossos algoritmos são efetivos em identificar termos que ocorrem em templates - obtendo valores de medida F por volta de 0,9 - e que eles podem melhorar a acurácia de métodos de agrupamento e classificação de páginas web.*

1. Introdução

A World Wide Web é hoje um gigantesco repositório de dados distribuídos publicamente, cujo número de páginas tem crescido com grande velocidade nos últimos anos. Esse ritmo de crescimento acelerado deve-se em parte ao uso de ferramentas de projeto de web sites que aumentam a produtividade dos provedores de conteúdo. Um dos principais recursos usados por essas ferramentas automáticas são os *templates*, porções de código HTML embutidas nas páginas web que predefinem a apresentação e a estrutura de um conjunto de páginas (ex.: menus, barras de navegação, logotipo do site, etc). Assim, uma vez definido o template, as ferramentas permitem aos projetistas de páginas web a inserção de conteúdo, seja manualmente ou automaticamente por meio de acesso a um banco de dados. Além disso, templates geralmente permitem uma melhor navegação pelas páginas de um web site, pois promovem a uniformidade de recursos visuais e de navegação.

Num estudo recente, Gibson et. al. [Gibson et al. 2005] mostraram que de 30 a 40% do volume de dados nas páginas da Web estão contidos nos templates, e que este volume vem crescendo num ritmo de 6 a 8% ao ano. Trabalhos na literatura mostram que templates podem afetar negativamente as tarefas de busca e mineração de dados tais como

classificação e agrupamento, pois uma vez que estes métodos se baseiam na freqüência e distribuição dos termos na coleção de documentos, termos que aparecem em templates podem influenciar os resultados. Templates podem também prejudicar o uso de recursos computacionais por sistemas que manipulam páginas web. Como o uso de templates implica em conteúdo replicado em múltiplas páginas de um site, processar templates pode desperdiçar recursos como espaço de armazenamento e ciclos de processamento. Dado o crescimento da Web, isso deve ser levado em consideração.

Neste contexto, métodos eficientes para automaticamente detectar templates em coleções de páginas web tornam-se necessários. Nesta dissertação de mestrado propomos dois novos algoritmos para detecção de templates em coleções de páginas web, chamados RTDM-TD e RBM-TD, e discutimos suas aplicações em classificação e agrupamento de páginas. Nossos algoritmos para detectar templates são baseados na idéia de encontrar um mapeamento entre as estruturas de árvores subjacentes às páginas web. Ambos os algoritmos foram publicados em artigos completos derivados desta dissertação. O algoritmo RTDM-TD foi publicado nos anais da conferência ACM CIKM 2006 [Vieira et al. 2006] e o algoritmo RBM-TD foi publicado no periódico World Wide Web Journal em 2009 [Vieira et al. 2009]

Para avaliar nossa abordagem, realizamos dois tipos de experimentos. No primeiro, verificamos quanto os algoritmos são eficazes para detectar templates. Os resultados mostram valores de medida F por volta de $0,9$ usando poucas dezenas de páginas para ambos os algoritmos. No segundo tipo, avaliamos o efeito que eles produzem quando aplicados aos problemas de agrupamento e classificação páginas web. Os resultados dos experimentos de classificação e agrupamento mostram uma melhora significativa na qualidade desses métodos e são superiores aos obtidos em [Yi et al. 2003].

Este artigo está organizado da seguinte forma. Na Seção 2 apresentamos os principais trabalhos relacionados. Em seguida, na Seção 3 apresentamos a descrição de nossos algoritmos para a detecção de templates em páginas web. A Seção 4 descreve os experimentos para atestar a qualidade dos algoritmos, seguido de experimentos sobre os efeitos que os nossos algoritmos produzem ao serem aplicados em classificação e agrupamento de páginas web. Finalmente, na Seção 5 apresentamos as nossas conclusões e trabalhos futuros. Por limitação de espaço, apenas nossos resultados principais foram transcritos neste documento. A descrição de nossos algoritmos assim como experimentos mais extensos são apresentados na dissertação original em `http://www.dcc.ufam.edu.br/~karane/dissertacao-mestrado-karane.pdf`.

2. Trabalhos Relacionados

Motivados pelo potencial impacto que templates podem ter nos algoritmos de busca e mineração na Web, trabalhos recentes neste tópico têm sido publicados na literatura. Bar-Yossef et al. [Bar-Yossef and Rajagopalan 2002] foram os primeiros a tratarem o problema de detecção de templates. Eles propuseram dois algoritmos, ambos baseados na identificação de *pagelets* comuns dentre as páginas presentes numa coleção de páginas com muitos apontadores entre si. Experimentos baseados na máquina de busca Clever usando o conjunto de consultas ARC mostraram melhora considerável para um dos algoritmos, o *Local Template Detection Algorithm*.

Uma abordagem alternativa é proposta por Lan Yi. et al. [Yi et al. 2003] onde o objetivo é achar “ruídos”, ou seja, conteúdo considerado irrelevante, em páginas do

web site. Nessa abordagem, uma estrutura chamada *árvore de estilos* (SST - Site Style Tree) é construída para representar um sumário de todos os estilos de apresentação e de todos os conteúdos achados em um site. Os autores construíram um *estimador* baseado na diversidade de estilos e em um limiar dado pelo usuário para decidir se um dado nodo é ruidoso. Menor diversidade indica maior chance de um nó ser ruidoso. O processo de remoção de ruídos é obtido mapeando a árvore DOM subjacente de cada página do site à SST. Se um dado nó DOM de uma página é mapeado para a SST é determinado um certo grau de ruído, esse nó é removido. Esta abordagem apresenta uma desvantagem: é preciso que um grande número de páginas sejam avaliadas para alcançar dados estatísticos confiáveis. Nos experimentos relatados pelos autores, 500 páginas são usadas em cada site para detectar os elementos que representam ruído. Como mostraremos em nossos experimentos (Seção 4), nosso método apresenta resultados comparáveis aos produzidos pela SST com um número bem menor de páginas exemplo.

3. Algoritmos de Detecção de Templates

Os algoritmos propostos nesta dissertação para detectar templates se baseiam na idéia de encontrar um mapeamento entre árvores DOM subjacentes às páginas web. De posse desse mapeamento, identificamos os nós idênticos nas árvores e subárvores que contêm esses nós. Intuitivamente, quando uma subárvore é encontrada, ela pode ser facilmente identificada nas outras páginas da coleção. O uso da estrutura de página para a detecção de template alcança alta precisão mesmo quando uma pequena quantidade de páginas exemplo é dada.

A Figura 1 mostra como nossa abordagem de mapeamento de árvores pode ser usada para detectar templates em um conjunto de páginas que compartilham o mesmo template. Na Figura 1(a), dado um conjunto de páginas, representamos o seu código HTML de cada página do conjunto como uma árvore DOM. A cada par de páginas do conjunto, fazemos o mapeamento entre seus nodos de mesmo rótulo. para encontrar o template. A Figura 1(b) mostra como podemos iteragir entre as páginas de uma coleção de páginas para detectar o template. Observe que o mapeamento é realizado no primeiro par de árvores DOM que foram derivadas das duas primeiras páginas do conjunto. O template então encontrado é usado para ser mapeado com a terceira página. Este passo é repetido até um determinado número de páginas da coleção sejam processadas, quando finalmente obtemos um template final.

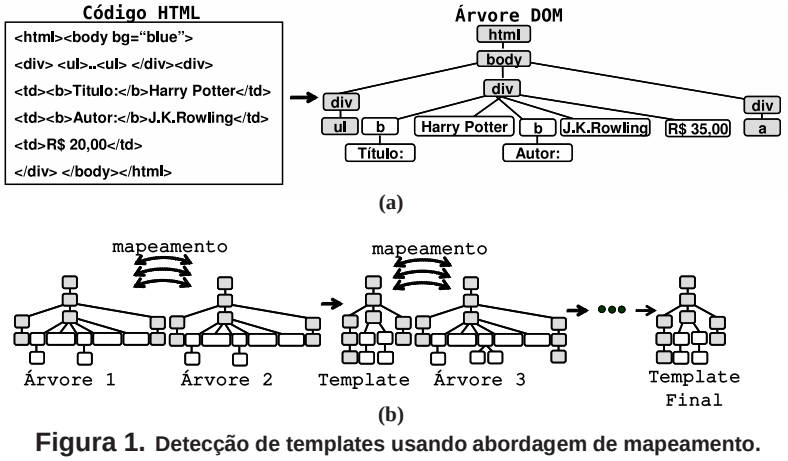

Figura 1. **Detecção de templates usando abordagem de mapeamento.**

3.1. RTDM-TD

O primeiro algoritmo é baseado numa formulação restritiva do problema de *mapeamento top-down* entre duas árvores, que é particularmente adequado para detectar similaridades estruturais entre páginas web. A Figura 2 mostra um exemplo de mapeamento top-down restrito. Este é um mapeamento top-down, onde se pode apenas mapear nodos cujos ancestrais já foram mapeados. Estes mapeamentos são representados pelas linhas tracejadas, onde nodos de mesmo rótulo tem maior prioridade para serem escolhidos no mapeamento. O mapeamento top-down restrito, por sua vez, adiciona a regra de não mapear subárvores cujas raízes sejam diferentes. Observe na Figura 2 que as subárvores enraizadas em $T_1[8]$ e $T_2[8]$[1] não foram mapeadas, apesar de possuírem nodos de rótulo P. No caso do mapeamento top-down, estas subárvores seriam aceitas por inteiro no mapeamento.

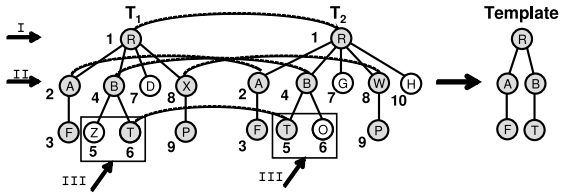

Figura 2. Mapeamento Top-Down Restrito e template correspondente.

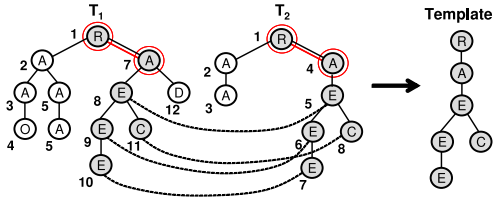

Figura 3. Mapeamento Bottom-Up Restrito e template correspondente.

O algoritmo, que nós chamamos de RTDM-TD (Restricted Top-Down Mapping - Template Detection), é uma adaptação do proposto por Reis et al. [de Castro Reis et al. 2004], que possui complexidade $O(|T_1||T_2|)$, onde $|T_x|$ denota o número de nodos da árvore T_x. Como apresentaremos na dissertação na original, apesar desse algoritmo ser relativamente custoso, a natureza do tipo de mapeamento de árvores que ele procura, faz o nosso método eficaz e eficiente na prática.

O algoritmo funciona da seguinte forma. Primeiro, tenta-se alinhar as raízes de T_1 e T_2 indicados pela seta I (Figura 2). Em seguida, o RTDM-TD faz uma chamada recursiva para alinhar os nodos em ambas as árvores indicados pela seta II. Quando tenta alinhar os nodos $T_1[4]$ e $T_2[4]$, ambos de rótulo B, o RTDM-TD faz outra chamada recursiva para alinhas os nodos filhos dos nodos correntes. Este mapeia os nodos $T_1[6]$ e $T_2[5]$. E assim o algoritmo prossegue recursivamente em todos os pares de subárvores que procura mapear. No final, um template é formado a partir do conjunto de nodos, em cinza, no mapeamento.

3.2. RBM-TD

O segundo algoritmo é derivado do trabalho de Valiente [Valiente 2001], onde o autor propõe um algoritmo para encontrar o *mapeamento bottom-up* entre duas árvores e cuja complexidade é $O(|T_1| + |T_2|)$. A este novo algoritmo damos o nome de RBM-TD (Restricted Bottom-up Mapping - Template Detection). O RBM-TD encontra mapeamentos bottom-up mais restritos e possui a mesma complexidade de tempo que o algoritmo proposto por Valiente. A diferença é que o RBM-TD apenas considera as subárvores localizadas no mesmo caminho desde as raízes das árvores válidas no mapeamento. A Figura 3 mostra um exemplo de mapeamento bottom-up restrito entre duas árvores. Este é um mapeamento bottom-up, onde nodos só podem ser mapeados se e somente se os nodos filhos também forem mapeados. O mapeamento bottom-up restrito, por sua vez, adiciona a regra de mapear subárvores que estejam co-localizadas no mesmo caminho deste as raízes

[1] $T_x[y]$ indica o nodo de índice y na árvore T_x.

das árvores. Por exemplo, as subárvores $T_1[5]$ e $T_2[2]$ estão co-localizadas no caminho destacado com linhas e círculos duplos.

O RBM-TD funciona da seguinte forma. Primeiro o algoritmo detecta todos os conjuntos de subárvores idênticas. Cada conjunto distinto chamamos de classe de equivalência. Em seguida, ele caminha nas árvores de forma top-down e mapeia as subárvores idênticas que possuem o mesmo caminho desde as raízes. No final, o template é formado a partir do conjunto de nodos, em cinza, no mapeamento.

4. Experimentos

Nesta seção, apresentamos os experimentos realizados para validar os algoritmos de detecção de templates que propomos nesta dissertação. Primeiro, apresentamos uma avaliação de qualidade dos templates detectados. Em seguida, mostramos como nossos algoritmos influenciam tarefas de classificação e agrupamento de páginas web.

4.1. Avaliação Direta

Neste experimento, avaliamos a qualidade dos templates produzidos pelos nossos algoritmos RTDM-TD e RBM-TD. Para este experimento coletamos na web uma amostra de páginas dos sites de comércio eletrônico PCConnection(PC), CNet, J&R, PCMagazine e ZDNet. Sobre esta coleção[2] de páginas aplicamos o CyberNeko[3] para remover erros de código HTML mal formado. Para cada site i, extraímos manualmente o template e depois definimos um conjunto de referência S_i contendo as palavras que ocorrem no template de cada site i. A extração manual do template foi realizada através de edição das páginas HTML utilizando uma ferramenta de edição visual de páginas web. A decisão que quais porções visuais compõem o template para o site foi realizada manualmente inspecionando 100 páginas aleatórias de cada site. Uma vez definido os conjuntos S_i, aplicamos os nossos algoritmos, o RTDM-TD e o RBM-TD, para automaticamente detectar os templates destes sites e geramos o conjunto W_i de palavras presentes nos templates detectados. Em todos os conjuntos S_i e W_i palavras repetidas são consideradas, dado a possibilidade que palavras podem aparecer mais de uma vez nos templates. Usamos a medida F[4] para avaliar o nossos algoritmos quanto à qualidade do template detectado.

As Figuras 4(a) e 4(b) mostram um comportamento semelhante para ambos os algoritmos. Para todos os sites exceto CNet, valores de medida F acima de $0,9$ foram alcançados com um número pequeno de páginas, e esses valores não mudaram quando mais páginas exemplo são usadas. Como pode ser observado nos gráficos, 30 é um bom número de páginas exemplo para garantir alta qualidade de templates extraídos para ambos os algoritmos. Os valores inferiores de medida F obtidos para o site CNet porque alguns erros de HTML mal formado ainda persistiram, apesar de termos usado o CyberNeko. Isso causou a degeneração de algumas subárvores DOM das páginas do site CNet. Assim, nossos algoritmos identificaram templates menores porque algumas dessas subárvores não foram identificadas como constituintes do template para o site, o que produziu menores valores de medida F. Apesar disso, ainda conseguimos obter bons resultados nos nossos experimentos de classificação e agrupamento, apresentados nas Seção 4.2.

[2]Disponível em `http://www.dcc.ufam.edu.br/~karane/sites.trunc.tar.gz`
[3]http://sourceforge.net/projects/nekohtml
[4]Descrita em detalhes na dissertação original.

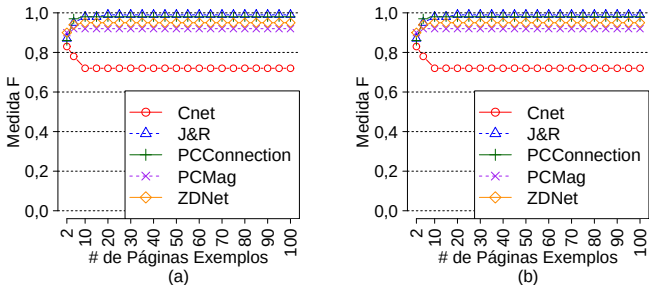

Figura 4. **Qualidade da detecção de templates com o RTDM-TD(a) e RBM-TD(b)**

4.2. Aplicação no Agrupamento de Páginas

Neste experimento, usamos os mesmos cinco sites mencionados na Seção 4.1 todos com páginas de produtos das seguintes categorias: Notebook, Camera Digital, Celular, Impressora (Imp) e TV. A Tabela 1 mostra a distribuição das páginas de cada site em cada categoria.

Tabela 1. **Número de páginas em cada categoria por site.**

Sites	PC	CNet	J&R	PCMag	ZDNet
Notebook	600	497	74	150	218
Camera	169	227	216	143	132
Celular	8	120	47	52	118
Imp	496	511	133	117	97
TV	278	451	161	103	141

Tabela 2. **Ganhos Relativos no Experimento de Agrupamento.**

Bases	Média de F	Ganho
Original	0,28	————
SST	0,40	41%
RTDM-TD	0,45	58%
RBM-TD	0,46	61%

Para avaliar esta aplicação, usamos nossa implementação do algoritmo K-Means para construir agrupamentos de páginas das mesmas categorias. Os experimentos foram realizados de maneira muito similar em [Yi et al. 2003], como descrito a seguir. Rodamos o K-Means 800 vezes com sementes selecionadas randomicamente para que obtivéssemos cinco grupos. Para avaliar o agrupamento, usamos a medida F. O cálculo da medida F foi obtido tendo as categorias de páginas como referência. O K-Means gera cinco grupos correspondentes às cinco classes da nossa coleção de páginas. Uma vez que estamos realizando uma tarefa de agrupamento, não sabemos qual grupo corresponde a cada classe. Para avaliarmos a qualidade do agrupamento atribuímos todos os arranjos de possíveis classes a estes grupos e calculamos a medida F para cada arranjo. Decidimos por aquele arranjo de classes aos grupos que gera a melhor média de medida F entre os grupos.

A Figura 5 mostra o número de execuções que alcançaram os valores de medida F nas faixas $[0.0, 0.1), [0.1, 0.2), ..., [0.9, 1.0]$. Cada barra corresponde à média de valores de F das cinco categorias, antes e depois de removermos os templates. Os algoritmos tiveram ganhos significativos no experimento em relação aos resultados do agrupamento com as páginas originais. No gráfico, podemos observar que com os algoritmos RTDM-TD e RBM-TD, foi possível gerar mais casos onde o média de medida F alcança valores nas faixas $0,5$, $0,6$ e $0,7$, enquanto a SST conseguiu no máximo alcançar a faixa $0,6$ em poucas iterações.

A Tabela 2 mostra os ganhos relativos do K-Means quando aplicado a cada base gerada pelos métodos. Esses ganhos foram calculados comparando a média da medida F na base original e as bases com templates removidos de acordo com cada algoritmo SST, RTDM-TD e RBM-TD. Os algoritmos de mapeamento de árvores tiveram ganhos relativos superiores ao da SST, o que confirma que nossos algoritmos baseados em mapeamento de árvores são superiores para aplicação na tarefa de agrupamento de páginas.

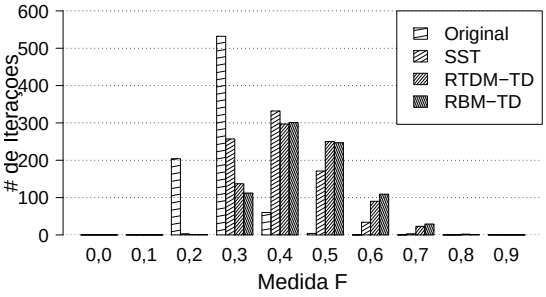

Figura 5. **Resultados do Agrupamento.**

Tabela 3. **Configuração dos Experimentos de Classificação**

Conf.	Conjunto de Treinamento (TR)	Conjunto de Teste (TE)
1	Páginas das categorias p e q do site i	Páginas das categorias p e q de todos os sites exceto i
2	Páginas das categoria p do site i e páginas da categoria q do site $j \neq i$	Páginas das categorias p e q de todos os sites exceto i e j
3	Páginas da categoria p do site i e páginas da categoria q do site $j \neq i$	Páginas da categoria p e q e todos os sites exceto as páginas no Conjunto de Teste

Também é possível observar que o RBM-TD obteve ganho relativo de 61%, melhor que o RTDM-TD, que obteve 58%. A superioridade dos nossos algoritmos é reforçada pelo fato de que eles demandam menor quantidade de amostras para inferir um template correto que o algoritmo SST.

4.3. Aplicação na Classificação de Páginas

Aqui apresentamos a avaliação da aplicação de nossos algoritmos na classificação de páginas web. Para esta avaliação, utilizamos os mesmos sites e categorias da seção anterior. Esta avaliação também seguiu os procedimentos de nosso baseline [Yi et al. 2003], descritos a seguir.

Usando um classificador Naïve Bayes[5] realizamos um experimento com três configurações diferentes usando todos os pares possíveis de categorias presentes na coleção de páginas web que coletamos. As configurações são resumidas na Tabela 3. Para cada uma delas, geramos um classificador a partir do Conjunto de Treino (TR) e então o rodamos sobre o Conjunto de Teste (TE) correspondente. Cada configuração impõe ao classificador uma situação diferente onde os templates podem confundi-lo em maior ou menor grau.

As Tabelas 4 e 5 apresentam apenas as médias de medida F (F) e Acurácia (A), respectivamente, para os algoritmos RTDM-TD e RBM-TD e o método baseado na SST. Nas Tabelas 4 e 5, o rótulo *O* denota páginas originais, o rótulo S, páginas sem template, e G denota o ganho relativo para cada par de categorias em cada configuração. Observe que em ambas as tabelas e em todas as configurações, nossos algoritmos obtêm ganhos relativos superiores aos do método baseado na SST. Estes valores nos fazem concluir que os nossos algoritmos de detecção de templates são uma alternativa melhor para detectar templates quando o objetivo é melhorar a qualidade de respostas das tarefas de classificação de páginas web.

Tabela 4. **Médias dos valores de medida F nas tarefas de classificação.**

	Conf. 1			Conf. 2			Conf. 3		
	F1(O)	F1(S)	G	F2(O)	F2(S)	G	F3(O)	F3(S)	G
RTDM-TD	0,881	0,938	0,064	0,570	0,761	0,335	0,416	0,603	0,450
RBM-TD	0,881	0,939	0,065	0,570	0,759	0,332	0,416	0,607	0,459
SST	0,881	0,861	-0,022	0,570	0,671	0,177	0,416	0,521	0,252

5. Conclusão

Nesta dissertação de mestrado, apresentamos dois novos algoritmos, o RTDM-TD e o RBM-TD para o problema de detecção de templates em coleções de páginas web. Em

[5]Implementação disponível no pacote "Bow Toolkit" (`http://www.cs.cmu.edu/~mccallum/bow/`)

Tabela 5. **Médias dos valores de Acurácia nas tarefas de classificação.**

	Conf. 1			Conf. 2			Conf. 3		
	A1(O)	A1(S)	G	A2(O)	A2(S)	G	A3(O)	A3(S)	G
RTDM-TD	0,892	0,945	0,060	0,607	0,781	0,288	0,447	0,631	0,411
RBM-TD	0,892	0,945	0,060	0,607	0,781	0,288	0,447	0,631	0,411
SST	0,892	0,916	0,027	0,607	0,770	0,270	0,447	0,600	0,342

ambos os algoritmos, templates são detectados construindo mapeamentos entre árvores DOM de páginas distintas e encontrando subárvores que são comuns a essas páginas. O primeiro algoritmo RTDM-TD detecta mapeamentos top-down restritos entre árvores em tempo proporcional a $O(n^2)$. Procurando encontrar uma alternativa mais rápida para a detecção de templates, criamos o RBM-TD, que encontra mapeamentos bottom-up restritos e cuja complexidade de tempo é proporcional a $O(n)$. Mostramos que não só somos capazes de fazer estes mapeamentos de forma eficiente com ambos os algoritmos, mas também que alta precisão pode ser obtida com um número pequeno de páginas exemplo. Nossos experimentos mostram que nossa abordagem melhora a qualidade dos resultados em aplicações de classificação e agrupamento de páginas. Tanto o RBM-TD quando o RTDM-TD foram capazes de obter ganhos maiores do que os produzidos pelo método baseado na SST, nosso baseline.

Como trabalho futuro, pretendemos investigar o problema de como os nossos algoritmos podem interagir com as evidências usadas comumente em máquinas de busca web, além do modelo vetorial, para melhorar a qualidade das respostas e economizar recursos como espaço de armazenamento e ciclos de processamento. Os resultados apresentados nesta dissertação são promissores e apontam para uma solução deste problema.

Referências

Bar-Yossef, Z. and Rajagopalan, S. (2002). Template detection via data mining and its applications. In *Proc. of the Int. Conf. on the World Wide Web*, pages 580–591.

de Castro Reis, D., Golgher, P. B., da Silva, A. S., and Laender, A. H. F. (2004). Automatic web news extraction using tree edit distance. In *Proc. of the Int. Conf. on the World Wide Web*, pages 502–511.

Gibson, D., Punera, K., and Tomkins, A. (2005). The volume and evolution of web page templates. In *Proc. of the Int. Conf. on the World Wide Web - Poster Session*, pages 830–839.

Valiente, G. (2001). An efficient bottom-up distance between trees. In *Proc. of the Int. Symposium on String Processing and Information Retrieval*.

Vieira, K., Costa Carvalho, A. L., Berlt, K., Moura, E. S., Silva, A. S., and Freire, J. (2009). On finding templates on web collections. *World Wide Web*, 12(2):171–211.

Vieira, K., da Silva, A. S., Pinto, N., de Moura, E. S., Cavalcanti, J. M. B., and Freire, J. (2006). A fast and robust method for web page template detection and removal. In *Proc. of the ACM Int. Conf. on Information and Knowledge Management*, pages 258–267.

Yi, L., Liu, B., and Li, X. (2003). Eliminating noisy information in web pages for data mining. In *Proc. of the Int. ACM Conf. on Knowledge Discovery and Data Mining*, pages 296–305.