Universidade Federal de Pernambuco
Centro de Informática

Graduação em Ciência da Computação

Um Coletor Inteligente para Entidades Imobiliárias Estruturadas em Sites

Bertha Maria Correia Andaluz

Trabalho de Graduação

Recife
16 de Dezembro de 2016

Universidade Federal de Pernambuco
Centro de Informática

Bertha Maria Correia Andaluz

Um Coletor Inteligente para Entidades Imobiliárias Estruturadas em Sites

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Prof. Luciano de Andrade Barbosa*

Recife
16 de Dezembro de 2016

Agradecimentos

Em primeiro lugar, agradeço a Deus o cuidado, o carinho e o sustento infalíveis: Tua seja toda honra e toda glória.

Ao professor Luciano de A. Barbosa que tornou possível este trabalho. Obrigada pela orientação ao longo deste semestre e por estar sempre disposto a tirar dúvidas e ajudar.

Ao corpo docente. Em especial ao professor Pedro Manhães que, acredito, desconhece o tremendo impacto que teve nas vidas acadêmicas e pessoais de muitos de nós ingressados no primeiro semestre de 2012; à professora Kátia Guimarães a quem sou imensamente grata pelo exemplo de grande profissional e grande pessoa e finalmente ao professor Paulo Gustavo Fonseca, que tem se mostrado mais que apenas professor para muitos de nós, seus alunos e orientandos: professor, *keep up the good work*.

Ao projeto da Maratona de Programação, parte fundamental da minha formação, no qual tive a oportunidade de conhecer e aprender de pessoas brilhantes. Agradeço às professoras Liliane Salgado e Kátia Guimarães por todos os anos de dedicação a esse projeto.

Ao PET-Informática, através do qual tive a oportunidade de estar em contato com colegas excepcionais e desenvolver atividades de ensino, pesquisa e extensão relevantes para alunos da graduação e também para a comunidade do Centro.

Aos amigos feitos ao longo da graduação. Em especial aos amigos do Lontra: Lucas Lima, Duhan, Eduardo, Guilherme, João Pedro, Larissa, Leonardo, Lucas Netto, Maria Gabriela, Marina, Mateus, Rafael Acevedo, Rafael Francisco, Raíssa e Vinícius, eu não poderia ter sorte maior. Qualquer dedicatória seria incapaz de traduzir o quanto devo e o orgulho que sinto de cada um de vocês!

A todos os funcionários do Centro de Informática que a cada dia empenham-se e contribuem para que este seja um lugar de excelência.

Por fim, agradeço a minha família: agradeço o exemplo de amor incondicional, a dedicação incansável e o papel desempenhado por cada um na formação da pessoa que sou hoje.

What we find changes who we become.
—PETER MORVILLE

Resumo

A World Wide Web é uma grande fonte de entidades estruturadas. Entidades podem ser definidas como objetos do mundo real com seus atributos e valores associados. Exemplos de entidades estruturadas na Web são especificação de produtos e infoboxes no Wikipédia. O valor prático de repositórios compreensivos de entidades estruturadas de um domínio específico é facilmente observável, por exemplo, na melhoria de resultados para consultas de entidades - o tipo mais frequente de buscas após consultas navegacionais - e decisões baseadas em dados. Antes de poderem-se utilizar esses repositórios, é necessário que essas entidades estruturadas sejam coletadas. O objetivo principal deste trabalho é implementar uma solução eficaz para o sub-problema de localização e identificação de entidades estruturadas do domínio imobiliário em um mesmo website. Para localizar páginas de entidades, desenvolvemos uma estratégia que utiliza um classificador de links para predizer se um link aponta para uma página de entidade e, assim, evitar áreas improdutivas dos sites. Para detectar páginas de imóveis, criamos um classificador de páginas que idenitifica com alta precisão páginas com entidades imobiliárias.

Palavras-chave: recuperação de informação, localização de entidades, identificação de entidades, dados estruturados, coleta de dados, web

Abstract

The World Wide Web is a great resource for structured entities. Entities can be defined as real world objects defined by their attributes and associated values. The practical value of comprehensive repositories on domain specific structured entities can easily be seen, for instance, on improving results on entity queries - the most frequent type of query search after navigational queries - and data-driven decisions. Before making use of those repositories, one must collect such structured entities. The main goal of this work is to implement an effective solution to the sub-task of locating and identifying pages containing relevant structured entities from the real estate domain within a website. To locate entitity pages, we developed a strategy that makes use of a link classifier to predict if a link points towards an entity page, thus avoiding unproductive areas of the websites. To detect real estate enitity pages, we built a page classifier that identifies pages with real estate entities with high precision.

Keywords: information retrieval, entity location, entity identification, structure data, data collection, web

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Vem-se observando um aumento de interesse na utilização de dados estruturados na Web para diversas aplicações como integração de dados, análises estatísticas e buscas por entidades [3], que são o tipo mais frequente de buscas após consultas navegacionais [4, 5]. Neste trabalho, voltamos nossa atenção a uma parcela específica dos dados Web estruturados que chamaremos de entidades estruturadas - dados que descrevem objetos do mundo real, como produtos, lugares e pessoas, através de seus atributos e valores associados.

A tarefa de coletar e organizar em um repositório todas essas entidades estruturadas presentes na Web, independentemente do domínio a que elas pertençam, demanda uma quantidade altíssima de recursos computacionais: a obtenção de um repositório completo requeriria a análise de bilhões [6, 7] de páginas Web. Levando isso em consideração, optamos por restringir nosso escopo a entidades pertencentes ao domínio específico de imóveis.

É fácil encontrar motivações práticas para repositórios de entidades de um domínio específico: ter informações como horário de funcionamento, meios de contato, localização e avaliações de todos os restaurantes em uma dada região; informações sobre todos os livros, suas avaliações, gêneros e autores; informações sobre artistas musicais, suas discografias e influências entre inúmeros outros possíveis exemplos.

O objetivo deste trabalho é localizar e detectar páginas de entidades imobiliárias estruturadas em um dado site. Esse problema possui uma escala menor em relação à de identificação de sites pertencentes ao domínio já que não é mais necessário navegar toda a Web. Todavia é de extrema importância que seja fornecida uma solução eficiente porque, para a obtenção de um repositório de entidades completo para o domínio em questão, é necessário visitar não apenas grandes agregadores de entidades, mas também sites periféricos [8]. O custo, portanto, de se visitar esse grande número de sites é bastante alto e para poder dimnuí-lo, precisa-se evitar regiões improdutivas desses sites.

CAPÍTULO 2

Fundamentos

Tarefas como a criação de repositórios compreensivos de entidades estruturadas demandam que seja possível explorar páginas da web de maneira automática. Essa exploração automática é alcançada com o uso de *crawlers*. Enquanto *crawlers* de propósito geral desempenham muito bem a tarefa de explorar a web como um todo, esses podem ser muito dispendiosos para problemas que envolvam coleta de dados de um domínio específico, já que boa parte dos recursos será gasta em páginas não pertencentes ao domínio. Para essa categoria de problemas, é mais vantajosa a utilização de *focused crawlers*. Em sua versão mais simples, o coletor emprega um classificador capaz de controlar a prioridade de visitação dos links presentes na fronteira do crawler, dessa forma podendo direcionar mais recursos à exploração mais profunda de páginas do domínio. Neste trabalho utilizamos um grau adicional de foco a fim de priorizar a descoberta de páginas de entidades do nosso domínio.

2.1 Aprendizado supervisionado

Em 1959, Arthur Samuel definiu aprendizagem de máquina como o campo de estudo que dá aos computadores a habilidade de aprender sem que sejam explicitamente programados. Aprendizagem de máquina pode ser sub-dividida em várias categorias; uma dessas é a aprendizagem supervisionada.

O objetivo do aprendizado supervisionado é inferir uma função a partir de exemplos rotulados para predizer a classificação de novos exemplos [9]. Cada exemplo utilizado no treinamento consiste em um par formado por um objeto de entrada e a classificação desejada para esse objeto.

As tarefas de aprendizagem supervisionada podem ser de regressão, quando a função inferida produz um resultado numérico, ou de classificação, quando o objetivo da função é determinar a que grupo um objeto pertence.

As características utilizadas para descrever os objetos possuem grande importância para o resultado da classificação. A escolha de *features* informativas, discriminativas e independentes é um passo crucial para algoritmos de classificação.

Ainda que o esperado seja que a função inferida produza resultados corretos inclusive para dados que não pertençam ao conjunto de treinamento, é possível que essa apresente alguns problemas. Caso a função apresente baixo desempenho até mesmo no conjunto de treinamento diz-se que há um *underfitting* dos dados e sugere-se que mais *features* sejam adicionadas, pois as características utilizadas não são suficientes para descrever bem o problema. Já quando a função apresenta alto desempenho no conjunto de treinamento e baixo desempenho no conjunto

de testes diz-se que há um *overfitting* do dados; neste caso sugere-se uma remoção de features pois o grau de especificidade das características utilizadas impede uma boa generalização.

2.2 Trabalhos relacionados

Tanto a utilização de *focused crawlers* quanto a recuperação automática de entidades não são problemas recentes.

Em [10] é apresentado o conceito e a primeira implementação de um *focused crawler*. Os autores definem o objetivo de coletores focados como recuperar páginas pertencentes a um conjunto de tópicos pré-definido de maneira seletiva. A solução proposta para alcançar esse objetivo consiste em dois programas de mineração de hipertexto: um classificador que avalia a relevância de um documento de hypertexto em relação ao conjunto de tópicos de interesse e um distilador que identifica pontos de acesso para várias páginas relevantes dentro de alguns links.

O trabalho proposto em [11] emprega um coletor focado em formulários para a localização automática de bancos de dados na *hidden-web*. O *crawler* desenvolvido pelos autores utiliza dois classificadores para auxiliar na busca por formulários: um classificador de páginas e um classificador de links. Utilizam ainda um terceiro classificador a fim de filtrar formulários inúteis. O *crawler* adota algumas medidas para lidar com a esparsidade dos formulários na *web*, dentre elas focar em um tópico específico.

O classificador de links aprende características dos links presentes na âncora, URL e texto próximo à URL e atribui uma nota referente à distância entre o link a ser explorado e uma página relevante, isto é, uma página de formulário pesquisável. Os autores empregam essa estratégia, em detrimento de selecionar apenas links que apontem diretamente para páginas alvo, para evitar a não coleta de páginas boas por essas estarem localizadas em uma profundidade maior.

A classificação de páginas é feita por meio de um classificador naive-bayes treinado a partir de amostras obtidas na taxonomia do Dmoz (dmoz.org). Ao explorar uma página, o classificador atribui uma probabilidade p de que essa seja pertencente ao tópico especificado e, caso p seja superior a 50% essa página é considerada pelo *crawler* como relevante.

Já em [12] o autor apresenta uma solução para a coleta de páginas conversacionais, *thread pages*, em fórums web. A proposta consiste em uma coleta em duas etapas: *crawlers* inter e intra-site.

A fim de detectar e descobrir sites de fórums de maneira precisa e eficiente, o coletor inter-site foca na vizinhança desses sites no grafo da Web e explora os padrões de conteúdo dos links nessa região como guia para determinar sua política de visitação.

Uma vez encontrado um site de fórums, o coletor intra-site é utilizado para localizar e detectar *thread pages*. Para tanto, o *crawler* explora os padrões dos links dentro de sites de fórums através de grafos de contexto, estando as páginas de *threads* localizadas no centro desses grafos. A principal suposição feita ao utilizar-se desse tipo de grafos é que links com distâncias iguais para o centro possuem contextos similares. O classificador de links é componente do *crawler* intra-site que faz uso das informações do grafo de contexto para localizar páginas de *threads*, isto é, utiliza as informações contextuais para estimar a distância de um dado link para

páginas de *threads*. O contexto é formado por tokens na URL, âncora e termos ao redor da âncora, que também são os atributos utilizados pelo classificador de links.

CAPÍTULO 3

Solução

Neste trabalho pretendemos desenvolver os classificadores utilizados para a localização e detecção de entidades estruturadas dentro de sites que permitam um desempenho eficiente do coletor intra-site. O escopo deste trabalho está restrito o domínio específico de imóveis.

3.1 Localização de Páginas de Entidades Imobiliárias

Na coleta de páginas em sites imobiliários, consideramos relevantes exclusivamente páginas de entidades imobiliárias, ou seja, páginas de anúncios ou descrição de um único imóvel, como exemplificado na Figura 3.1. Comumente existe nos sites um grande número de páginas não relevantes como páginas de notícias - que não contém entidades - e páginas de índice de imóveis - que podem conter várias entidades. É fundamental, portanto, para um bom desempenho do coletor, que regiões irrelevantes e improdutivas dos sites sejam evitadas sempre que possível.

A abordagem adotada para mitigar esse problema é a utilização de um classificador de links que nos fornece um indicativo do potencial de sucesso para cada link a fim de que páginas relevantes sejam visitadas mais rapidamente pelo coletor intra-site. Definimos sucesso como a exploração de um link que aponte para uma página de entidade imobiliária e potencial de sucesso de um link como a chance de encontrarmos uma página de entidade imobiliária ao explorá-lo. Esse indicativo gerado pelo classificador é utilizado pelo coletor intra-site para determinar a prioridade de exploração de cada link.

O fato de definirmos potencial de sucesso como um conceito contínuo, diferentemente da definição binária adotada para a relevância de páginas, é bastante conveniente para o problema em questão, pois permite que diferenciemos a prioridade de, por exemplo, páginas de índices de imóveis e páginas de notícias, sendo mais promissora a exploração de links do primeiro tipo.

Trabalhos anteriores mostram que informações presentes em HREFs, URLs ou fragmentos destas, trazem informações de grande valor para a determinação de que prioridade deve ser atribuída a links ainda não explorados [13] [11]. Nas palavras que constituem os links, é notável a presença frequente de termos fortemente relacionados a seus domínios como: *career* e *job* em domínios empregatícios; *car* e *used* em domínios automobilísticos[11] e *casa*, *aluguel* e *apartamento* no domínio abordado neste trabalho. Em mãos desse conhecimento, entendemos como uma boa estratégia utilizar os tokens existentes nas URLs e âncoras de links a serem explorados como atributos para a classificação de links desconhecidos. A esses tokens adicionamos o contexto que diz respeito a sua origem, isto é, se é um token proveniente da URL ou da âncora do link. Consideramos ainda algumas características estruturais sendo elas: quantidade de tokens da URL, quantidade de tokens da âncora, tamanho da URL, tamanho

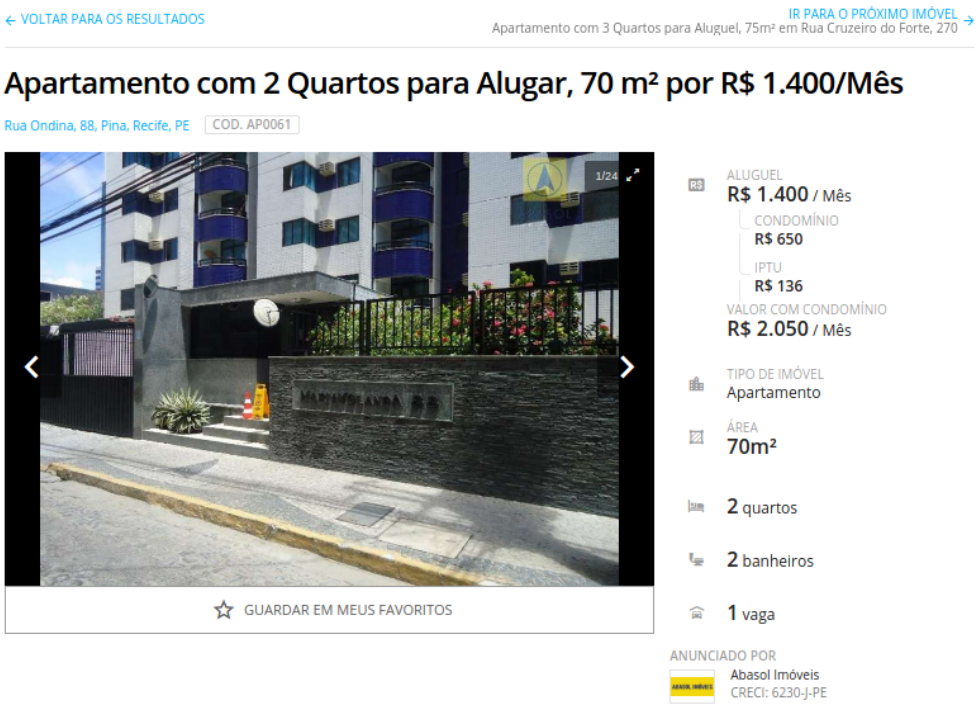


Figura 3.1: Exemplo de página de entidade imobiliária.

da âncora e quantidade de caracteres não alfa-numéricos, com exceção dos caracteres `'/'` e `'-'` (`[^a-zA-Z0-9\\-/]`).Vemos a distribuição dessas *features* nos conjuntos positivo e negativo na Figura 3.2.

Destacamos que por realizarmos um treinamento offline do classificador, não temos disponíveis nesse momento informações referentes a âncoras de links. Como alternativa, utilizamos informações presentes nos títulos das páginas e, como durante a classificação, tokens e tamanho das URLs. Pela maneira que a coleta de páginas para a construção do conjunto de dados foi feita, tinhamos disponíveis as informações presentes nas páginas, mas não a informação de em que páginas encontravam-se os links que apontavam para as páginas do conjunto de dados. A escolha de utilizar as informações presentes no título das páginas foi feita por observarmos uma distribuição de características parecidas entre títulos e âncoras.

Como mencionado anteriormente, o score final gerado pelo classificador de links é um indicativo da relevância de um dado link. Esse score é gerado de acordo com o Algoritmo 1:

O score é composto pela probabilidade gerada de que um link seja relevante somado a um fator aleatório. Além disso, atentando à existência de algumas URLs irrelevantes, porém cujos tokens fornecem fortes indicativos de uma alta relevância como `www.imoveisnovacpi.com.br/fotos/F00/80/102288000-07_COZINHA___APARTAMENTO_EM_PRAIA_GRANDE.JPG`, penalizamos duramente URLs contendo algumas substrings. Justificamos essas escolhas como forma de garantir que o coletor intra-site não selecione apenas páginas que tragam retorno imediato, mas ao mesmo tempo gaste menos recursos com páginas conhecidamente improdutivas. Adotamos essa heurística de penalizações ao identificarmos um

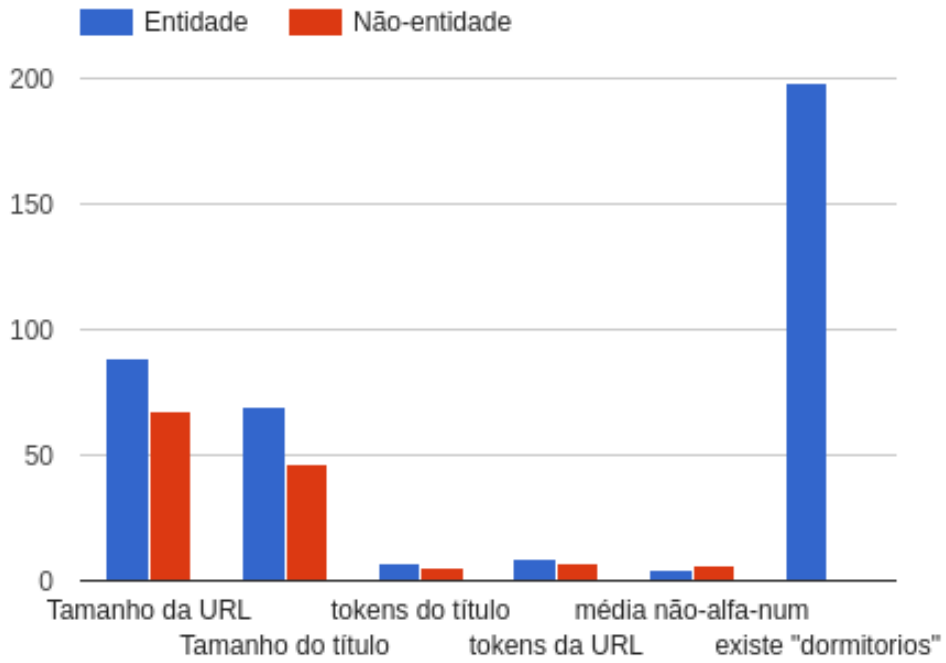


Figura 3.2: Distribuição nos conjuntos positivos e negativos das features estruturais: tamanho médio da URL e do título; número médio de tokens na URL e no título; número médio de caracteres não alfa-numéricos (a menos de `'/'` e `'-'`). Também exibimos o número de páginas que possuem `dormitorios` em seu título. Destacamos a utilização de tokens do título e não da âncora por não possuirmos dados referentes às âncoras no momento do treinamento.

Algoritmo 1 Definição do sinal final gerado pelo classificador de links.

```
1: function RELEVANCIAFINAL(classificador, instancia)
2:     relevancia ← classifier.distributionForInstance(instancia)
3:     red_flags ← ocorrencias(instancia.URL, "jpg")
4:     red_flags ← relevancia + ocorrencias(instancia.URL,"css")
5:     red_flags ← relevancia + ocorrencias(instancia.URL,"png")
6:     red_flags ← relevancia + ocorrencias(instancia.URL,"jpeg")
7:     if red_flags = 0 then relevancia ← relevancia + (random(10, 40) / 100.0)
8:     if red_flags > 0 then relevancia ← relevancia * 0.1
       return relevancia
```

problema quanto a diversidade de páginas do conjunto de treinamento: enquanto arquivos de imagens não estavam presentes no conjunto de treinamento, esses foram relativamente comuns durante a realização de nossos experimentos. Optamos pela heurística porque adicionar essas *features* ao classificador demandaria um grande esforço manual para rotular novas páginas.

3.2 Detecção de Páginas de Entidades Imobiliárias

Uma abordagem simples para classificar páginas como relevantes ou não seria utilizar a probabilidade estimada pelo classificador de links descrito na seção anterior, classificando como relevantes páginas cujos links apresentassem potencial de sucesso superior a um dado threshold. Essa abordagem, contudo, nos garantiria uma confiança aquém da desejada na classificação atribuída às páginas: facilmente encontramos exemplos de páginas de entidades com URLs extremamente genéricas como `http://classificados.diariodonordeste.com.br/click/225822` ou páginas de índices de entidades com URLs e âncoras com um vocabulário bastante similar ao de páginas de entidades. Exemplos como esses justificam a necessidade de um classificador específico para a detecção de páginas de entidades.

Pela similaridade de conteúdo entre páginas de entidades imobiliárias e páginas de especificações de produtos, exemplificada na Figura 3.3, inicialmente levantamos a hipótese de que a tarefa de identificar páginas de entidades imobiliárias seria similar à de identificar páginas de especificações. Nos estágios iniciais do desenvolvimento do classificador de entidades, baseados nos resultados de [14], construímos um classificador baseado em atributos estruturais: *número de links do mesmo domínio, de tabelas, de listas, de imagens* e *frequência do padrão 'R$'*.

Através de uma observação mais atenta dos dados, percebemos que, na realidade, as tarefas de detecção de páginas de entidades imobiliárias e de especificações de produtos são significativamente distintas: em um grande número de sites, páginas de entidades imobiliárias não possuem a estrutura esperada, como podemos notar na Figura 3.4.

Grande parte das informações relevantes e descritivas sobre essas entidades é encontrada em um ou mais parágrafos ao invés de em listas ou tabelas e, em não raros casos, aparentes listas de características de imóveis encontram-se em tags *div* ou *headers* e não em listas de fato.

Em vista dessa observação sobre o domínio de entidades imobiliárias, apoiados em [11], decidimos usar um modelo bag-of-words com tokens provenientes do corpo, URL e título das páginas em nosso classificador de entidades. Como o valor dos números dificilmente significa algo importante, codificamos os valores numéricos do corpo como `[# dígitos]DIGITS`.

Especialmente por extrairmos tokens do corpo das páginas, geramos mais de `45000` atributos em um primeiro momento. A fim de trazer mais rapidez ao processo de treinamento, mas ao mesmo tempo evitar o descarte de termos relevantes, utilizamos apenas termos que aparecessem ao menos `5` vezes no conjunto de treinamento. Com esse filtro reduzimos para `6523` o número final de atributos fornecidos ao classificador.

Empiricamente notamos que a existência de alguns termos é um sinal mais relevante que suas frequências, desta forma optamos por uma suavização logarítmica da frequência dos atributos. Optamos também pela adição de contexto aos tokens utilizados, significando a adição

(a) Página de entidade com listas

(b) Página de especificações

Figura 3.3: Exemplo de caso em que tanto uma página de especificações quanto de entidade imobiliária possuem a presença de listas.

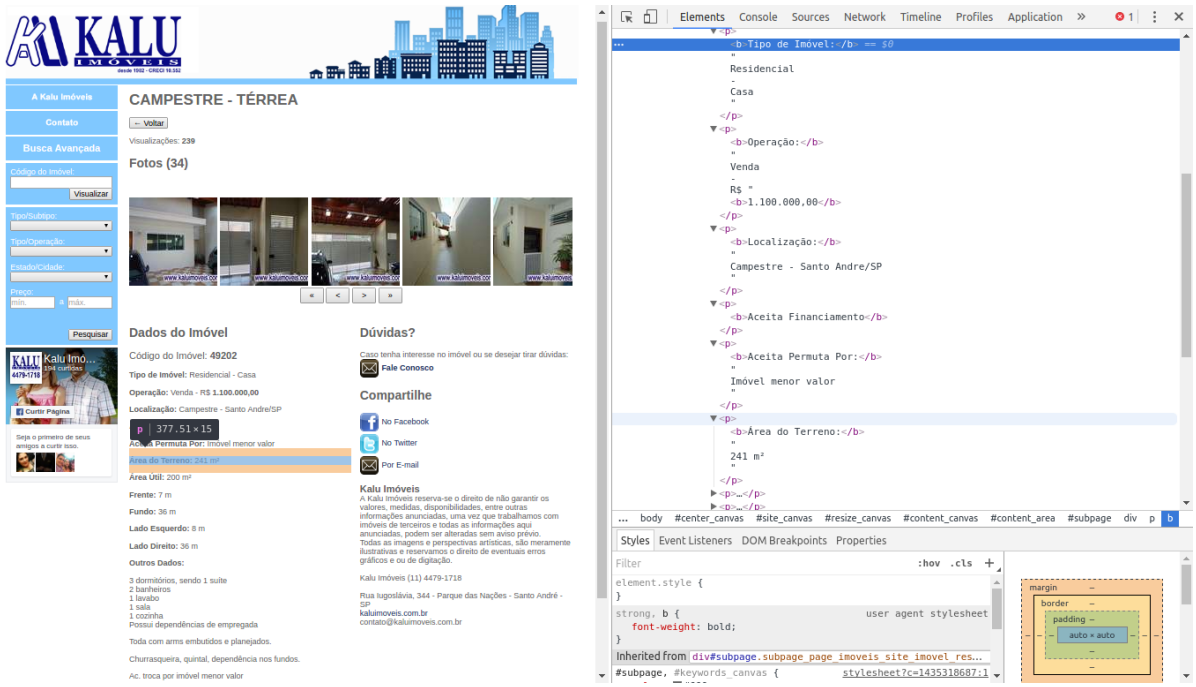


Figura 3.4: Exemplo de página de entidade que aparenta ter listas, porém os dados estão organizados em múltiplos parágrafos

aos tokens de informações que possibilitem determinar se esses foram extraídos da URL, do título ou do corpo da página.

A relevância final de uma página é determinada de acordo com o procedimento descrito no Algoritmo 2:

A relevância de uma página é penalizada de maneira proporcional à frequência de determinadas substrings: `"amp&"`, `"amp;"`, `"filtragem"`, `"filter"` e `"busca"`. Há ainda uma penalização constante caso saibamos que a página em questão é uma imagem. Novamente, justificamos essas heurísticas por observarmos a existência de páginas não relevantes, especialmente de imagens ou índices de entidades, cujas URLs possuem termos associados fortemente a páginas de entidades. Assim como no classificador de links, adotamos essa heurística de penalizações ao por identificarmos um problema em relação à diversidade de páginas do conjunto de treinamento. Optamos pela heurística porque adicionar essas *features* ao classificador demandaria um grande esforço manual para rotular novas páginas.

Algoritmo 2 Definição do sinal final gerado pelo detector de entidades.

```
1:  function RELEVANCIAFINAL(classificador, instancia)
2:      relevancia ← classifier.distributionForInstance(instancia)
3:      yellow_flags ← ocorrencias(instancia.URL, "amp%")
4:      yellow_flags ← yellow_flags + ocorrencias(instancia.URL,"amp;")
5:      red_flags ← ocorrencias(instancia.URL,"filtragem")
6:      red_flags ← red_flags + ocorrencias(instancia.URL,"filter")
7:      red_flags ← red_flags + ocorrencias(instancia.URL,"busca")
8:      imagem ← ocorrencias(instancia.URL, "jpg")
9:      if yellow_flags > 0 & red_flags > 0 then
10:         flags ← yellow_flags + red_flags
11:         relevancia ← relevancia - (0.075 + 0.0125 * flags)
12:     else
13:         if red_flags > 0 then
14:             relevancia ← relevancia - (0.065 + 0.0200 * red_flags)
15:     if imagem > 0 then
16:         relevancia ← (relevancia - 0.09)
17:     relevancia ← max(relevancia, 0.0) return relevancia
```

CAPÍTULO 4

Avaliação

Nesta seção apresentamos avaliações sobre o detector de páginas e estratégias de localização.

4.1 Seleção dos modelos de classificação

4.1.1 Coleção de páginas

Para a construção da coleção de páginas sobre a qual foram treinados e testados os modelos de classificação rotulamos manualmente 2649 páginas de 92 sites distintos. Desse conjunto de páginas, apenas 1918 foram incluídas na base de dados final em decorrência de restrições de memória do Weka; o número de sites distintos manteve-se o mesmo.

A coleta dessas páginas deu-se majoritariamente de forma automática por meio de crawlers. A maior parte dos sites utilizados foi escolhida de maneira aleatória a partir de uma lista de páginas de índice geradas pelo coletor inter-site apresentado em [15]. Alguns dos sites dessa lista foram removidos como possíveis candidatos pela impossíbilidade de coleta de páginas de entidades através da simples navegação de seus links, sendo a utilização de JavaScript, exemplificada na Figura 4.1, nos websites a causa mais recorrente dessa dificuldade.

Dividimos a base de dados nos conjuntos de treinamento e de teste. Para construir o modelo utilizado pelo classificador de links designamos 70% dos documentos da base de dados para treinamento e os demais 30% para teste. Já para a construção do modelo utilizado pelo detector de entidades, utilizamos 60% dos documentos para o conjunto de treinamento e os demais 40% para teste. Todos os conjuntos foram bem balanceados em relação ao número de documentos positivos e negativos, sendo a maior diferença presente nos conjuntos entre esses dois tipos de apenas 0.8%. Destacamos que dividimos os documentos de modo que não existissem no conjunto de teste páginas de sites presentes no conjunto de treinamento.

4.1.2 Testes

Para avaliar o desempenho dos modelos utilizados pelo classificador de links e pelo detector de entidades em seus respectivos conjuntos de treinamento utilizamos as métricas de precisão, cobertura e F1.

A elaboração dos modelos foi feita com o auxílio da biblioteca Weka [16], uma coleção de algoritmos de aprendizagem de máquina para tarefas de mineração de dados, que, dentre outros recursos, possui diversos classificadores disponíveis.

Na Tabela 4.1 apresentamos os resultados dos testes feitos para a escolha do modelo a ser utilizado pelo classificador de links sobre um total de 2465 *features*. Notamos que, a

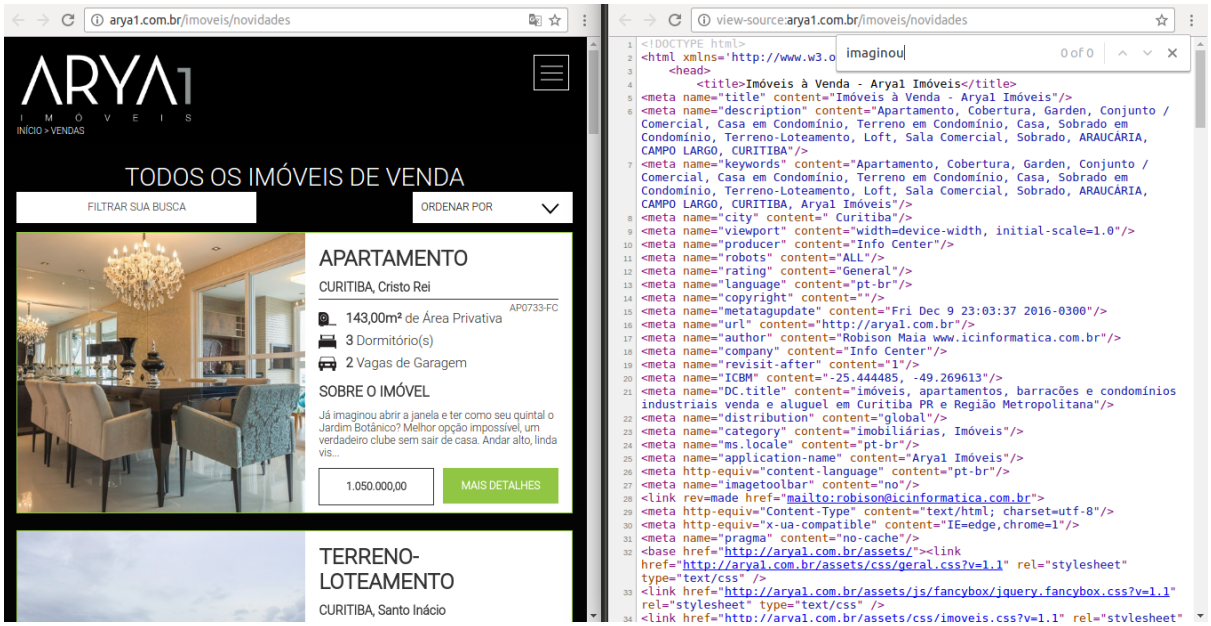


Figura 4.1: Exemplo de página de índice em que o acesso a páginas de entidade é feito por meio de `JavaScript`.

Versão	Precisão	Cobertura	F1
Naive Bayes	0.865	0.776	0.818
J48	0.887	0.759	0.818
Simple Logistic	0.892	0.738	0.808
Logistic	0.590	0.238	0.339

Tabela 4.1: Métricas para modelos de classificadores de URL testados. A classe `J48` gera árvores de decisão C4.5 [1], já `Simple Logistic` e `Logistic` são as classes utilizadas para a construção de modelos de regressão logística linear e multinomial, respectivamente.

menos do regressão logística multinomial, os modelos apresentaram resultados bons e bastante próximos. Os resultados obtidos validam nossa decisão, detalhada na Seção 3.1, de utilizar informações presentes na URL e títulos das páginas para o treinamento do classificador. A fim de evitar a não exploração de links positivos, atribuímos uma maior relevância à metrica de cobertura na escolha de que classificador utilizar. Decidimos utilizar o `Naive Bayes` porque consideramos sua precisão satisfatória e esse apresentou a maior cobertura dentre os classificadores testados.

Destacamos a obtenção de resultados contrários a nossa intuição quanto a importância das *features* extraídas das URLs em relação ao ganho de informação: inicialmente esperávamos uma grande importância dos atributos relacionados ao vocabulário do domínio de imóveis presentes nas URLs, porém, como podemos observar na Figura 4.2, todos os atributos relevantes em relação ao ganho de informação relacionados ao vocabulário foram extraídos dos títulos das páginas. Há, contudo, atributos de grande relevância relacionados à estrutura das URLs:

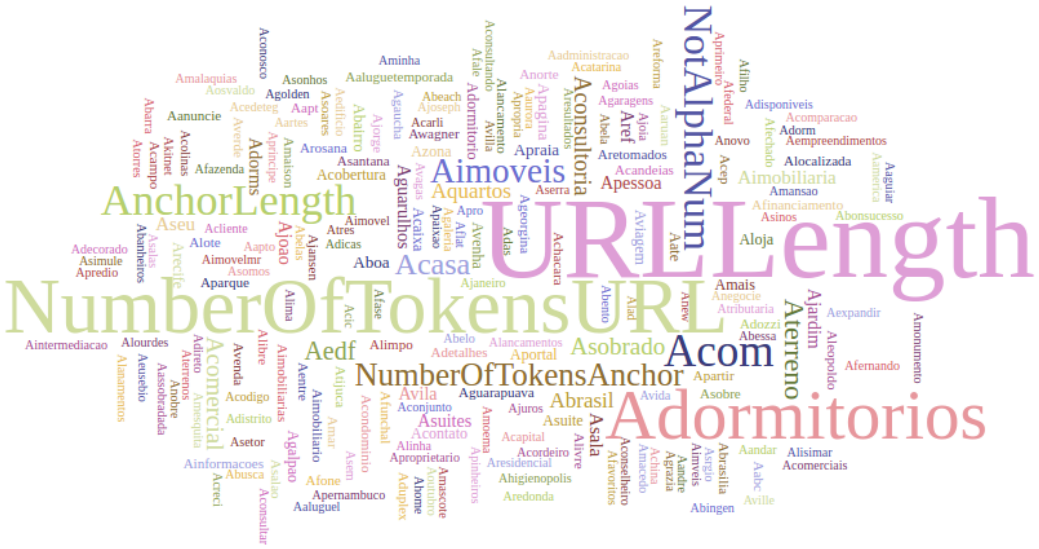


Figura 4.2: *Word cloud* dos atributos mais relevantes para o classificador de links de acordo com o ganho de informação. Atributos provenientes do título tem `A` como prefixo e atributos provenientes das URLs tem `U` como prefixo.

- **URLLength**: o tamanho das URLs;
- **NumberOfTokensURL**: o número de tokens extraídos da URL;
- **NotAlphaNum**: o número de caracteres não alfa-numéricos presentes na URL — a menos dos caracteres `-` e `/`.

Visto que os atributos relacionados ao vocabulário que possuem alguma relevância foram extraídos exclusivamente do título e que no momento de classificação de um link extraímos essas informações da âncora, é natural que seja questionado se os resultados apresentados nesta seção possuem significado real para a solução apresentada neste trabalho. Argumentamos que sim pela constante presença dos principais termos também nas âncoras, exemplificado na Figura 4.3 e apresentamos os resultados referentes ao classificador de links da próxima seção como evidência.

Na Tabela 4.2 apresentamos os resultados dos testes feitos para a escolha do modelo utilizado pelo detector de entidades. Diferentemente do classificador de links, desta vez priorizamos a precisão em detrimento da cobertura e, portanto, optamos por utilizar o modelo de `Regressão Logística Linear`. Através desses resultados podemos notar a importância da informação contida no corpo das páginas para a detecção de entidades imobiliárias.

Por acreditarmos que algumas informações estruturais como o número de links poderiam ser relevantes, mantivemos a extração desses atributos. Essas *features* todavia não foram selecionadas pelo classificador adotado, sendo utilizados exclusivamente atributos referentes ao vocabulário disponível na página como podemos observar na Figura 4.4. É interessante observar, porém, que ao rankearmos as *features* extraídas pelo ganho de informação, exibidas na

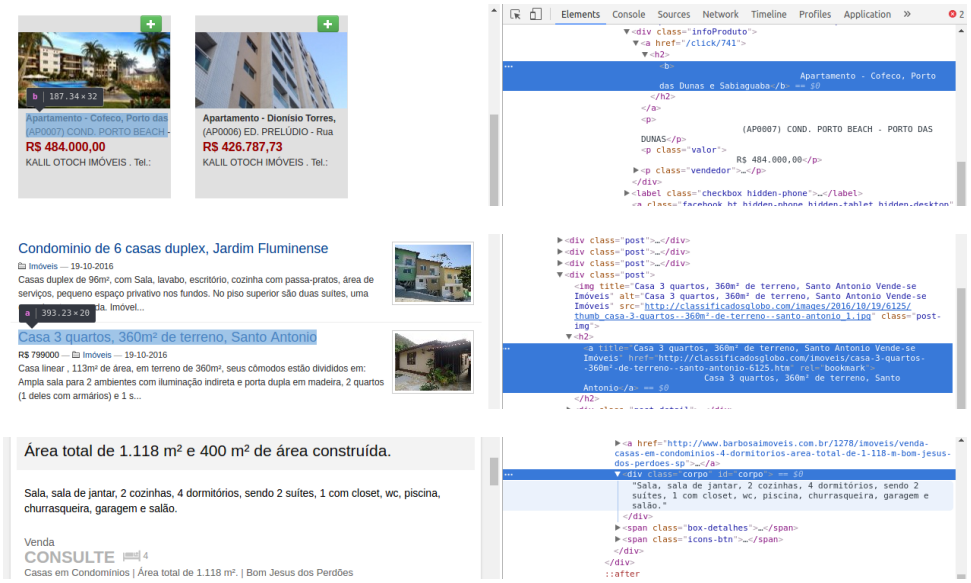


Figura 4.3: Exemplos de âncoras de links apontando para páginas de entidade com vocabulário relevante.

Versão	Precisão	Cobertura	F1
Simple Logistic	0.925	0.846	0.884
SMO	0.860	0.823	0.841
J48	0.871	0.871	0.871
Naive Bayes	0.908	0.786	0.842

Tabela 4.2: Métricas para modelos de classificadores candidatos para o detector de entidades. A classe `J48` gera árvores de decisão C4.5 [1], já `Simple Logistic` é uma classe utilizada para a construção de modelos de regressão logística linear e `SMO` é uma implementação do *Sequential Minimal Optimization* [2] utilizado para o treinamento de SVM.

Figura 4.4: *Word cloud* dos atributos mais relevantes para o classificador de entidades. Atributos provenientes do título tem `T` como prefixo, atributos provenientes das URLs tem `U` como prefixo e atributos provenientes do corpo tem `B` como prefixo.

Figura 4.5, notamos a presença dos seguintes atributos estruturais:

- **NumberImages:** Número de imagens presentes na página;
- **NumberLinks:** Número de links do mesmo domínio presentes na página;
- **NumberR$:** Número de símbolos de Real presentes na página.

4.2 Avaliação em Coletas de Sites

Para a avaliação da solução proposta, realizamos a coleta de `50000` sites a partir de `60` páginas de índice de imóveis utilizadas como *seeds*. Comparamos o desempenho de nossa solução com uma implementação baseline que atribui relevâncias aleatórias para os links ainda não explorados. As *seeds* foram novamente escolhidas de maneira aleatória a partir de uma lista de páginas de índice geradas pelo coletor apresentado em [15].

Inicialmente, o coletor intra-site permitia que fossem adicionados à fronteira links extraídos de páginas quaisquer independentemente de suas relevâncias. Chamaremos essa configuração de *soft focus*. Com a utilização dessa configuração nossa implementação apresentava um desempenho aquém do esperado: exibia um *harvest ratio* muito bom nos primeiros momentos, mas sua performance declinava rapidamente. Acreditando que o classificador de links pudesse estar impedindo o coletor de navegar por páginas de índice, testamos adicionar aleatoriedade à relevância final gerada pelo classificador. Conforme aumentamos a aleatoriedade adicionada, observamos uma melhora no desempenho do coletor. Essa melhora, contudo, apenas atrasava discretamente o momento de convergência dos *harvest ratios* da implementação baseline e da

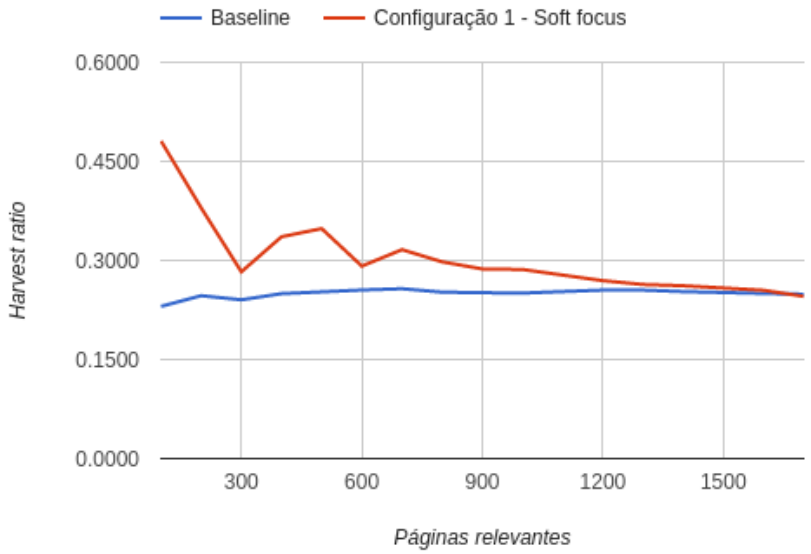
Figura 4.5: *Word cloud* dos atributos mais relevantes de acordo com o ganho de informação. Atributos provenientes do título tem `T` como prefixo, atributos provenientes das URLs tem `U` como prefixo e atributos provenientes do corpo tem `B` como prefixo.

nossa. Na figura Figura 4.6 comparamos nossa implementação que obteve o melhor desempenho ainda com *soft focus* e a implementação baseline. Nessa adicionamos um valor entre 0.1 e 0.4 à relevância de um link ainda não explorado. Apresentamos os dados até o momento de convergência das duas implementações. Destacamos como, além de convergir rápidamente ao *harvest ratio* da solução baseline, nossa solução apresentava uma maior tendência de piora, quando utilizando *soft focus*.

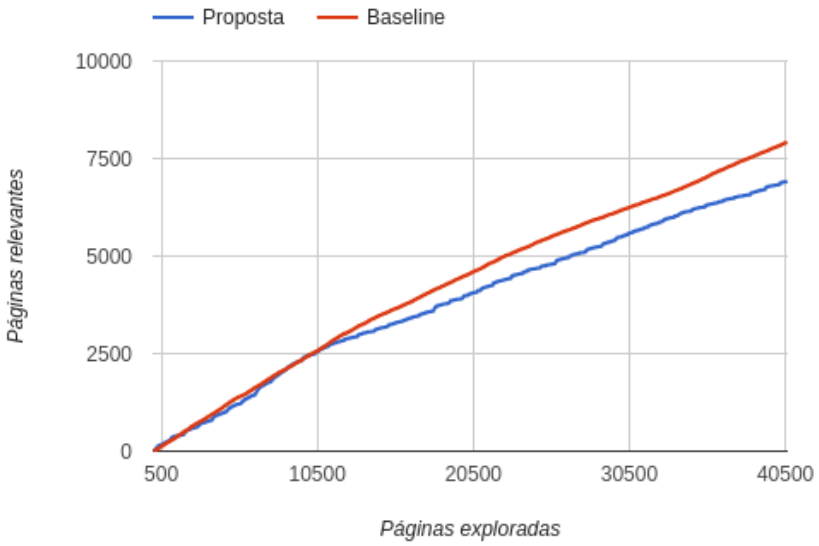
Analisando possíveis causas para o mau desempenho de nossa proposta, observamos que o coletor era gravemente afetado por *crawler traps*: um conjunto de páginas que, intencionalmente ou não, pode ser utilizado para que *crawlers* façam um número infinito de requisições. Em particular o site `http://classificados.diariodonordeste.com.br/imoveis/` mostrou-se bastante problemático por gerar dinamicamente endereços para páginas de índice, mesmo que as novas URLs não recuperassem imóvel algum; nesses casos, a página carregava todos os imóveis disponíveis, o que, em termos práticos, fazia com que o coletor visitasse a mesma página inúmeras vezes.

A estratégia de *soft focus* era a adotada até então porque com *hard focus*, isto é, adicionando apenas links provenientes de páginas relevantes à fronteira, seria impossível coletar qualquer página de entidade já que as *seeds* utilizadas são páginas de índice de entidades — páginas com uma relevância inferior às de entidade — e, portanto, os link presentes nelas jamais seriam adicionados à fronteira.

A fim de contornar esses dois problemas, o mau desempenho de nossa implementação com *soft focus* e impossibilidade de utilizar *hard focus*, adotamos uma configuração flexível que chamaremos de *changing focus*. Nessa configuração, o coletor passa `3/4` do tempo com *hard focus* para que priorize a coleta links extraídos de páginas relevantes e `1/4` com *soft focus* para que não deixe de adicionar à fronteira links presentes em páginas de índice.



(a) Comparação de harvest ratios iniciais



(b) Comparação de número de páginas coletadas

Figura 4.6: Comparação dos harvest ratios iniciais e do total de páginas coletadas. Notamos na primeira figura que o desempenho da proposta é bastante elevado inicialmente, porém rapidamente piora.

Com a adoção do *changing focus* à nossa solução final, obtivemos uma melhora expressiva. Na Figura 4.7 apresentamos os *harvest ratios* da implementação baseline e da nossa solução. Podemos correlacionar essa melhora com o aumento da robustez contra *crawler traps*: já que essas estavam presentes principalmente em páginas de índice, ao restringirmos parcialmente a coleta de novos links para apenas links extraídos de páginas classificadas como de entidades imobiliárias, evitamos entrar em *loops* de requisições infinitas a páginas de índice geradas dinamicamente.

Ainda motivados a ter uma maior compreensão quanto ao desempenho do sistema como um todo, avaliamos sua performance em três sites específicos:

- **http://www.gilbertopinheiroimoveis.com.br/**
- **http://www.leonardolobo.com.br/alugar-imoveis**
- **http://www.grandesareassc.com.br/**

Escolhemos esses sites por ilustrarem dois tipos distintos de problemas enfrentados pelo coletor e um cenário ideal.

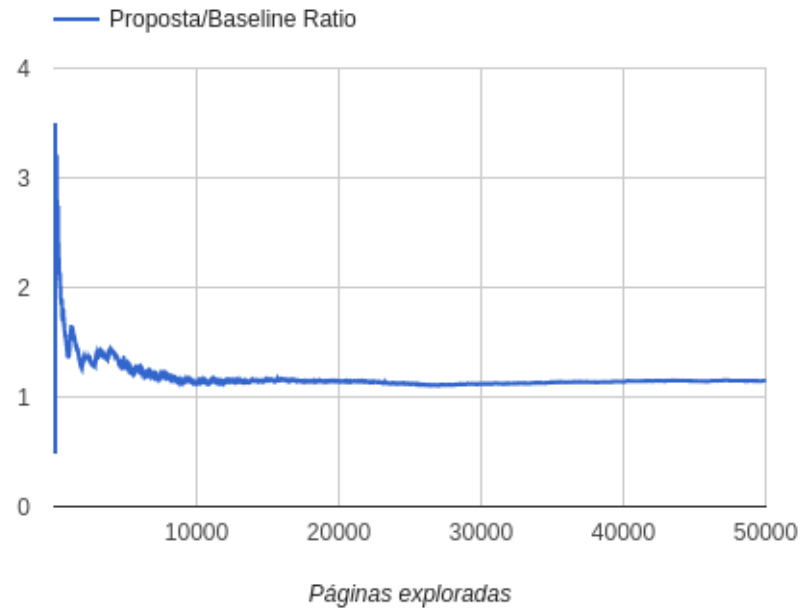
No primeiro site temos um exemplo de situação ideal: páginas de entidade bem estruturadas, com títulos e URLs descritivos e totalmente acessíveis a partir de um índice de busca que permite a não definição de filtros. É importante notar que o *harvest ratio* apresentado pela solução proposta bem como pelo baseline nos indica que esse é um cenário comum e não uma exceção.

O segundo site, como podemos ver na Figura 4.9 possui páginas bem estruturadas, URLs e títulos descritivos. Enquanto o detector de entidades não tem problemas classificando páginas desse site, ele está limitado às páginas acessíveis a partir da página de índice fornecida como ponto de entrada. Esta limitação se dá porque, em consultas cujos resultados ocupam múltiplas páginas, a mudança de página se dá por meio de `JavaScript` fazendo com que os imóveis que não estejam na primeira página fiquem inacessíveis ao coletor.

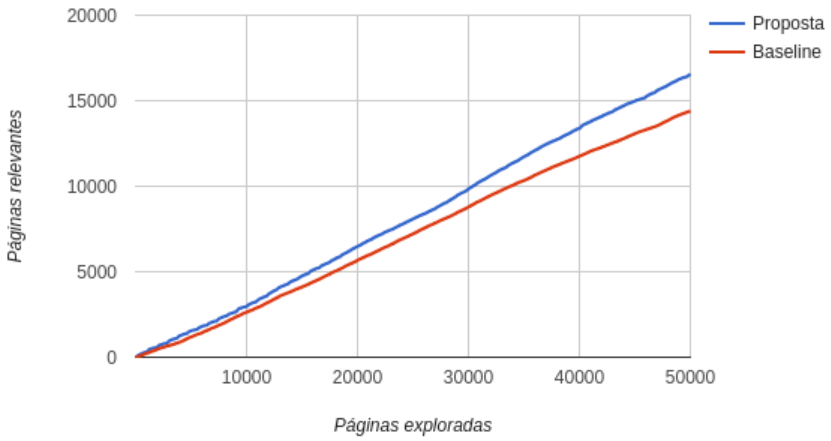
Já o terceiro site apresenta um desafio semelhante ao observado na primeira implementação do detector de links: *crawler traps*. Desta vez, porém, observamos as URLs criadas dinâmicamente em páginas de entidade como, por exemplo, `http://www.grandesareassc.com.br/apartamento-com-detalhes-diferenciados-no-centro-de-camboriu/3-dormitorios-no-calcadao-em-balneario-camboriu-quadra-do-mar-1/contato/contato/contato/solicite-seu-imovel/?&/`. Esses links não são evitadas simplesmente pela adição do mecanismo de *changing focus* porque são extraídos de páginas relevantes e ainda apontam páginas relevantes. Este problema é bastante relevante não só por gerar resultados duplicados, mas também por ocupar recursos que poderiam ser utilizados para recuperar entidades de sites menos explorados.

Desempenho do Detector de Entidades.

A principal métrica pela qual avaliamos o detector de entidades foi a precisão, já que avaliar a cobertura de maneira abrangente é inviável pelo grande número de páginas coletadas. Para avaliar a precisão do detector selecionamos aleatoriamente `500` das páginas classificadas como páginas de entidades imobiliárias. Nessa amostra identificamos apenas `26` páginas falso-positivas, resultando em uma precisão de `94.8%`.



(a) Razão entre o *harvest ratio* da proposta e o *harvest ratio* da implementação baseline



(b) Comparação de número de páginas coletadas

Figura 4.7: Comparação dos *harvest ratios* e do total de páginas coletadas. Podemos notar um impacto bastante positivo da adoção do *changing focus*, especialmente na implementação proposta.

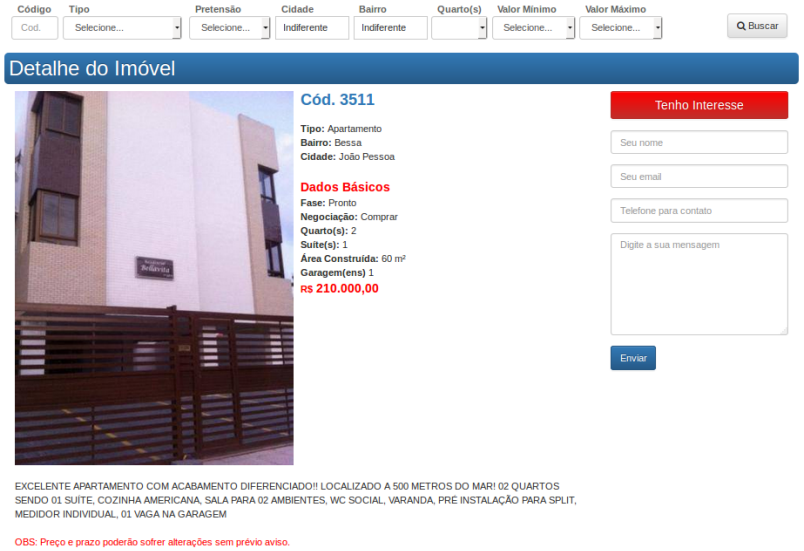


Figura 4.8: Exemplo de falha do detector de entidades: página descritiva e bem estruturada, porém com URL e título pouco descritivos.

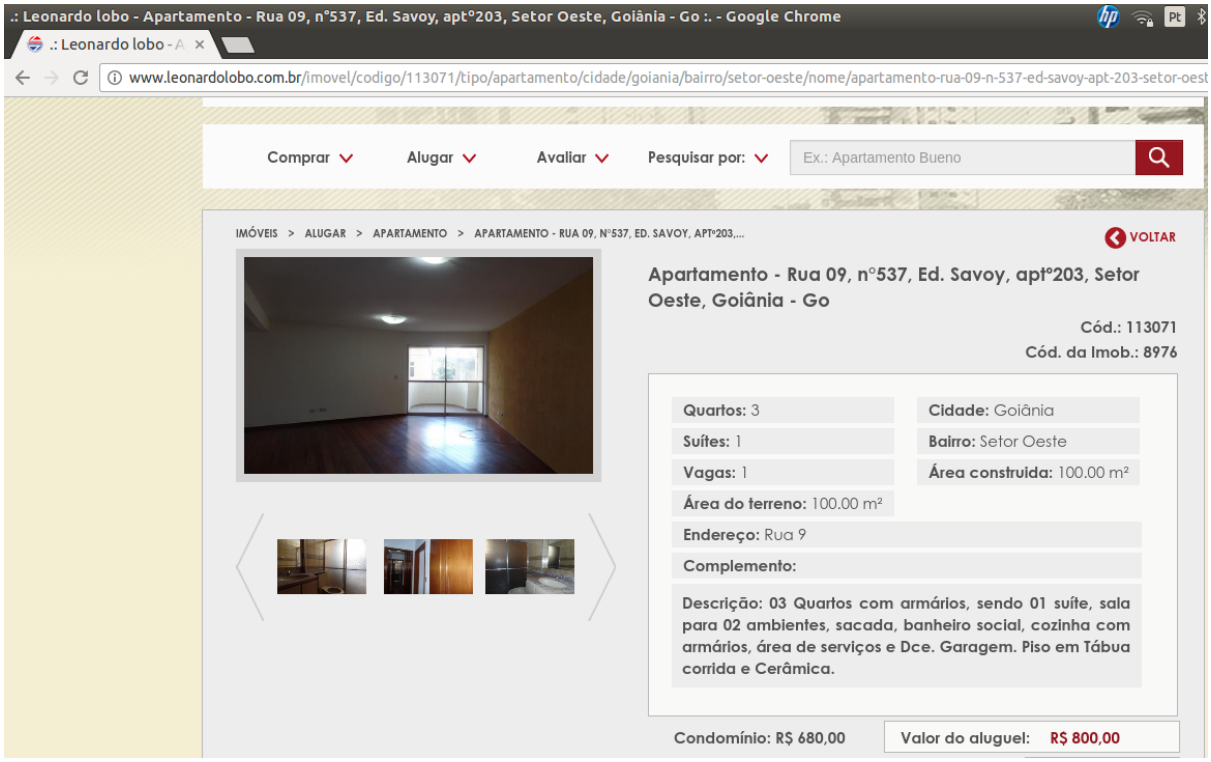


Figura 4.9: Exemplo de página do site leonardolobo.com.br.

A fim de ter uma maior compreensão quanto ao desempenho do detector de entidades, procuramos encontrar algum site em que o detector estivesse apresentando uma taxa de acerto inferior à esperada. Escolhemos, então, analisar `http://www.rrimobiliaria.com.br/` porque notamos que apenas uma pequena parte de suas páginas de entidades estava sendo classificada como relevante e suas páginas de entidade aparentavam ser bastante descritivas e, até certo ponto, bem estruturadas, como podemos ver na Figura 4.8. Notamos que as páginas de entidade nesse site não possuem URLs nem títulos descritivos: `http://www.rrimobiliaria.com.br/detalhe-imovel.php?id=3511&t=1&vini=210.000,00&vfin=210.000,00` é a URL referente à página exibida na Figura 4.8 e `RR Imobiliária - Imóveis em João Pessoa - PB` seu título. Procuramos outro site em que esse mesmo padrão de páginas descritivas e URLs genéricas fosse repetido a fim de verificar se o detector falharia novamente: analisamos as páginas do site `https://www.imocasa.com.br/`. Observamos que as únicas páginas classificadas como positivas são as que contém o termo `sobrado` no título; um forte indicativo de relevância como observado na Figura 4.4. Demais páginas de imóveis não obtiveram score alto o suficiente para serem classificadas como páginas de entidade. A análise desses dois sites deflagra a grande dependência do detector em relação aos tokens presentes especialmente na URL das páginas e nos fornece indícios de um possível *overfitting* nos dados de treinamento. É possível que uma forma de remediar essa dependência seja com a diminuição do número de features relacionadas ao vocabulário das páginas e com uma adição de features relacionadas à estrutura não somente das página de entidades como também das páginas de que sejam extraídos os links que as apontam.

CAPÍTULO 5

Conclusão

Neste trabalho apresentamos uma solução eficaz para o problema de localização e identificação de entidades imobiliárias em um mesmo website. Para tanto apresentamos estratégias que utilizam dois classificadores: um utilizado para a localização de páginas de entidade do domínio imobiliário em um mesmo site e outro para a detecção dessas páginas de entidade. Utilizando o classificador de links alcançamos um *harvest ratio* de `33.08%`, totalizando uma melhora de 14% em relação ao baseline, com uma precisão de `94.8%` do detector de entidades.

Identificamos ainda algumas problemáticas que permitem uma clara visão dos passos a serem tomados em trabalhos futoros: para contornar a dependência do detector de entidades quanto a informações presentes nas URLs podemos empregar técnicas de visualização a fim de ter uma melhor compreensão dos atributos que caracterizam páginas de entidades, e adotar *online learning* para a construção do classificador de links pode ser uma boa estratégia para precavermo-nos das falhas observadas no conjunto de treinamento assim como expandirmos a capacidade de generalização do classificador.

Referências Bibliográficas

[1] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[2] J. C. Platt, "Advances in kernel methods," ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208, Cambridge, MA, USA: MIT Press, 1999.

[3] D. Qiu, L. Barbosa, X. L. Dong, Y. Shen, and D. Srivastava, "Dexter: Large-scale discovery and extraction of product specifications on the web," *Proc. VLDB Endow.*, vol. 8, pp. 2194–2205, Sept. 2015.

[4] F. Alahmari and L. Magee, "Linked data and entity search: A brief history and some ways ahead," in *3rd Australasian Web Conference (AWC 2015)* (J. G. Davis and A. Bozzon, eds.), vol. 166 of *CRPIT*, (Sydney, Australia), pp. 29–38, ACS, 2015.

[5] J. Pound, P. Mika, and H. Zaragoza, "Ad-hoc object retrieval in the web of data," in *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, (New York, NY, USA), pp. 771–780, ACM, 2010.

[6] M. de Kunder, "Worldwidewebsize.com | the size of the world wide web (the internet)," 2016. [Online; accessed 13-September-2016].

[7] M. de Kunder, "Geschatte Grootte van het Geïndexeerde World Wide Web," Master's thesis, Tilburg University, the Netherlands, 2006.

[8] N. Dalvi, A. Machanavajjhala, and B. Pang, "An analysis of structured data on the web," *Proc. VLDB Endow.*, vol. 5, pp. 680–691, Mar. 2012.

[9] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. The MIT Press, 2012.

[10] S. Chakrabarti, M. van den Berg, and B. Dom, "Focused crawling: A new approach to topic-specific web resource discovery," *Comput. Netw.*, vol. 31, pp. 1623–1640, May 1999.

[11] L. Barbosa and J. Freire, "Searching for hidden-web databases," in Doan *et al.* [17], pp. 1–6.

[12] L. Barbosa, "Harvesting forum pages from seed sites," 2016.

[13] S. Chakrabarti, K. Punera, and M. Subramanyam, "Accelerated focused crawling through online relevance feedback," in *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, (New York, NY, USA), pp. 148–159, ACM, 2002.

[14] D. Qiu, L. Barbosa, X. L. Dong, Y. Shen, and D. Srivastava, "DEXTER: large-scale discovery and extraction of product specifications on the web," *PVLDB*, vol. 8, no. 13, pp. 2194–2205, 2015.

[15] D. Caraciolo, "Detectando Sites de Entidades em Larga Escala: o Caso de Sites de Imóveis," 2016.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, Nov. 2009.

[17] A. Doan, F. Neven, R. McCann, and G. J. Bex, eds., *Proceedings of the Eight International Workshop on the Web & Databases (WebDB 2005), Baltimore, Maryland, USA, Collocated mith ACM SIGMOD/PODS 2005, June 16-17, 2005*, 2005.

[18] S. Chakrabarti, M. van den Berg, and B. Dom, "Focused crawling: A new approach to topic-specific web resource discovery," in *Proceedings of the Eighth International Conference on World Wide Web*, WWW '99, (New York, NY, USA), pp. 1623–1640, Elsevier North-Holland, Inc., 1999.

Este volume foi tipografado em LaTeX na classe UFPEThesis (www.cin.ufpe.br/~paguso/ufpethesis).