5 Winetag.com.br: um estudo de caso

Foi realizado um estudo de caso instanciando o framework proposto. O domínio escolhido foi o da comercialização de vinhos, que apresenta ângulos e desafios muito interessantes para os problemas da busca de informação em bases da Deep Web e de detecção de duplicatas. O objetivo dessa aplicação foi utilizar a nossa abordagem para enriquecer uma base de dados pré-existente.

O primeiro passo do processo foi estudar quais os atributos deste tipo de entidade seriam suficientes para identificá-los. Assumimos que um vinho típico é produzido por uma vinícola e apresenta um nome. O nome pode ser composto de um substantivo próprio, assim como elementos que ajudam a caracterizar o vinho. Estes elementos podem referenciar regiões produtoras (típico em vinhos franceses como “Rothschild Médoc Rèserve Spéciale”, da região de Médoc), podem ter o nome de uma classificação oficial (“DOCG” em “Batasiolo Barbaresco DOCG”), ou podem conter nomes próprios da vinícola onde foram produzidos (“Reservado” em “Concha Y Toro Reservado Cabernet Sauvignon”).

Em muitas ocasiões, vinícolas produzem o mesmo tipo de vinho, i.e. vinhos produzidos utilizando-se uma mesma metodologia, a partir de diferentes uvas. É comum os rótulos dos vinhos serem diferenciados pelo nome de sua uva. Exemplos são “Cobos Felino Chardonnay” e “Cobos Felino Cabernet Sauvignon”. O mesmo padrão é observado quando o produtor deseja destacar que utilizou totalmente, ou em grande parte, apenas um tipo de uva para produção de um vinho.

Neste estudo de caso estamos lidando exclusivamente com vinhos finos, desta forma é necessário considerar também a safra, pois além do vinho apresentar diferenças características de acordo com o ano da colheita das uvas, o mercado também comercializa os vinhos de acordo com a qualidade obtida em cada safra.

Informações adicionais, tais como o volume da garrafa e percentual alcoólico, normalmente estão presentes nos rótulos, mas não são relevantes para a diferenciação de vinhos, pois são informações que repetem-se freqüentemente nestas instâncias e, portanto, não têm utilidade no processo de diferenciação. O preço ainda pode ser útil para identificar versões tais como meia-garrafa, garrafa tradicional de 750ml e outras como *Magnum*.

Desta forma, para identificação de vinhos, no cenário web utilizamos os seguintes três atributos, do tipo cadeia de caracteres (string):

 (vinícola, nome, safra)

Com este modelo, foi iniciada a instanciação de nosso framework. O passo seguinte foi identificar as fontes de dados: sites de e-commerce de vinhos. Os candidatos foram escolhidos levando em consideração a nacionalidade e o tamanho da coleção de produtos disponíveis para coleta. Neste estudo de caso levamos em conta apenas sites brasileiros por dois motivos: o primeiro que a aplicação que terá informações enriquecidas é voltada para o mercado nacional e o segundo é que seriam inseridas outras dificuldades neste trabalho, desnecessariamente, relacionadas ao idioma.

5.1 Metodologia de trabalho

5.1.1 Identificação das fontes de dados

Nesta etapa foi feito um levantamento sobre possíveis sites candidatos a fontes de pesquisa entre os sites de e-commerce de vinhos disponíveis. Os candidatos foram escolhidos levando em consideração a nacionalidade (neste caso consideramos apenas sites localizados no Brasil), e o tamanho da coleção de produtos disponíveis para coleta. Também foi observada a viabilidade de se conseguir obter informações através de URL's. Foram escolhidas as lojas Expand[3], Grand Cru[4], Mistral[5], Vinci[6], Wine.com.br[7], Zahil[8] e Estação do Vinho[9].

5.1.2 Alinhamento de Esquemas - Schema Matching

Nesta etapa foi realizado o mapeamento entre modelo de objeto o qual deseja-se enriquecer conteúdo e objetos retornados nas pesquisas nas fontes. Foram identificadas as entidades relacionadas aos produtos disponíveis do site e qual o relacionamento delas com o modelo proposto pelo site do projeto.

3 http://www.expand.com.br
4 http://www.grandcru.com.br
5 http://www.mistral.com.br
6 http://www.vincivinhos.com.br
7 http://www.wine.com.br
8 http://www.zahil.com.br
9 http://www.estacaodovinho.com.br

Para o mapeamento correto verificou-se ainda a necessidade de criar mediadores para converter dados. Verificou-se por exemplo que cada loja apresentava uma classificação própria de categoria do vinho. Para resolver isso foram criadas tabelas de conversão.

É preciso notar algumas peculiaridades relacionadas às uvas para conversão. Existem nomes de uva que referenciam a mesma entidade. Essas variações foram criadas em grande parte devido a questões culturais. Existe, por exemplo, a uva Tempranillo que também é chamada de Tinto Fino, Cencibel ou Aragonês.

5.1.3 Descoberta de padrões em URL's e montagem de consultas

Nesta etapa foram estudadas quais as regras de montagem de URL deveriam ser seguidas para executar as consultas nas fontes. Códigos referentes a categorias ou departamentos de loja e tamanho de nomes necessários para busca foram estudados.

A eficácia do framework é altamente dependente dos resultados obtidos por estas consultas, deste modo, é preciso encontrar meios de produzir consultas que, idealmente, retornem apenas a entidade procurada. Se a consulta for muito genérica será preciso confrontar o modelo com uma quantidade muito grande de possíveis duplicatas. Por outro lado, se muito específica, pode nunca retornar a referência para e entidade procurada.

É comum encontrar formulários de busca de sites que executam consultas diretamente sobre um banco de dados utilizando expressões do tipo "like" no campo título da tabela correspondente. Este tipo de implementação impede que façamos consultas utilizando o nome completo, pois raramente o nome é exatamente igual, (ou uma sub cadeia exata) do nome armazenado na fonte utilizada.

A estratégia utilizada nesta instância foi de executar diversas consultas, explorando várias formas do modelo procurado. Para cada nome de vinho, foi enviada uma consulta por token presente no nome. Para evitar a realização de consultas genéricas demais, tokens com termos comuns (*stopwords*), tokens com nomes de uvas, palavras de classificação como "Reserva", ou nomes de regiões produtoras foram suprimidas.

Desta forma esperamos garantir que ao menos uma das consultas (aquela com um termo mais significativo para o vinho) retorne o vinho desejado nas

primeiras posições. Por outro lado, com esta estratégia é possível que um mesmo vinho retorne no resultado de diversas consultas, ou que o vinho procurado nunca apareça (nos casos onde o nome do vinho é composto por uma combinação de termos muito comum) .

5.1.4 Criação de parsers das fontes de dados

Todas as fontes disponibilizaram seus dados apenas no formato HTML, com alguns metadados utilizados para aplicação do layout. Foi utilizada a técnica de Screen Scraping, referenciada no Capítulo 2, para a captura da informação destas páginas.

Como o framework não contempla nenhum mecanismo de Identificação automática de entidades nas páginas retornadas pelas pesquisas, foi preciso identificar os atributos manualmente através das *tags* presentes no código. Para encontrar as marcações corretas foram utilizados aplicativos como o *Developer Tools do Internet Explorer*[10] *e Firebug*[11], que oferecem ferramentas de identificação visual de elementos do código. Para cada página de interesse, foi preciso descobrir a estrutura de organização dos dados no código.

10 http://msdn.microsoft.com/en-us/library/dd565628%28VS.85%29.aspx
11 https://addons.mozilla.org/pt-BR/firefox/addon/1843/

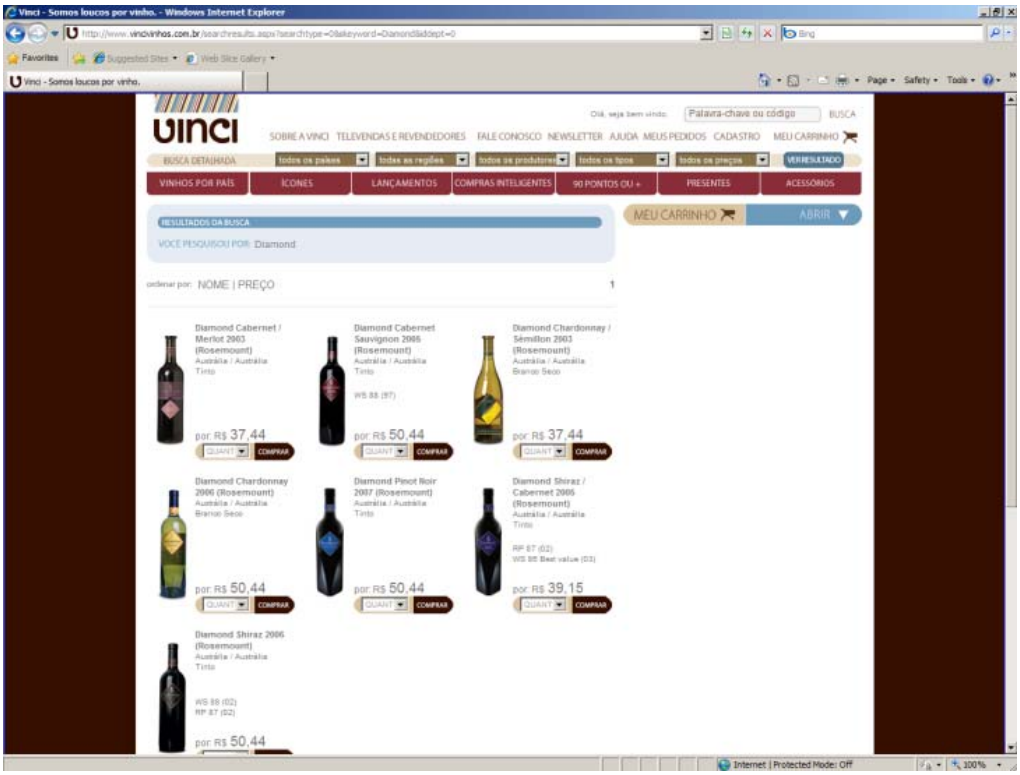

Figura 17 - Exemplo de fonte de dados

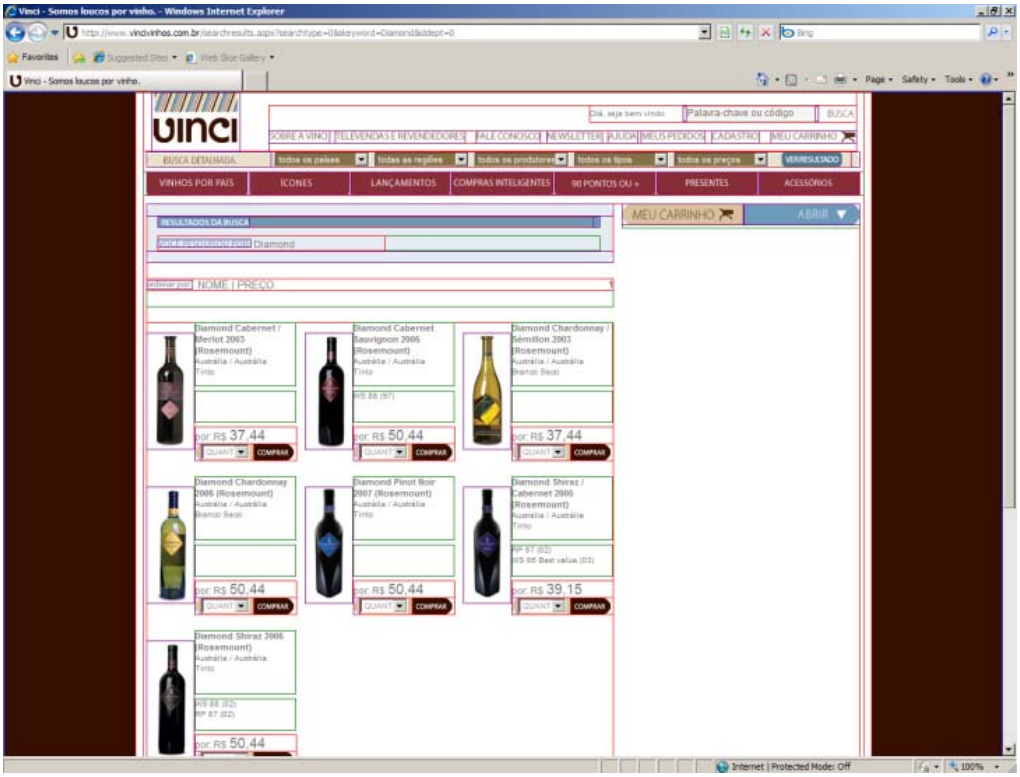

Figura 18 - Exemplo elementos identificados pelo Developer Tools

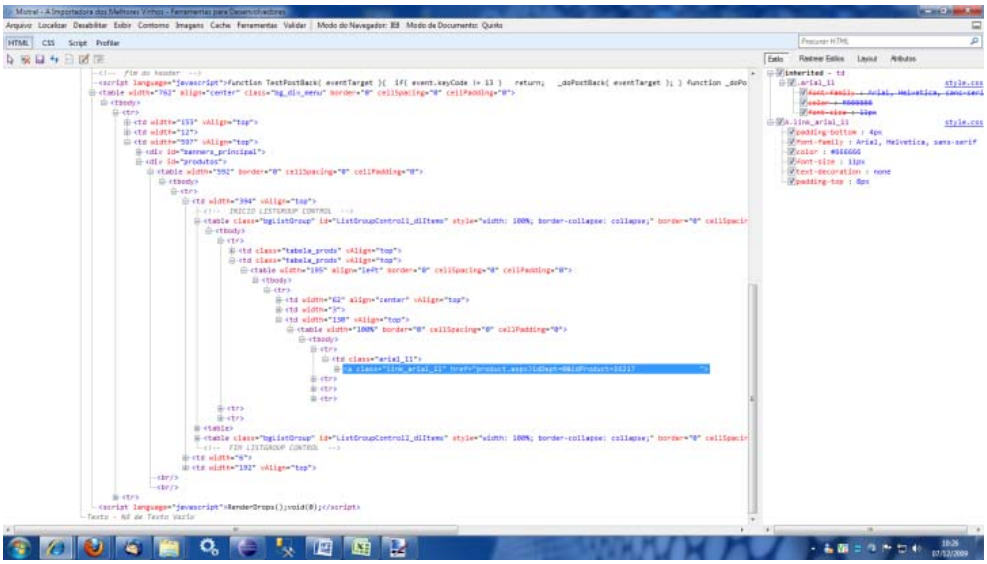

Figura 19 - Elementos destacados no código-fonte

Para cada página de busca, e de informações detalhadas de vinhos, foram codificados parsers especificos. O código foi escrito visando oferecer alguma flexibilidade às mudanças de layout que pudessem vir a ocorrer nas páginas de pesquisa. A codificação é semelhante a criação de um parser de arquivos XML. Com uma API adequada para extração de informações desse tipo de arquivo como Jericho[12], foram necessários em média 30 minutos e 25 linhas de código para codificação do *parser* de cada página.

Para evitar problemas durante o processo de captura de informação, foram criados testes automatizados de extração de entidades. Assim, antes da execução do aplicativo era possível verificar se os *parsers* de cada página estavam adequados (corretos).

5.1.5 Escolha de funções de cálculo similaridade

Foram estudadas funções de cálculo de similaridade capazes de atender aos requisitos deste projeto. Durante esta etapa, realizamos uma série de testes de modo a medir a confiabilidade das medidas de cálculo de similaridade escolhidas para o domínio dos vinhos, em particular. Para um conjunto base de instâncias de vinho foram calculados os graus de similaridade utilizando 15 funções baseadas em caracteres, *tokens* e híbridas.

A partir dos resultados foram determinadas as funções que seriam mais adequadas para este domínio. Foram escolhidas as funções capazes de gerar valores de similaridade altos, para itens considerados duplicatas, e valores bai-

12 http://jericho.htmlparser.net/docs/index.html

xos para itens considerados não duplicatas. Verificou-se que as funções Dice, Cosine, Block e TFIDF, baseadas em *tokens*, foram as mais adequadas para nosso problema.

Para evitar que erros de ortografia interferissem nos resultados, todas as cadeia de caracteres utilizadas para comparação, de modelos e possíveis duplicatas, foram filtradas. O primeiro filtro consistiu em transformar todos os caracteres em minúsculos.

O segundo filtro consistiu em transformar alguns caracteres no equivalente de uma tabela. Caracteres acentuados, por exemplo, foram transformados em caracteres simples. Caracteres especiais como aspas foram substituídos por espaços. Esta é a mesma idéia apresentada no trabalho de Davis [14].

5.1.6 Codificação do Classificador

Nesta etapa, já com as funções de cálculo de similaridade definidas, codificou-se o classificador. No framework é preciso sobrescrever um método para treinar o classificador, diferenciando casos positivos e negativos. Para ser aceito como caso exemplo, definiu-se que os vinhos que apresentassem similaridade maior que um determinado limiar, a ser escolhido empiricamente, nas funções Cosine, Dice, Block ou TFIDF seriam aceitos como possíveis casos positivos, caso contrário seriam aceitos como casos negativos.

O classificador recebe como entrada sempre uma cadeia de caracteres representando a entidade do vinho (nome do produtor e do vinho) e retorna um valor referente ao resultado da classificação. Se o valor é maior que o valor de corte (*cutoff)* do classificador, definido empiricamente, podemos afirmar que o classificador considera o item como duplicata.

Como não incluímos a safra no processo de classificação, mesmo que classificado como positivo pelo classificador, o vinho só pode ser considerado duplicata se os valores das safras forem iguais. Não utilizar a safra no treinamento do classificador é importante pois o valor da safra, com quatro dígitos, deve ser comparado de forma exata. Se incluíssemos esta informação no classificador, estaríamos incluindo ruído.

Nos testes realizados verificou-se que alguns vinhos do porto, com diferentes anos de envelhecimento no nome não foram identificados corretamente. O classificador não foi capaz de diferenciar os vinhos apenas por causa deste *token* numérico.

Se um vinho apresenta a safra e outro não, com o mesmo nome, não podemos afirmar nada, logo consideramos como caso negativo.

5.1.7 Fusão de dados

Após a identificação de duplicatas é preciso realizar a operação de *merge* dos dados. Dados provenientes de diferentes fontes podem estar em diferentes formatos ou conter informações incompletas e/ou contraditórias. Como o objetivo principal da aplicação dos *Mashups* era apenas coletar dados de preços e imagens dos vinhos, optou-se pela estratégia mais simples. A primeira informação não existente na base é persistida.

Para fins ilustrativos, temos algumas das classes representadas no diagrama da figura 20:

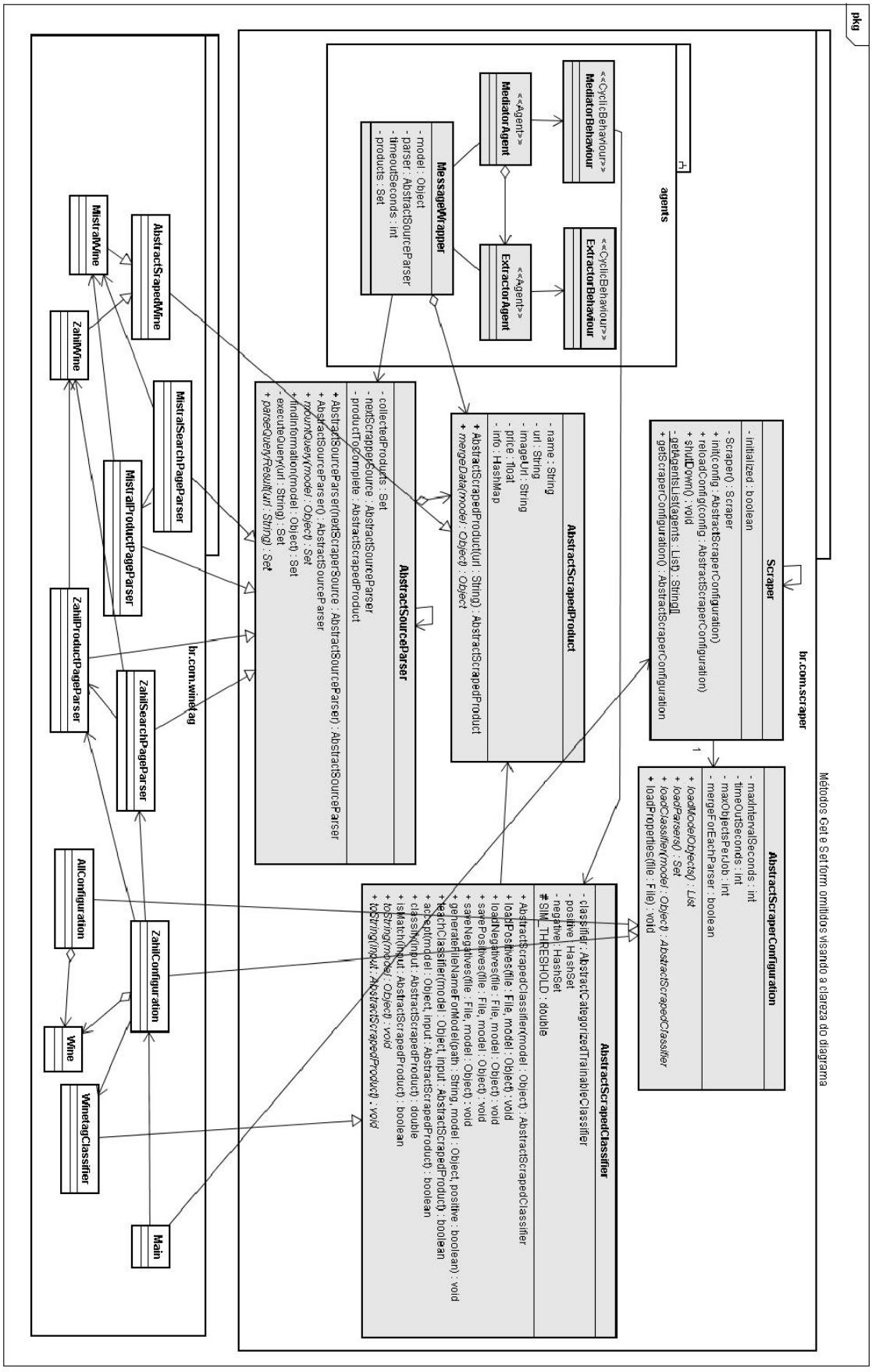

Figura 20 – Diagrama de classes para instância de vinhos

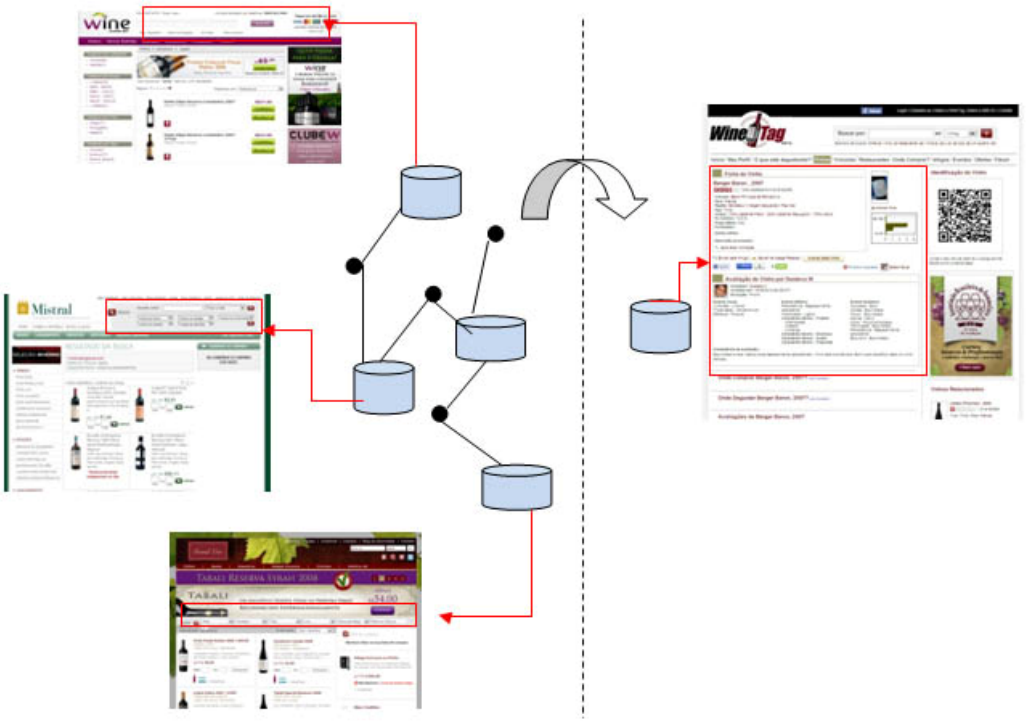

Figura 21 - Winetag.com.br: mashup com dados da Deep Web

5.2 Resultados

Para validar a solução proposta ao problema dos vinhos temos que nos apoiar em algumas métricas comuns de estatística para classificação. Estas métricas são baseadas nas quantidades de itens classificados corretamente como “verdadeiramente positivo” (TP) e “verdadeiramente negativo” (TN), além dos itens classificados incorretamente como “incorretamente positivo” (FP) e “incorretamente negativo” (FP). Temos as definições:

Precisão (*Precision*): É a probabilidade de que um item, aleatoriamente escolhido, seja relevante.

$$Precision = \frac{Tp}{Tp + Fp}$$

Recall: É a probabilidade de que um item, aleatoriamente escolhido, seja recuperado em uma busca.

$$Recall = \frac{Tp}{Tp + Fn}$$

Acurácia (*Accuracy*): Reflete o grau de certeza de que uma medida é reprodutível.

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn}$$

F-Measure: É uma média harmônica que combina precisão e recall. É útil para medir a qualidade da classificação.

$$F = 2 \times \frac{precision \times recall}{precision + recall}$$

Foram realizadas consultas a duas fontes de dados em Fevereiro de 2010. Utilizamos uma amostra de 1.402 vinhos com nomes únicos, de safra maior que 2004. A instância do framework retornou 21.302 duplicatas, ou seja, vinhos que poderiam ser classificados como similares.

Para a realização do enriquecimento da base de dados foram necessárias 48h. Para reduzir o tempo de escolha dos parâmetros de execução (*threshold* e *cutoff*), foram executados os testes com os dados capturados na primeira execução.

Deduzimos que temos ao menos 142 duplicatas corretas. Este valor foi estimado observando o maior número de casos considerados verdadeiramente positivos, menos a quantidade de falsos positivos encontrados nos testes.

Visando obter bons resultados e explorar o funcionamento do framework para este domínio, foram realizados experimentos variando os parâmetros “threshold” da função de similaridade e o “cutoff” do classificador. Estes são os principais parâmetros envolvidos na solução do problema de “Entity Matching” proposta por este framework.

Temos a tabela de itens considerados relevantes, ou seja, casos positivos e respectivamente a porcentagem de operações de fusão de dados em relação a amostra:

		Cutoff					
Threshold	P	1.00	0.95	0.90	0.85	0.80	0.75
	1.00	29	29	32	66	85	125
	0.95	49	49	58	105	162	212
	0.90	79	79	85	128	187	236

Tabela 1 - Quantidade de classificações positivas (P)

		Cutoff					
Threshold	Merge	1.00	0.95	0.90	0.85	0.80	0.75
	1.00	2,07%	2,07%	2,28%	4,71%	6,06%	8,92%
	0.95	3,50%	3,50%	4,14%	7,49%	11,55%	15,12%
	0.90	5,63%	5,63%	6,06%	9,13%	13,34%	16,83%

Tabela 2 – Percentual de operações realizadas de merge em relação à amostra

Seguem as tabelas com os valores encontrados de TN, FP, TN, FN, verificados manualmente:

		Cutoff					
Threshold	TP	1.00	0.95	0.90	0.85	0.80	0.75
	1.00	29	29	32	66	84	113
	0.95	49	49	58	103	144	177
	0.90	56	53	57	81	134	181

Tabela 3 - Quantidade de classificações de verdadeiramente positivas (TP)

		Cutoff					
Threshold	FP	1.00	0.95	0.90	0.85	0.80	0.75
	1.00	0	0	0	0	1	12
	0.95	0	0	0	2	18	35
	0.90	23	26	28	47	53	55

Tabela 4 – Quantidade de classificações falsamente positivas (FP)

		Cutoff					
Threshold	TN	1.00	0.95	0.90	0.85	0.80	0.75
	1.00	21.273	21.273	21.270	21.236	21.217	21.177
	0.95	21.253	21.253	21.244	21.297	21.140	21.090
	0.90	21.223	21.223	21.217	21.174	21.115	21.066

Tabela 5 – Quantidade de classificações incorretamente negativas (TN)

		Cutoff					
Threshold	FN	1.00	0.95	0.90	0.85	0.80	0.75
	1.00	113	113	110	76	59	41
	0.95	93	93	84	41	16	0
	0.90	109	115	113	108	61	16

Tabela 6 – Quantidade de classificações incorretamente negativas (FN)

Nas tabelas a seguir apresentamos os resultados referentes à performance do framework:

		Cutoff					
Threshold	Precision	1.00	0.95	0.90	0.85	0.80	0.75
	1.00	100.00%	100.00%	100.00%	100.00%	98.82%	90.40%
	0.95	100.00%	100.00%	100.00%	98.10%	88.89%	83.49%
	0.90	70.89%	67.09%	67.06%	63.28%	71.66%	76.69%

Tabela 7 - Valores de Precisão

		Cutoff					
Threshold	Recall	1.00	0.95	0.90	0.85	0.80	0.75
	1.00	20.42%	20.42%	22.54%	46.48%	58.74%	73.38%
	0.95	34.51%	34.51%	40.85%	71.53%	90.00%	100.00%
	0.90	33.94%	31.55%	33.53%	42.86%	68.72%	91.88%

Tabela 8 - Valores de Recall

		Cutoff					
Threshold	Acurácia	1.00	0.95	0.90	0.85	0.80	0.75
	1.00	99.47%	99.47%	99.49%	99.64%	99.72%	99.75%
	0.95	99.57%	99.57%	99.61%	99.80%	99.84%	99.84%
	0.90	99.38%	99.34%	99.34%	99.28%	99.47%	99.67%

Tabela 9 - Valores de Acurácia

		Cutoff					
Threshold	F-measure	1.00	0.95	0.90	0.85	0.80	0.75
	1.00	33.92%	33.92%	36.78%	63.46%	73.68%	81.00%
	0.95	51.31%	51.31%	58.00%	82.73%	89.44%	91.00%
	0.90	45.90%	42.91%	44.71%	51.10%	70.16%	83.60%

Tabela 10 - Valores de F-Measure

O experimento foi útil para determinar quais seriam os valores de *threshold* e *cutoff* para nosso domínio. Como o framework ainda não é capaz de auto ajustar-se, temos que selecionar manualmente os valores que maximizam as métricas utilizadas: 0,95 para o *threshold* e 0,80 para o *cutoff* do classificador.

Esses valores foram obtidos empiricamente e são aqueles que mantêm precisão e recall balanceados com altas taxas. Esses valores devem ser os melhores para o domínio do nosso problema. Com eles obtivemos valores de 88,9% de precisão, 90% de recall e 89,44% de *F-Measure*.

5.3 Resumo

Neste capítulo apresentamos um estudo de caso, que é a instanciação do framework, para o enriquecimento dos dados para o site Winetag.com.br[13]. No início deste capítulo há uma breve discussão acerca de detalhes relacionados aos nomes de vinhos, domínio do site estudado. Após uma definição de quais atributos seriam suficientes para diferenciar cada objeto foi demonstrado todos os passos necessários para criação da solução.

A metodologia de trabalho, que contou com 7 fases: (1) identificação de fontes, (2) alinhamento de esquemas, (3) descoberta de padrões de URL e montagem de consultas, (4) codificação de parsers, (5) escolha de funções de cálculo de similaridade, (6) codificação do classificador e finalmente (7) fusão de dados. A metodologia contém detalhes específicos do domínio dos vinhos que podem ser úteis para outras experiências de outros domínios.

Para medir o progresso realizado foi apresentado na seção de resultados as métricas que são utilizadas em trabalhos similares [6, 12, 33, 36], assim como o conjunto de testes que foi utilizado. A partir dos resultados obtidos variando dois parâmetros do framework: *threshold* da função de cálculo de similaridade e *cutoff* do classificador obtivemos os valores mais apropriados para esta instância.

13 http://www.winetag.com.br