Descoberta de ruído em páginas da *web* oculta através de uma abordagem de aprendizagem supervisionada

João A. F. Lutz, Carlos A. Heuser

Universidade Federal do Rio Grande do Sul, Brazil
{jaflutz, heuser}@inf.ufrgs.br

Abstract.

Um dos problemas da extração de dados na *web* é a remoção de ruídos existentes nas páginas. Esta tarefa busca identificar todos os elementos não informativos em meio ao conteúdo, como por exemplo cabeçalhos, menus ou propagandas. A presença de ruídos pode prejudicar seriamente o desempenho de motores de busca e tarefas de mineração de dados na *web*. Este trabalho aborda o problema da descoberta de ruídos em páginas da *web* oculta, a parte da *web* que é acessível apenas através do preenchimento de formulários. No processamento da *web* oculta, a extração de dados geralmente é precedida por uma etapa de inserção de dados, na qual os formulários que dão acesso às páginas ocultas são automaticamente ou semi-automaticamente preenchidos. Durante esta fase, são coletados dados do domínio em questão, como os rótulos e valores dos campos. A proposta deste trabalho é agregar este tipo de dados com informações sintáticas dos elementos que compõem a página. É mostrado empiricamente que esta combinação atinge resultados melhores que uma abordagem baseada apenas em informações sintáticas.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Information filtering

Keywords: *Web* oculta, Recuperação de Informações, Eliminação de Ruídos *Web*

1. INTRODUÇÃO

O crescimento exponencial da *web* gerou uma divisão em diferentes segmentos dentro desta. Denomina-se "*web* visível" a parte da *web* cujo conteúdo é possível acessar diretamente, seja através de motores de busca ou *URLs (Uniform Resource Locators)* estáticas. Porém existe outra porção da *web* chamada "*web* oculta", que é composta de páginas geralmente acessíveis apenas através do preenchimento de formulários, e são renderizadas dinamicamente com informações provenientes de um banco de dados. [Bergman 2001] estima que a *web* oculta é 550 vezes maior do que a *web* visível. A exploração dessa parte oculta é dividida em três fases distintas [Raghavan and Garcia-Molina 2000]: primeiro, é necessário descobrir os pontos de entrada para este conteúdo, ou seja, encontrar formulários que, quando preenchidos corretamente, direcionam o usuário para páginas com conteúdo que normalmente não é acessível via *URLs* ou motores de busca. A segunda fase envolve o uso de técnicas para preencher tais formulários, visto que diferentes preenchimentos levam a diferentes resultados. Após a submissão do formulário, a última fase compreende a extração dos dados da página resultante.

Este trabalho está inserido no contexto da terceira fase, onde os resultados de uma busca devem ser extraídos. Busca-se neste trabalho eliminar o ruído nas páginas *web* resultantes. Ruído em páginas *web* é todo conteúdo que não é informativo nem de interesse do usuário final [Yi et al. 2003]. Entre estes elementos não informativos, podem ser citados banners, propagandas, painéis de navegação, informações de contato, e até mesmo políticas de privacidade. [Gibson et al. 2005] estima que até 50% de todo conteúdo da *web* é composto de algum tipo de ruído. A importância da remoção de ruídos

está no fato de que estas porções de conteúdo prejudicam seriamente a performance de sistemas de recuperação de informações na *web*.

Diversos métodos foram desenvolvidos para alcançar este objetivo, e é possível classificá-los em grupos relativos à maneira como o ruído é identificado e removido. [Cai et al. 2003], [Fernandes et al. 2007], [Li and Ezeife 2006], [Song et al. 2004], [Kovacevic et al. 2002] e [Burget and Rudolfova 2009] utilizam algoritmos baseados em visão, já que todos estes trabalhos classificam o conteúdo da página *web* em ruído ou não baseado nos seus atributos visuais. O segundo grupo, composto pelos trabalhos de [Bar-Yossef and Rajagopalan 2002], [Debnath et al. 2005], [Lin and Ho 2002], [Chen et al. 2006] e [Wang et al. 2008], tenta identificar o ruído procurando por *templates* nas páginas *web*. Este procedimento é feito procurando-se por repetição de conteúdo nestas páginas através de heurísticas bem definidas. O outro grupo, com os trabalhos de [Yi et al. 2003] e [Vieira et al. 2006], tenta identificar o ruído baseado na similaridade estrutural da árvore *DOM (Document Object Model)* das páginas *web*. Por fim, existe um grupo que busca identificar o ruído baseado em propriedades textuais, como [Kohlschütter et al. 2010], [Weninger et al. 2010] e [Sun et al. 2011]. Este último será usado neste trabalho como *baseline*. O grande problema da maior parte destes trabalhos é que todos precisam assumir que as páginas *web* possuem uma estrutura específica, ou precisam procurar por elementos *HTML* especiais.

O objetivo principal deste trabalho, dada sua inserção no contexto da *web* oculta, é melhorar os resultados da remoção de ruídos baseando-se em uma técnica que utiliza a densidade de texto e *links* em uma página *web* para encontrar o ruído [Sun et al. 2011]. O trabalho referido assume que elementos que representam conteúdo informativo possuem uma densidade de texto puro mais alta que o ruído. Elementos ruidosos, por sua vez, possuem um número alto de elementos de formatação, além de muito mais *links* em seu conteúdo. Após a implementação do trabalho como *baseline*, foi possível observar que pode-se obter resultados melhores utilizando dados de fases anteriores da exploração da *web* oculta. Este objetivo é atingido utilizando-se um método de aprendizagem supervisionado para classificar elementos de texto em ruído ou não.

O trabalho está organizado da seguinte maneira: no capítulo 2 é mostrada uma técnica para remoção de ruído utilizada como *baseline*; no capítulo 3, é explicado como é possível melhorar a remoção de ruídos e obter melhores resultados utilizando um método de aprendizagem supervisionado; no capítulo 4, são descritos e comparados os experimentos realizados com ambas as técnicas, utilizando-se diferentes conjuntos de dados; no capítulo 5, são discutidos os resultados encontrados e possibilidades de trabalhos futuros.

2. ELIMINAÇÃO DE RUÍDOS ATRAVÉS DE PROPRIEDADES TEXTUAIS

Diversos trabalhos focaram-se na criação de técnicas automáticas de remoção de ruídos e, através destes, diferentes metodologias foram desenvolvidas. Esta diversidade impede a definição de um padrão claro para este tipo de extração de dados. Um grupo de métodos, composto entre outros por [Kohlschütter et al. 2010] e [Weninger et al. 2010], busca identificar o ruído baseado em propriedades textuais contidas nas páginas *web*. Neste trabalho, foi implementada e utilizada como *baseline* a técnica descrita em [Sun et al. 2011]. Ela assume que o conteúdo principal da página possui mais elementos de texto do que elementos de formatação. Já um trecho ruidoso, por sua vez, seria composto de um número maior de elementos de formatação, além de possuir textos mais curtos e mais *links*. São definidas então duas medidas, chamadas de Densidade de Texto (DT_i) e Densidade de Texto Composta (DTC_i). Estas medidas funcionam da seguinte maneira: após realizar o parsing dos elementos do código *HTML* e representar a página *web* como uma árvore, todos os comentários, scripts e elementos de estilo são removidos. Então, para cada nodo da árvore resultante, são contados o número de caracteres abaixo deste nodo, juntamente com o número de *tags*. A proporção entre estes dois parâmetros é chamada de Densidade de Texto, ou seja, quanto maior o número de caracteres existentes em uma ramificação da árvore *DOM*, maior é a probabilidade desta sub-árvore representar

conteúdo relevante. Se o número de *tags* for muito alto, a Densidade de Texto possuirá um valor baixo, o que significa que a taxa de formatação é alta e a probabilidade desta ramificação ser ruído também será alta.

Apesar de ser um bom indicativo, esta medida pode ser acrescida de mais informações, como a quantidade de *links*, por exemplo. Isto é possível devido ao fato da maior parte de ruídos em páginas *web* serem consituídos de *links* para outras páginas. Assim, outra medida é criada, a Densidade de Texto Composta, que incorpora o número de caracteres de *links* (CL_i) em cada sub-árvore, juntamente com o número de elementos de *links* (TL_i) existentes em cada ramificação. A Densidade de Texto Composta pode ser calculada através da fórmula abaixo:

$$DTC_i = \frac{C_i}{T_i} \log_{\ln(\frac{C_i}{\neg CL_i} CL_i + \frac{CL_b}{C_b} C_i + e)} (\frac{C_i}{CL_i} \frac{T_i}{TL_i})$$

Além do número de *links*, a fórmula é composta também pelo número de caracteres que não são *links* ($\neg CL_i$), pelo número total de caracteres de *links* abaixo da *tag* <body> (CL_b), e pelo número de caracteres de texto abaixo de <body> (C_b). Quando qualquer denominador é zero, tal valor é ajustado para um. Nesta fórmula, observa-se a existência de uma proporção entre caracteres de texto e caracteres de *links* ($\frac{C_i}{CL_i}$), assim como uma proporção entre *tags* comuns e *tags* de *links* ($\frac{T_i}{TL_i}$). Contudo, após o cálculo da Densidade de Texto Composta, alguns nodos ruidosos podem receber erroneamente altos valores de densidade, assim como nodos de conteúdo podem receber valores baixos. Desta maneira, nodos de conteúdo acabariam sendo descartados na fase de extração. Para resolver o problema, os autores propuseram outra medida, chamada de Soma das Densidades, que é representada pela soma de todas as Densidades de Texto Compostas dos descendentes de um determinado elemento.

3. INCORPORANDO EVIDÊNCIAS OBTIDAS EM FASES ANTERIORES DA EXPLORAÇÃO DA *WEB* OCULTA

Visto que este trabalho está inserido no contexto da *web* oculta, é possível melhorar os resultados da remoção de ruídos fazendo uso de dados obtidos em fases anteriores da exploração da *web* oculta. Para isto, são coletados todos os rótulos e valores utilizados para preencher os campos de todos os formulários submetidos. Este conjunto de termos representa uma grande *query* de um domínio para ser buscada nas páginas *web* renderizadas dinamicamente. Estes novos dados são coletados a partir do trabalho de [Kantorski et al. 2012], que explora técnicas de preenchimento automatico de formulários, os quais representam pontos de entrada da *web* oculta.

Assim, para cada nodo da árvore *DOM* da página *web* resultante, são criadas novas evidências baseadas nos dados previamente coletados. Uma destas evidências é a *QueryScore*, que representa a soma do número de vezes que um termo da *query* é encontrado em algum dos nodos descendentes, divido pela distância relativa. Essa distância é o número de nodos existente entre o nodo descendente que contém o texto até o nodo ancestral que está sendo calculado. Esta medida será relevante para todos os nodos que contém não apenas texto, mas para nodos contendo texto e outras sub-árvores. Também são utilizadas evidências mais simples, como o número simples de correspondências entre o texto diretamente abaixo do nodo e a *query* (gerada a partir dos rótulos e valores), e a distância absoluta entre o nodo sendo processado e a *tag* <body> do documento.

A fim de resolver o problema de agregação entre as medidas criadas na *baseline* (Densidade de Texto Composta e Soma das Densidades) e estas novas evidências, pode ser utilizada uma abordagem de aprendizagem de máquina. Empregando um classificador do tipo árvore de decisão, é possível classificar os nodos de texto em ruído ou não. Desta forma, para cada conjunto de dados é criado um arquivo contendo 6 atributos: Densidade de Texto Composta, Soma das Densidades, *QueryScore*, correspondências simples e distância absoluta. A última coluna do arquivo indica se um nodo é ruído ou não, verificado juntamente à avaliação manual de cada código *HTML* resultante.

Tabela I. Resultados da *baseline* e da abordagem supervisionada (TP) com conjunto de dados da *web* oculta

Conjunto de dados	Precisão-baseline	Revocação-baseline	Precisão-TP	Revocação-TP	F1-TP
e4s	38.09%	47.75%	98.3%	98.4%	98.3%
missing	60.28%	90.42%	99.7%	99.7%	99.6%
pubs	23.34%	56.64%	99.8%	99.8%	99.8%
drinkgenius	41.82%	65.32%	99.8%	99.8%	99.8%
onlinerace	25.33%	17.06%	99.2%	99.2%	99.2%
posteritati	00.34%	23.46%	99.7%	99.7%	99.6%
book	89.05%	88.97%	97.5%	98.3%	97.5%
career	56.85%	83.55%	98.7%	98.5%	98.5%
mldb	14.91%	36.53%	96.6%	95.6%	95.9%

Para classificar os nodos em ruído ou não, é utilizado um classificador de árvore de decisão, implementado no algoritmo C4.5. Esse método possui duas fases, a fase de treinamento e a fase de teste. A entrada do algoritmo, o arquivo com os atributos previamente apresentado, é dividido em dois grupos. O primeiro grupo é empregado na fase de treinamento, e a partir deste conjunto é construída uma árvore onde cada folha é um possível valor de classe gerado pelo classificador. Um caminho completo da árvore descreve as decisões feitas baseadas nos atributos de cada nodo para, depois de seguir o caminho, chegar a um resultado, neste caso ruído ou não. Na segunda fase, cada nodo do grupo de teste é experimentado contra esta árvore, e verificado se o resultado é correto ou não. Desta maneira é possível obter a precisão e revocação do método proposto.

4. EXPERIMENTOS

Para validar a hipótese de que o uso de informações da *web* oculta pode auxiliar a identificação de ruído em páginas *web*, implementou-se a solução proposta em [Sun et al. 2011]. Primeiramente, utilizou-se os conjuntos de dados fornecidos pelos autores da técnica em seu trabalho, e obteve-se os valores indicados no trabalho em questão. O conteúdo das páginas deste conjunto foi extraído e avaliado manualmente pelos autores da *baseline*, e todos estes dados foram coletados do *website* dos autores. Introduziu-se então neste trabalho outros conjuntos de dados, um grupo restrito dos dados utilizados em [Kantorski et al. 2012]. Os conjuntos de dados utilizados são: *e4s*, uma ferramenta de busca de empregos para estudantes; *missing*, um formulário para busca de pessoas desaparecidas; *pubs*, um motor de busca de bares; *drinkgenius*, um formulário para procura de receitas e drinques; *onlinerace*, uma busca de resultados de corridas de carro; *posteritati*, uma ferramenta para procura de cartazes de filmes; *book*, uma busca de livros à venda; *career*, uma ferramenta para procura de empregos e *mldb*, uma busca em um banco de dados de música e letras. Nestes formulários de entrada para a *web* oculta, diversas combinações possíveis de preenchimento são realizadas, já que cada preenchimento leva a uma página diferente, resultante de uma consulta em um banco de dados. Após a submissão de cada uma destas combinações, foi possível executar a implementação da *baseline* com o código *HTML* resultante. Como estes dados não estavam avaliados, selecionou-se um grupo de páginas resultantes após a submissão dos formulários para a avaliação manual, definindo quais elementos fazem parte do conteúdo e quais são ruído. O conteúdo correto foi então extraído e armazenado em um arquivo separado, para os testes a seguir.

Optou-se por utilizar uma abordagem de aprendizagem de máquina, de maneira que fosse possível combinar as medidas criadas na *baseline* (Densidade de Texto Composta e Soma das Densidades) com as evidências derivadas dos termos coletados no preenchimento dos formulários. Tais evidências são constituídas pelo que foi chamado de *QueryScore*, que leva em consideração o número de correspondências entre termos da *query* e termos abaixo de um nodo, junto com todos seus descendentes e a distância relativa ao nodo sendo calculado. Além disso, também adicionou-se o número de correspondências entre a *query* e termos diretamente abaixo do nodo sendo analisado. Por fim, adicionou-se a distância absoluta do nodo da árvore até o elemento <body>.

Para cada conjunto de dados, agrupou-se todos os nodos de todas as páginas *HTML* resultantes, juntamente com o valor resultante da avaliação manual que indica se tal nodo é conteúdo ou ruído. Com esta informação, foi possível utilizar uma abordagem supervisionada de aprendizagem para determinar se um nodo é ruído ou não. Para esta tarefa, utilizou-se a ferramenta de mineração Weka [Hall et al. 2009], que permite ao usuário realizar uma limpeza no conjunto de dados e possui diversos algoritmos de aprendizagem de máquinas para serem aplicados. Utilizando o algoritmo de árvore de decisão J48 (implentação Java do algoritmo de classificação C4.5), os dados foram divididos em dois grupos: o primeiro, que possui 66% dos nodos *HTML* e é utilizado para a fase de treinamento do algoritmo, e o segundo, contendo o restante dos nodos a serem utilizados para a fase de teste. Como existiam muito mais nodos ruído, o resultado da fase de teste é uma média ponderada dos resultados de cada categoria (ruído ou não), levando-se em consideração os resultados de precisão e revocação.

Para avaliar os resultados da implementação da *baseline*, foi utilizado um subconjunto das métricas contidas em [Sun et al. 2011], entre elas a Precisão, Revocação, e F1. O conteúdo extraído é comparado ao conteúdo correto, da seguinte maneira: para calcular a precisão, encontra-se a proporção entre o tamanho da substring comum mais longa ($LCS(a, b)$) de ambos os resultados e do tamanho do conjunto de texto extraído. Para calcular a revocação, busca-se a proporção entre a substring comum mais longa entre os resultados e o tamanho do conjunto de texto correto. Desta maneira, é possível calcular a F1, precisão média e revocação média para cada conjunto de dados. Testando a implementação da *baseline* com os conjuntos de dados dos autores, obteve-se resultados semelhantes aos originais descritos no artigo *baseline*.

Entretanto, ao executarmos a implementação da *baseline* com o conjunto de dados da *web* oculta, os resultados foram muito abaixo dos resultados fornecidos pelos autores em [Sun et al. 2011]. Isto provavelmente acontece pelo fato da técnica não estar preparada para lidar com páginas contendo dados tabulares em sua grande maioria. Então torna-se clara a necessidade de uma outra abordagem para lidar com este tipo de páginas *web*, contendo grandes porções de dados de forma tabulada. A segunda metade da tabela I mostra a execução do algoritmo de árvore de decisão utilizando dados da *web* oculta incorporados como evidência, e deixa claro a melhora obtida. Observando-se a matriz de confusão resultante do algoritmo de classificação, pode-se perceber que os resultados da detecção de ruídos foram positivos, já que foi possível classificar nodos ruidosos com altos valores de precisão e revocação. Todos os conjuntos de dados obtiveram valores de precisão e revocação acima de 95%. A matriz de confusão mostra que a técnica desenvolvida atinge bons resultados para detectar ruído, que é o objetivo principal deste trabalho, mas deixa a desejar para classificar nodos de conteúdo em si.

5. CONCLUSÃO

Neste trabalho, aborda-se o problema da eliminação de ruídos em páginas *web*. Este ruído, composto por todos os tipos de elementos não informativos ao usuário final, pode afetar negativamente o desempenho de motores de busca, índices *web*, e mineração de dados, como classificação e clusterização. Propõe-se neste trabalho a incorporação de novas medidas às já existentes em [Sun et al. 2011], utilizando dados obtidos a partir da segunda fase da exploração da *web* oculta, como os rótulos e valores dos preenchimentos dos formulários que dão acesso às páginas renderizadas dinamicamente. Acrescendo estas novas evidências, é possível observar uma grande melhora dos resultados para as páginas da *web* oculta através da utilização de um algoritmo de aprendizagem de máquina. Com uma abordagem supervisionada, um algoritmo classificador de árvore de decisão pode obter resultados médios de precisão e revocação acima de 95%, mostrando a melhoria gerada.

Após os experimentos executados este trabalho, é possível identificar espaço para muitas melhorias nos resultados. A melhor opção seria focar na melhoria da descoberta do conteúdo, não apenas do ruído, já que os resultados da descoberta de ruído foram suficientemente bons, mas não a busca pelo conteúdo em si. Para atingir este objetivo, seria possível testar muitos algoritmos de classificação diferentes com múltiplas escolhas de parâmetros, levando a um melhor entendimento das mudanças

dessas propriedades na descoberta do ruído. Ainda poderiam ser explorados outros tipos de conjuntos de dados, não apenas aqueles resultantes de submissão de formulários. Estas experimentações parecem vitais para a descoberta de novas possibilidades em relação ao tema abordado.

REFERENCES

Bar-Yossef, Z. and Rajagopalan, S. Template detection via data mining and its applications. In *Proceedings of the 11th international conference on World Wide Web*. WWW '02. ACM, New York, NY, USA, pp. 580–591, 2002.

Baroni, M., Chantree, F., Kilgarriff, A., and Sharoff, S. Cleaneval: a competition for cleaning webpages.

Bergman, M. K. The Deep Web: Surfacing Hidden Value. *JEP: The Journal of Electronic Publishing* 7 (1), 2001.

Burget, R. and Rudolfova, I. Web page element classification based on visual features. In *Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on*. pp. 67 –72, 2009.

Cai, D., Yu, S., rong Wen, J., ying Ma, W., Cai, D., Yu, S., rong Wen, J., and ying Ma, W. 1 vips: a vision-based page segmentation algorithm, 2003.

Chen, L., Ye, S., and Li, X. Template detection for large scale search engines. In *Proceedings of the 2006 ACM symposium on Applied computing*. SAC '06. ACM, New York, NY, USA, pp. 1094–1098, 2006.

Debnath, S., Mitra, P., and Giles, C. L. Automatic extraction of informative blocks from webpages. In *Proceedings of the 2005 ACM symposium on Applied computing*. SAC '05. ACM, New York, NY, USA, pp. 1722–1726, 2005.

Fernandes, D., de Moura, E. S., Ribeiro-Neto, B., da Silva, A. S., and Goncalves, M. A. Computing block importance for searching on web sites. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. CIKM '07. ACM, New York, NY, USA, pp. 165–174, 2007.

Gibson, D., Punera, K., and Tomkins, A. The volume and evolution of web page templates. In *Special interest tracks and posters of the 14th international conference on World Wide Web*. WWW '05. ACM, New York, NY, USA, pp. 830–839, 2005.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The weka data mining software: an update. *SIGKDD Explor. Newsl.* 11 (1): 10–18, Nov., 2009.

Kantorski, G. Z., Moraes, T. G., Moreira, V. P., and Heuser, C. A. Choosing values for text fields in web forms. In *ADBIS (2)*. pp. 125–136, 2012.

Kohlschütter, C., Fankhauser, P., and Nejdl, W. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*. WSDM '10. ACM, New York, NY, USA, pp. 441–450, 2010.

Kovacevic, M., Diligenti, M., Gori, M., and Milutinovic, V. Recognition of common areas in a web page using visual information: a possible application in a page classification. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. pp. 250 – 257, 2002.

Kushmerick, N. Learning to remove internet advertisements. In *Proceedings of the third annual conference on Autonomous Agents*. AGENTS '99. ACM, New York, NY, USA, pp. 175–181, 1999.

Li, J. and Ezeife, C. Cleaning web pages for effective web content mining. In *Database and Expert Systems Applications*, S. Bressan, J. Kng, and R. Wagner (Eds.). Lecture Notes in Computer Science, vol. 4080. Springer Berlin / Heidelberg, pp. 560–571, 2006.

Lin, S.-H. and Ho, J.-M. Discovering informative content blocks from web documents. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '02. ACM, New York, NY, USA, pp. 588–593, 2002.

Raghavan, S. and Garcia-Molina, H. Crawling the hidden web. Technical Report 2000-36, Stanford InfoLab, 2000.

Song, R., Liu, H., Wen, J.-R., and Ma, W.-Y. Learning block importance models for web pages. In *Proceedings of the 13th international conference on World Wide Web*. WWW '04. ACM, New York, NY, USA, pp. 203–211, 2004.

Sun, F., Song, D., and Liao, L. Dom based content extraction via text density. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. SIGIR '11. ACM, New York, NY, USA, pp. 245–254, 2011.

Vieira, K., da Silva, A. S., Pinto, N., de Moura, E. S., Cavalcanti, J. a. M. B., and Freire, J. A fast and robust method for web page template detection and removal. In *Proceedings of the 15th ACM international conference on Information and knowledge management*. CIKM '06. ACM, New York, NY, USA, pp. 258–267, 2006.

Wang, Y., Fang, B., Cheng, X., Guo, L., and Xu, H. Incremental web page template detection by text segments. *Semantic Computing and Systems, IEEE International Workshop on* vol. 0, pp. 174–180, 2008.

Weninger, T., Hsu, W. H., and Han, J. Cetr: content extraction via tag ratios. In *Proceedings of the 19th international conference on World wide web*. WWW '10. ACM, New York, NY, USA, pp. 971–980, 2010.

Yi, L., Liu, B., and Li, X. Eliminating noisy information in web pages for data mining. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '03. ACM, New York, NY, USA, pp. 296–305, 2003.