Departamento de Informática

Extração de conteúdo em páginas da web

Aluno: Pedro Lazéra Cardoso
Orientador: Eduardo Sany Laber

Antecedentes

Na primeira fase da Iniciação Científica, foi realizado um estudo dos conceitos básicos de algoritmos e programação, visto que o aluno estava dando os primeiros passos na área. Dentre os tópicos estudados, podemos destacar os seguintes: estrutura de dados, algoritmos em grafos e análise de algoritmos. Muitos dos algoritmos estudados foram implementados na linguagem de programação C ANSI.

Após essa etapa, o aluno começou a estudar a linguagem de programação Python, visto que esta é bastante prática para processamento de textos. Para praticar a linguagem, o aluno implementou alguns procedimentos simples para extrair conteúdo relevante de tabelas da web.

Na fase atual, estamos trabalhando no sentido de entender e aprimorar a heurística de extração de conteúdo relevante de páginas web, desenvolvida no laboratório em que o aluno realiza sua pesquisa. O projeto foi iniciado em janeiro, e o período de sete meses foi insuficiente para a sua conclusão. Esse relatório se foca somente na fase atual da iniciação científica, que será explicada a seguir.

Introdução

A disseminação do uso da internet na última década fez com que a linguagem HTML evoluísse e se tornasse mais complexa. Esse fato é claramente comprovado pela presença maciça de elementos de apresentação em páginas da web, como menus dinâmicos, conteúdo em flash, banners comerciais etc. Essa diversidade de possibilidades para o design de páginas contribuiu para que a maneira como os sites são construídos se tornasse bastante heterogênea e os elementos de apresentação do layout ganhassem importância. De fato, Gibson [5] estima que entre 40% e 50% de todo o conteúdo da web seja formado por esses elementos de apresentação de layout.

Figura 1. Exemplo de página com alta proporção de conteúdo irrelevante

Se por um lado essa gama de recursos alavancou a popularização da internet, ao permitir que qualquer pessoa fosse capaz de navegar através dela, por outro lado causou um efeito indesejado: aumentou em muito a dificuldade de os mecanismos de buscas operarem. Isso ocorre porque esses mecanismos, para estabelecer a lista de páginas relacionadas a um assunto, usam como base o conjunto de palavras das páginas. Parte desse conjunto, como menus e anúncios, pode não ter relação nenhuma com o assunto procurado, e assim as tarefas de busca e de ranking de páginas podem ficar comprometidas. *Tokens, frases* e *named-entites* pertencentes à sessão de anúncios são muitas vezes indistinguíveis daqueles presentes no corpo e no título de uma notícia, por exemplo.

A maneira de contornar essa dificuldade é extrair do conjunto de palavras de uma página seu conteúdo relevante. Por relevante, entendemos o texto que melhor representa o conteúdo de uma notícia ou de um blogpost, excluindo figuras, comentários e quaisquer elementos de apresentação.

Uma das soluções, talvez a mais difundida, para o problema de extração é a de renderizar a página, de uma forma muito semelhante ao que um navegador iria fazer para um humano, e explorar os atributos visuais obtidos a partir dela. Esse processo possibilita uma alta qualidade de extração, mas seu custo computacional é elevado.

Isso motivou a criação do NCE, uma heurística capaz de desempenhar essa tarefa com menor custo computacional [4]. No projeto de iniciação científica, estamos trabalhando no sentido de melhorar os resultados apresentados por essa heurística. Nas sessões a seguir, a maneira como o algoritmo funciona e os detalhes do projeto serão explicados com mais detalhes.

Um pouco mais sobre o NCE

O NCE é uma heurística que, a partir da análise da árvore DOM de uma página da web, desempenha a tarefa de extrair do seu conteúdo de texto a parte considerada relevante. O projeto foi focado no conteúdo de páginas de notícias e blogposts, e um de seus objetivos era garantir eficiência computacional.

Além disso, buscou-se tornar o código reutilizável - as únicas tags analisadas são as de link <a>, parágrafo <p> e título <title>. A utilização exagerada de tags poderia tornar o NCE obsoleto no futuro.

Não nos preocuparemos em explicar como o NCE funciona em detalhes – basta apenas entender que seu objetivo é extrair conteúdo relevante em páginas de news e de blogposts e que isso é feito marcando numa árvore DOM os nós cujo conteúdo é relevante. Por fim, também é importante ressaltar que seu código foi escrito inicialmente em Python, mas depois migrado para C Sharp.

Já existe uma primeira versão do projeto capaz de produzir resultados competitivos quando confrontada com outras heurísticas de extração da web, segundo critérios de velocidade e qualidade. No projeto de iniciação científica, portanto, não era nosso objetivo fazer alterações em seu código principal.

Motivações do Projeto de Iniciação Científica

Para estimar o grau de sucesso do NCE, foram usadas duas medidas, o *Recall* e o *Precision*.

1- Recall: quantidade de palavras que o NCE selecionou corretamente / quantidade de palavras relevantes.

2- Precision: quantidade de palavras que o NCE selecionou corretamente / quantidade de palavras que o NCE selecionou, corretamente ou não.

A seguir, resumidos em uma tabela, seguem alguns dados a respeito dos resultados do NCE, segundo essas duas medidas.

Tabela I – Resultados do NCE

NCE		
Página	**Recall (%)**	**Precision (%)**
dailyunion.com	99,6	97,7
iht.com	99,1	97,7
eastandard.net	98,9	97,1
nationaudio.com	99,9	95,3
nysun.com	98,8	94,1
thevalleychronicle.com	97	95,7
seattletimes.nwsource.com	95,8	95,1
news24.com	99,1	91
theadvertiser.com	97	93,2
dublinpeople.com	98,4	91
articles.lancasteronline.com	97,3	88,8
bradenton.com	96,1	91,1
greenvilleonline.com	93,7	91,7
beaumontenterprise.com	99,9	83,1
silive.com	95,8	87,2
delmarvanow.com	96,1	86,1
thehawkeye.com	98,2	83,2
news.sky.com	92,1	86
wspa.com	93,3	82,7
gtowntimes.com	89,7	76,6
baltimoresun.com	83,9	88,9
watfordobserver.co.uk	93	54,3
Média das páginas	**96,0**	**88,5**

Com base no corpus que usamos para avaliar o NCE, seu *Recall* supera notavelmente seu *Precision*. A Tabela I endossa essa percepção. Isso significa dizer que a heurística em geral não deixe de selecionar o conteúdo relevante de uma página, mas não tem tanto sucesso em ignorar aquilo que é irrelevante. Isso motiva, portanto, a idéia de escrever uma heurística de pós-processamento que seja capaz de remover o conteúdo que foi selecionado incorretamente pelo NCE.

Objetivos

O projeto tem como objetivo melhorar a precisão dos resultados do NCE – News Contet Extractor – heurística que extrai conteúdo relevante de páginas da web, a partir da análise de suas árvores DOM. Por relevante, entendemos o texto que melhor representa o conteúdo de uma página de notícia ou de um blogpost, excluindo figuras, comentários e quaisquer elementos de apresentação.

Em síntese, procuramos corrigir alguns erros de excesso do NCE, o que significa tentar buscar, naquilo que foi marcado como relevante, nós de texto que na verdade são irrelevantes, ou excesso.

Para isso, foram desenvolvidos métodos para identificar conteúdos de texto semelhantes de um conjunto qualquer de árvores DOM. Essa identificação serviu como pós-processamento para a heurística NCE.

É essencial ressaltar que o NCE, no período em que a pesquisa foi realizada, extraía conteúdo apenas de blogposts e páginas de notícias, e a heurística de pós-processamento só se refere a esses dois tipos de páginas.

Metodologia

A heurística de pós-processamento do NCE é baseada na hipótese de que um texto presente em muitas árvores DOM é provavelmente conteúdo irrelevante. Foi em cima dessa premissa que o projeto de iniciação científica começou a ser desenvolvido.

Dado um conjunto de árvores DOM, é possível localizar os nós de texto de cada árvore através de uma busca em profundidade. Nessa busca, um identificador de cada nó de texto é armazenado numa estrutura de hash. Esse identificador é composto por três campos:

1- o “id” do documento (árvore);
2- o “id” do nó da árvore;
3- o valor de retorno da função hash que é aplicada sobre o nó, explicada adiante.

Escolhemos a estrutura hash com o objetivo de utilizar pouca memória e ter rápido acesso aos dados armazenados, considerando que um dos principais objetivos do NCE é ter custo computacional baixo.

Para determinar em que posição da tabela o identificador de um nó é guardado, aplica-se uma função de hash em seu texto. Aproveitando que o código foi inicialmente implementado em Python, usamos sua a função padrão de hash. A seguir, de acordo com o tamanho da tabela de hash que se deseja utilizar, esse valor é manipulado de modo a não violar seus limites. Manipular significa utilizar o resto da divisão do valor de retorno da função hash por um certo parâmetro *p*, que é determinado empiricamente. Essa determinação deve se basear em dois objetivos: garantir alta probabilidade de que elementos associados a uma mesma posição sejam idênticos e evitar que a estrutura de armazenamento fique com muitas posições vazias.

Depois de percorrer todas as árvores e armazenar os identificadores de nós de texto, é possível verificar se um certo texto está presente em muitas delas analisando a estrutura de hash. Se o identificador de um nó está associado a uma posição da tabela com um grande número de colisões, isso indica que o texto desse nó ocorre em muitas das árvores.

Conjeturando que todo texto presente em muitas árvores é provavelmente conteúdo irrelevante, cada nó de texto com essa característica é removido da árvore. Nesse caso, remover significa marcar com algum atributo. Exemplos ilustrativos do que a ferramenta pode remover são o conteúdo de Copyright do pé das páginas e anúncios que se repetem em muitos sites.

Figura 2. Texto de copyright de páginas de notícias (UOL, CNN, Folha) em destaque

Em seguida, utilizando ferramentas para medir precisão e revocação, podemos descobrir como tal algoritmo afeta o resultado da extração de conteúdo relevante obtido pela heurística NCE.

É importante fazer algumas observações. Primeiro, essas árvores não representam páginas de web inalteradas – cada uma das páginas é parte de uma outra página que o NCE (News Content Extractor) marcou como relevante. Finalmente, esse pós-processamento da heurística foi programado usando o Python, em razão da facilidade de se escrever códigos com esta tecnologia, de sua eficiência e da experiência que o grupo do laboratório teve com ela em projetos anteriores.

Na fase atual do projeto, estamos desenvolvendo as ferramentas para medir precisão e revocação, ao mesmo tempo em que é feita a migração de Python para C Sharp. Nossa motivação principal para essa migração é o aumento de velocidade da heurística de pós-processamento.

Conclusões

Estudar os conceitos básicos de programação e de algoritmos foi essencial para que o aluno pudesse entender algumas das técnicas utilizadas em processamento textual.

Na fase atual, onde estamos desenvolvendo métodos para filtrar textos que ocorrem muitas vezes em um conjunto de páginas, o aluno está tendo a oportunidade de colaborar diretamente com uma pesquisa na área de extração de conteúdo relevante de páginas.

Ainda não podemos avaliar como a heurística afeta a precisão do NCE, porque estamos numa etapa intermediária, em que alguns parâmetros ainda precisam ser determinados e alguns detalhes, ajustados.

No entanto, já é possível afirmar que o pós-processamento está identificando corretamente conteúdos irrelevantes – só não sabemos ainda o quanto significativa é essa melhora. Além disso, a heurística tem baixo tempo de execução, conforme os objetivos do NCE.

Finalmente, é possível afirmar que nossa heurística pode ser estendida para identificar outros objetos diferentes de textos em estruturas diferentes de páginas da web. O que vai determinar o quanto a heurística deve ser modificada são os objetos e as estruturas com os quais ela vai trabalhar.

Referências

1 - KLEINBERG, Jon; TARDOS, Eva. **Algorithms Design**.

2 - TAN, Pang-Ning; KUMAR, Michael Steinbach Vipin. **Introduction to Data Mining**.

3 – TENGLI, Ashwin; YANG, Yiming; MA, Nian Li. Learning **Learning Table Extraction from Examples**. Pittsburgh, PA: Carnegie Mellon University

4- AMORIN, E. ; CARDOSO, E. T. ; LABER, E. S. ; JABOUR, I. ; SOUZA, C; TINOCO, L.; RENTERIA, R.; VALENTIM, C.. **A fast and simple method for extracting relevant content from news webpages.** IN ACM CIKM 2009

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