Métodos Estatísticos para Segmentação de Listas Web

William Marx[1], **Sergio L. S. Mergen**[1]

[1]Campus Alegrete - Universidade Federal do Pampa (UNIPAMPA)
CEP – 97.546-550 – Alegrete – RS – Brasil

william.f.marx@gmail.com, sergiomergen@unipampa.edu.br

Abstract. *Data extraction from HTML lists has the goal of transforming a record list into a tabular format, composed by records and columns. This sort of extraction serves many goals, such as feeding a dataspace that associates Web data sources that refer to the same subject. There already exist proposals of mechanisms that perform this extraction. One of them in particular divides the extraction process in three stages (Splitting, Alignment and Refinement), where the first one relies on a preexistent knowledge base. In this paper, we propose extraction rules whose purpose is to replace the initial Splitting stage. Such rules explore the existence of content delimiters in the list records, and they are not dependent on previous information. Experiments show the efficiency of the extraction rules applied over real HTML lists found on the Web, in comparison with the method currently used in the Splitting stage.*

Resumo. *A extração de dados a partir de listas HTML tem por finalidade transformar uma lista de registros em um formato tabular composto por registros e colunas. Essa extração serve a diversas finalidades, como alimentar um dataspace que relaciona fontes de dados da Web que tratam de um assunto comum. Já existem mecanismos propostos que realizam essa extração. Um deles em particular divide o processo de extração em três etapas(Splitting, Alignment e Refinement) sendo que a primeira delas depende de uma base de conhecimento preexistente. Neste artigo são propostas regras de extração cujo objetivo é substituir a etapa inicial de Splitting. Tais regras exploram a existência e frequência de delimitadores de conteúdo nas linhas da lista, e independem de informações prévias para funcionar. Os experimentos mostram a eficácia das regras na extração aplicadas em listas HTML reais encontradas na Web, em comparação com o método atualmente usado na etapa de Splitting.*

1. Introdução

A produção de dados na *Web* vem crescendo exponencialmente, desde simples páginas HTML até formatos mais sofisticados como *feeds* RSS e *Web Services*. De certa forma, essas informações podem ser vistas como fazendo parte de uma gigantesca base de dados heterogênea e sem uma autoridade central que regula a estrutura dos dados e tampouco as políticas para adição e remoção de informações.

Indo um pouco mais além, é possível criar uma linha imaginária que divida essa gigantesca base de dados em bases menores, para os diferentes tipos de domínios existentes. Dessas bases menores surge um conceito que está em voga nos dias atuais: *dataspaces*. Em poucas palavras, um *dataspace* pode ser descrito como um conjunto de fontes de

dados heterogêneas que atendem a um propósito comum [Franklin et al. 2005]. A possibilidade de acessar dados de um *dataspace* aumenta a qualidade das pesquisas, e tem aplicação direta em áreas como extração de conhecimento e recuperação de informação.

Um dos desafios a ser vencido para que *dataspaces* saiam do mundo das idéias e se tornem elementos tangíveis envolve reconhecer a estrutura presente nas fontes de dados, para que mais adiante um processo de integração possa consolidar esses dados em uma base unificada, seja ela virtual ou materializada. Esse problema se torna ainda mais desafiante quando a estrutura é implícita, oculta por detrás de padrões de documentação difíceis de ser identificados.

Esse é o caso por exemplo dos dados publicados nas páginas HTML. Em boa parte dos casos, o conteúdo das páginas HTML são textos escritos em linguagem natural, que por vezes são envoltos em *tags* de marcação, que mais tem o fim de definir a formatação visual do documento do que criar uma estrutura que organize o texto em conceitos semânticos. Ou seja, informações descritas em páginas HTML tendem a ser pobres em estrutura. Ironicamente, esses documentos são a fonte mais rica de informações da *Web*, visto que a linguagem HTML é o padrão de facto para a publicação de dados na *Web*.

Dentro da linguagem HTML, existem construtores que, embora voltados primariamente para formatação visual, servem também para estruturar a informação. Um desses construtores são as listas. Verificando blocos de texto contidos em listas, é possível perceber uma divisão inicial que organiza a informação como uma coleção de registros. No entanto, cada registro por si só pode ser estruturado, dividindo o conteúdo em campos que representem informações distintas, mas que estejam associadas.

Dada essas características das listas HTML, surgiram trabalhos acadêmicos que visam transformar os blocos de dados contidos nas listas em tabelas de dados, compostas por linhas e colunas. Um desses trabalhos em especial define processos a serem executados sobre uma lista, para que ao fim seja possível realizar a transformação. No entanto, os processos requerem o uso de bases de conhecimento para auxiliar na identificação das colunas contidas em cada linha.

Dentro deste contexto, este trabalho propõe uma série de regras estatísticas que visam transformar listas em tabelas. De modo geral, as regras se valem da presença e frequência no texto de caracteres especiais que costumam ser usados para realizar a separação de conteúdo. Ainda, as regras independem da existência de bases de conhecimento, o que torna o mecanismo de extração útil em situações onde não existe uma base que possa ser usada ou a base esteja indisponível.

O texto está organizado da seguinte forma: A seção 2 apresenta alguns trabalhos publicados a respeito de extração de dados na *Web*, inclusive o trabalho que será utilizado como base de comparação nos experimentos. A seção 3 apresenta a abordagem proposta, descrevendo cada uma das regras criadas, e as situações em que elas são úteis. Os resultados dos experimentos são discutidos na seção 4. Por fim, a seção 5 apresenta as considerações finais.

2. Trabalhos Relacionados

Existem diversos estudos a respeito de abordagens automatizadas que extraem listas de objetos a partir da *Web* [Arasu and Garcia-Molina 2003, Crescenzi et al. 2001]. O processo típico de extração consiste em três passos: extração de registros, extração de atributos e rotulação.

O primeiro passo envolve identificar os registros presentes no documento em análise. Em alguns tipos de documentos essa identificação é trivial, como em listas HTML, onde as próprias linhas já separam os registros. Já em páginas na *Web*, esses registros seriam segmentos no texto que encapsulam dados referentes a um objeto específico. Entre as técnicas usadas para realizar essa busca pode-se citar o uso de algoritmos de casamento de strings [Liu et al. 2003] e processamento de linguagem natural.

O segundo passo envolve extrair atributos do objeto identificado. Considerando objetos contendo dados de filme, esses atributos poderiam ser 'titulo' e 'ano', por exemplo. O último passo envolve interpretar os atributos identificados, ou o próprio objeto que os agrega, afim de rotulá-los com nomes apropriados. Normalmente, as soluções propostas para esse problema são baseadas modelos probabilísticos e aprendizado de máquina [Wang and Lochovsky 2003, Zhu et al. 2006].

Os trabalhos mais próximos ao proposto neste artigo são aqueles que realizam o segundo passo. Existem diversos trabalhos nesta categoria, como aqueles que analisam páginas de resposta de formulários *Web* para realizar a extração de atributos. Por exemplo, em [Zhao et al. 2007] é usado um método estatístico para minerar páginas de resposta na busca de dados que obedeçam *templates* pré-definidos. Já o trabalho de [Zhai and Liu 2005] utiliza informação visual de como os dados seriam renderizados pelo navegador de *Internet* para inferir como esses dados estão estruturados.

Existem também estudos mais específicos, e ainda mais próximos a este trabalho, cujo foco é a extração de atributos de listas HTML. Em [Machanavajjhala et al. 2011], a extração é realizada de forma coletiva. Ou seja, múltiplas listas são extraídas de forma simultânea, explorando a redundância de conteúdo e estrutura contidas nas listas analisadas

Neste artigo, será dada ênfase ao trabalho proposto por [Elmeleegy et al. 2009], que é capaz de extrair atributos de uma lista por vez. A técnica de extração empregada possui três fases, chamadas de Divisão (*Splitting*), Alinhamento (*Alignment*) e Refinamento (*Refinement*). Na fase de divisão, cada linha é dividida em múltiplos campos. Mais informações sobre essa divisão serão descritas mais adiante.

Na fase de alinhamento é calculado um número esperado de colunas por linha, com base nas divisão inicial feita. As linhas que tenham ficado com menos ou mais colunas do que o esperado são ajustadas.

Na fase de divisão, uma linha é separada em vários campos candidatos, sendo que a cada campo é atribui um escore. Em seguida é aplicada uma técnica de seleção dos melhores campos, que leva em consideração o escore e inexistência de sobreposição entre os campos selecionados.

Vale a pena destacar que a geração dos campos candidatos e seus escores é feita de acordo com três escores parciais, que são ponderados e normalizados para a obtenção do escore final:

1. ***Escore de Tipo:*** Um campo recebe escore máximo se o seu conjunto de palavras corresponder a um tipo de informação reconhecido, como valores numéricos, datas, *urls*, *emails* e telefones.
2. ***Escore de Modelo de Linguagem:*** Calcula o escore de um campo com base na probabilidade condicional de cada uma das palavra do campo w_i seguir uma sequência das palavras anteriores do campo $w_1, ..., w_{i-1}$, ou seja, calcula $P_r(w_i|w_1, ..., w_{i-1})$. O escore é calculado com base na probabilidade de que a primeira palavra do campo em questão siga a última palavra do campo candidato anterior.
3. ***Escore de Compatibilidade de Tabela:*** Compara o conteúdo das listas com um *corpus* de tabelas, afim de ientificar quais informações normalmente aparecem como parte de uma mesma coluna.

Apesar de a técnica proposta por [Elmeleegy et al. 2009] ser não supervisionada, durante a fase de divisão os escores parciais de Modelo de Linguagem e de Compatibilidade de Tabela são calculados com base em um corpus textual e um corpus de tabelas, respectivamente.

O único escore parcial que independe de dados externos é o de Tipo, que procura por padrões textuais dentro das linhas da lista. A dependência da etapa de divisão por bases de conhecimentos é uma limitação da abordagem, que impede que ela seja usada integralmente para separar os atributos de listas sem uso de informações preexistentes.

3. Métodos de Divisão Propostos

Com base no exposto na seção anterior, este trabalho propõoe novas regras que podem ser usadas para substituir a etapa de divisão proposta em [Elmeleegy et al. 2009]. As regras baseiam-se na noção de que existem caracteres específicos que costumam ser utilizados para fazer a delimitação de conteúdo.

Para exemplificar, considere o trecho “Star Wars - George Lucas”. Nesse trecho, o hífen é usado para separar o nome do diretor do filme que ele dirigiu. Assim como o hífen, existem outros caracteres que servem para separar informações distintas, mas que estejam relacionadas. Neste trabalho, esse conjunto de caracteres é chamado de caracteres de separação.

A determinação de quais caracteres devem compor essa lista está fora do escopo deste artigo. Para simplificar, adota-se um critério de seleção que considera os caracteres não alfanumérico como separadores (ex. -”e ”:”). Esse critério parte da intuição de que textos normalmente são compostos por caracteres alfanuméricos, e que é mais elegante usar como separador os caracteres que não costumam aparecer no corpo do texto.

Observe que nem sempre os caracteres de separação são usados para separar conteúdo. Por exemplo, em ”Batman - O Cavaleiro das Trevas, Christopher Nolan (2008)”, o hífen tem uma função de aposto especificador que aparece em meio a um título de filme. Já em ”Star Wars Episódio I: A Ameaça Fantasma - 1999”, o hífen é usado com a função de separador. Com base nesse exemplo, surge a necessidade de analisar o contexto onde os caracteres de separação ocorrem para determinar sua real função.

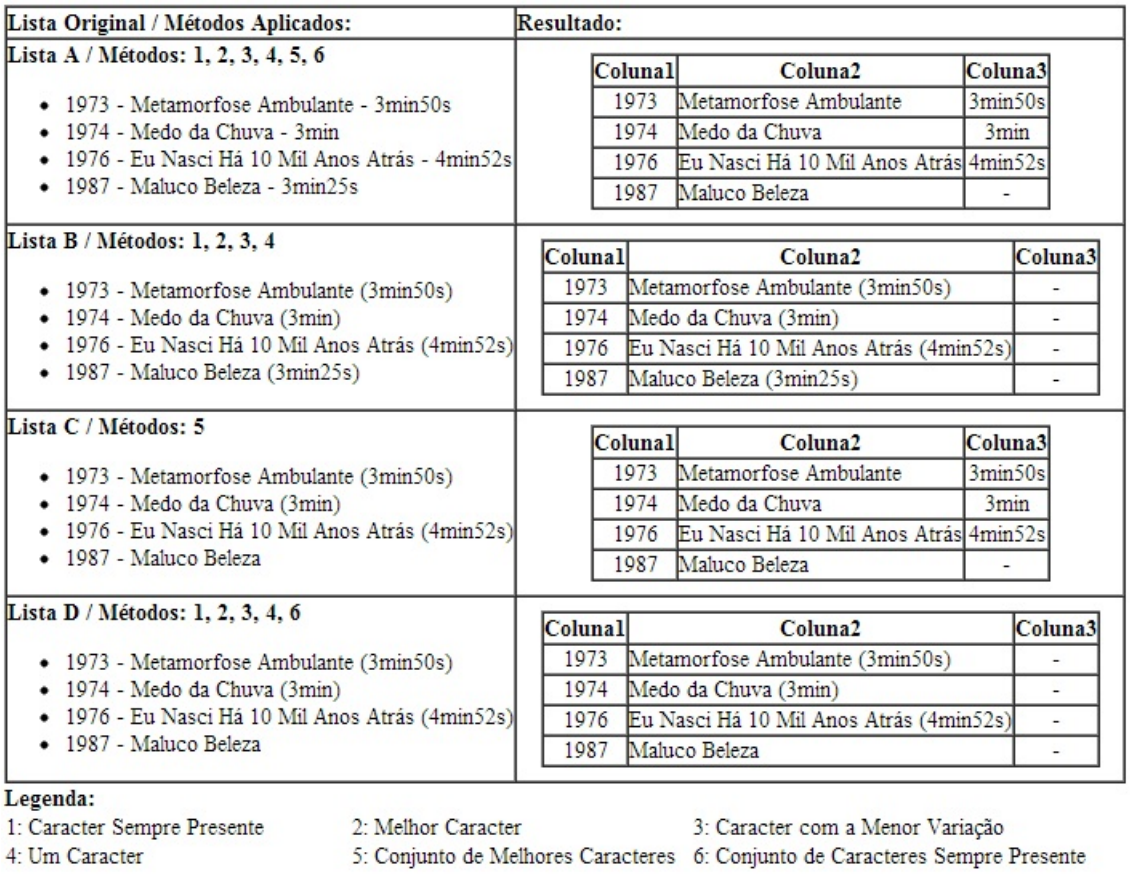
Nesta seção são descritos os métodos de segmentação de conteúdo propostos, cujo objetivo é encontrar os caracteres que exercem a função de separação. Os métodos criados

procuram inferir o contexto através de uma análise estatística, determinando a importância de cada caractere de separação com base na frequência com que ele ocorre na lista. A Figura 3.1 é usada para apoiar a explicação. Ela contém exemplos de listas, métodos usados na transformação e a respectiva segmentação, representada na forma de uma tabela com linhas e colunas.

3.1. Método Caractere Sempre Presente

Este método busca um caractere especial com maior número de ocorrências que necessariamente está presente em todas as linhas da lista.

Um exemplo de segmentação usando este método pode ser visto lista 'A' da Figura 1, onde a informação foi corretamente segmentada. Este método também foi aplicado nas listas 'B' e 'D' da Figura 1. Como em ambas existe mais de um tipo de caractere separador, o método só conseguiu acertar a primeira coluna.



Lista Original / Métodos Aplicados: | **Resultado:**

Lista A / Métodos: 1, 2, 3, 4, 5, 6

- 1973 - Metamorfose Ambulante - 3min50s
- 1974 - Medo da Chuva - 3min
- 1976 - Eu Nasci Há 10 Mil Anos Atrás - 4min52s
- 1987 - Maluco Beleza - 3min25s

Coluna1	Coluna2	Coluna3
1973	Metamorfose Ambulante	3min50s
1974	Medo da Chuva	3min
1976	Eu Nasci Há 10 Mil Anos Atrás	4min52s
1987	Maluco Beleza	-

Lista B / Métodos: 1, 2, 3, 4

- 1973 - Metamorfose Ambulante (3min50s)
- 1974 - Medo da Chuva (3min)
- 1976 - Eu Nasci Há 10 Mil Anos Atrás (4min52s)
- 1987 - Maluco Beleza (3min25s)

Coluna1	Coluna2	Coluna3
1973	Metamorfose Ambulante (3min50s)	-
1974	Medo da Chuva (3min)	-
1976	Eu Nasci Há 10 Mil Anos Atrás (4min52s)	-
1987	Maluco Beleza (3min25s)	-

Lista C / Métodos: 5

- 1973 - Metamorfose Ambulante (3min50s)
- 1974 - Medo da Chuva (3min)
- 1976 - Eu Nasci Há 10 Mil Anos Atrás (4min52s)
- 1987 - Maluco Beleza

Coluna1	Coluna2	Coluna3
1973	Metamorfose Ambulante	3min50s
1974	Medo da Chuva	3min
1976	Eu Nasci Há 10 Mil Anos Atrás	4min52s
1987	Maluco Beleza	-

Lista D / Métodos: 1, 2, 3, 4, 6

- 1973 - Metamorfose Ambulante (3min50s)
- 1974 - Medo da Chuva (3min)
- 1976 - Eu Nasci Há 10 Mil Anos Atrás (4min52s)
- 1987 - Maluco Beleza

Coluna1	Coluna2	Coluna3
1973	Metamorfose Ambulante (3min50s)	-
1974	Medo da Chuva (3min)	-
1976	Eu Nasci Há 10 Mil Anos Atrás (4min52s)	-
1987	Maluco Beleza	-

Legenda:

1: Caracter Sempre Presente
2: Melhor Caracter
3: Caracter com a Menor Variação
4: Um Caracter
5: Conjunto de Melhores Caracteres
6: Conjunto de Caracteres Sempre Presente

Figura 1. Exemplos de Segmentação Baseados nos Métodos Propostos

3.2. Método Melhor Caractere

Este método busca pelo caractere especial com maior número de ocorrências, sendo que o caractere não necessariamente precisa ocorrer em todas as linhas da lista.

O desempenho do método nos exemplos é igual ao do método anterior. Seu uso se justifica para casos de erros de digitação, onde o caractere de separação não foi incluído em alguma das linhas da tabela.

3.3. Método de Um Caractere

Este método busca pelo caractere especial com maior número de ocorrências em cada linha, e usa-o como separador na linha analisada. Isso é feito em cada linha da lista possibilitando que um caractere diferente seja escolhido em cada linha.

O desempenho do método nos exemplos é igual ao dos métodos anteriores. Seu uso se justifica para casos em que uma mesma lista é composta por tipos diferentes de linhas, situação que é bastante comum em arquivos do tipo *flat file*.

3.4. Método Caractere com a Menor Variação

Este método busca apenas um caractere separador para toda a lista, sendo que é escolhido o que apresentar a menor variação do número de ocorrências em relação a média de ocorrências de todos os caracteres encontrados na lista.

O desempenho do método nos exemplos é igual ao dos métodos anteriores. Seu uso se justifica para casos em que caracteres especiais ocorram em todas as linhas mas que não sejam de fato usados como separadores. A regra parte da intuição que esses caracteres terão uma frequência irregular pelas linhas da lista.

3.5. Método Conjunto de Melhores Caracteres

Este método faz a segmentação utilizando um conjunto de caracteres especiais como delimitadores de informação, sendo que este conjunto é o mesmo para todas as linhas da lista.

Ao contrário dos métodos anteriores, este pode selecionar um número variável de caracteres de separação. Para que um caractere seja selecionado, deve-se verificar se o número de ocorrências dele é maior que a média de ocorrências dos caracteres especiais por linha subtraída do seu desvio padrão.

Por utilizar mais caracteres de separação, o método conseguiu segmentar corretamente as colunas tanto das listas 'A' e 'C', conforme apresentado na Figura 1.

3.6. Método Conjunto de Caracteres Sempre Presente

Este método também faz a segmentação utilizando um conjunto de caracteres especiais como delimitadores de informação, sendo que este conjunto é o mesmo para todas as linhas da lista.

O método busca por todos os caracteres especiais que estão presentes necessariamente em todas as linhas da lista, independentemente do número de ocorrências em cada linha.

Apesar de trabalhar com um conjunto de caracteres, o método não segmentou corretamente a lista 'D' da Figura 1. Isso ocorreu pelo fato desta lista utilizar como separador de conteúdo os caracteres '(' e ')', que não aparecem em todas as linhas.

4. Experimentos

Esta seção apresenta os experimentos realizados para a segmentação de listas encontradas na *Web*. A seguir são apresentados os aspectos relacionados aos experimentos.

4.1. Criação dos conjuntos de teste

Para validar este trabalho foram construídos três conjuntos de testes, cada um constituído de 50 listas *Web*. As listas foram coletadas manualmente, seguindo os seguintes critérios:

- As linhas não sejam parte de listas de menu
- As linhas são compostas por textos curtos (não mais do que 500 caracteres por linha)
- As linhas possuem caracteres de separação

Com base nesses critérios, foram coletadas páginas considerando três contextos distintos, conforme descrito abaixo:

Wikipedia : Esta coleção possui listas aleatórias recuperadas a partir de páginas da *Wikipedia*. A geração da coleção foi realizada através de um recurso da *Wikipedia* que direciona o usuário a uma página aleatória. Nesta página, foram coletadas todas as listas que satisfizeram os critérios definidos acima. Em seguida, uma nova página aleatória foi acessada, e o processo se repetiu até que 50 listas tivessem sido coletadas.

Listas 10 : Esta coleção possui listas aleatórias existentes em sites que informam os 10 tópicos mais relevantes a respeito de um determinada categoria. Foram utilizadas aproximadamente 50 páginas do site *listas10.org* durante a coleta. Dentre as categorias coletadas encontram-se Cinema, Músicas e Esportes.

WT10G : Esta coleção possui listas aleatórias recuperadas a partir da *Web*. A geração da coleção foi realizada através de um *snapshot* da *Web* de 1997, contida na *corpus* WT10G[1]. As páginas da coleção estão divididas em múltiplas pastas numeradas, sendo cada pasta dividida em múltiplos documentos XML numerados. Cada documentos XML também é dividido em blocos numerados, referentes a uma página específica. Navegando aleatoriamente por essa estrutura, foi recuperada uma página a partir de onde foram coletadas as listas que satisfizeram os critérios definidos acima. Em seguida, uma nova página aleatória foi acessada, e o processo se repetiu até que 50 listas tivessem sido coletadas.

4.2. Marcação das Listas

As listas foram manualmente marcadas com a segmentação consideradas correta. Essa marcação é utilizada para medir a eficácia dos métodos propostos. Para fazer a segmentação, procurou-se dividir as linhas em colunas que pudessem ser rotuladas de modo simples e intuitivo.

Para exemplificar, a linha "Batman(1987)" seria separada em duas colunas, pois é intuitivo identificar que a primeira coluna refere-se ao nome de um filme enquanto a segunda se refere ao ano do filme. Já a linha "Batman bateu recorde de público (e de arrecadação também)" não seria separada, pois faz mais sentido manter toda a informação em apenas uma coluna, que poderia ser rotulada como "Notícia".

Outro exemplo envolve a presença de atributos multivalorados, que foram mantidos em um único segmento. Por exemplo, a linha "Batman: 1989, 1992, 1995, 1997" seria separada em dois segmentos, um para o nome do filme e outro para os anos de

[1] `http://ir.dcs.gla.ac.uk/test_collections/wt10g.html`

lançamento. Do contrário, seria necessário criar rótulos separados para cada ano, o que não é visto como uma boa prática de modelagem, até porque o número de valores de um atributo multivalorado é dinâmico.

4.3. Forma de Avaliação

A avaliação foi realizada através da medição da precisão na segmentação. Três níveis distintos de precisão foram medidos, conforme descrito abaixo:

Precisão dos Segmentos Verifica quantos dos segmentos que deveriam ser separados foram realmente identificados.

Precisão das Linhas Verifica quantas linhas foram devidamente segmentadas.

Precisão das Listas Verifica quantas listas tiveram todas suas linhas devidamente segmentadas.

A precisão das linhas e listas é mais restrita do que a precisão dos segmentos. Um único segmento não identificado na linha torna essa linha inválida, para fins de medição. Da mesma forma, uma única linha inválida em uma lista torna essa lista inválida.

4.4. Base de Comparação

Além de os métodos serem comparados entre si, eles também são comparados com uma das técnicas utilizadas no trabalho de [Elmeleegy et al. 2009], que segmenta uma lista com base na análise de padrões textuais presentes em cada linha.

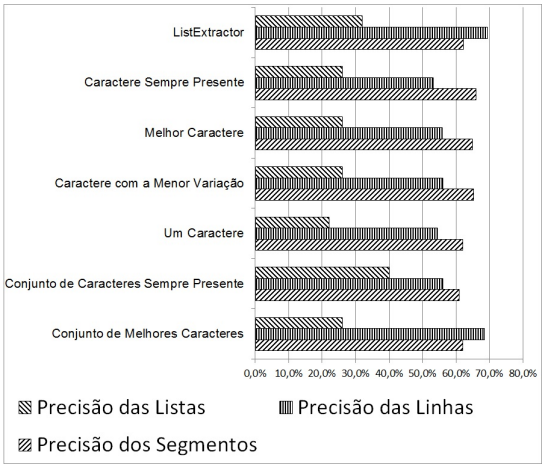
Para possibilitar a avaliação, a técnica foi implementada através de expressões regulares que reconhecem padrões textuais conhecidos, como datas, telefones e *emails*. Conforme descrito em [Elmeleegy et al. 2009], essa é uma das técnicas propostas para compor a etapa de *splitting*. As demais técnicas não foram avaliadas neste artigo uma vez que elas dependem de bases de conhecimentos preexistentes.

4.5. Análise dos Resultados

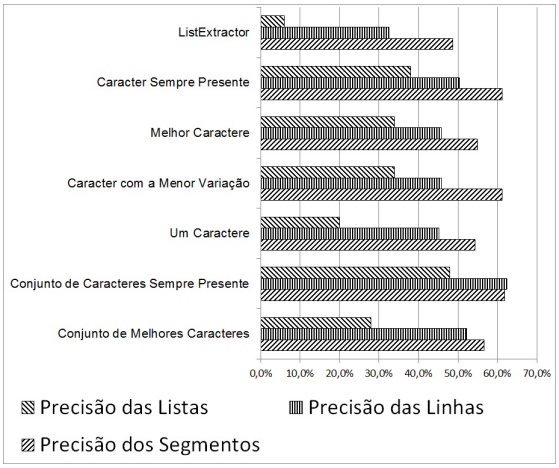
Os resultados alcançados pelos métodos de segmentação são apresentados na Figura 2. Como se pode ver, todas as técnicas obtiveram desempenho semelhante na coleção 'Wikipedia'. Destaque é dado para o método 'Conjunto de Caracteres Sempre Presente', que se sobressaiu no critério relativo ao número de listas corretamente segmentadas. Esse resultado indica que as listas publicadas em artigos da *Wikipedia* costumam usar caracteres de separação de uma forma bem comportada. O desempenho satisfatório do método 'ListExtractor' sugere que os segmentos das listas costumam usar padrões textuais bem definidos.

Na coleção 'listas 10', as listas também usam caracteres de separação de uma forma bem comportada, sendo que o método 'Conjunto de Caracteres Sempre Presente' mais uma vez se saiu melhor. Nesta coleção, destaca-se o baixo desempenho do método 'ListExtractor'. Uma possível explicação para esse fato é o número insuficiente de expressões regulares que foram implementadas. Por exemplo, valores monetários foram bastante comuns nas listas desta coleção, e nenhuma das expressões usadas conseguiam identificar esse padrão textual.

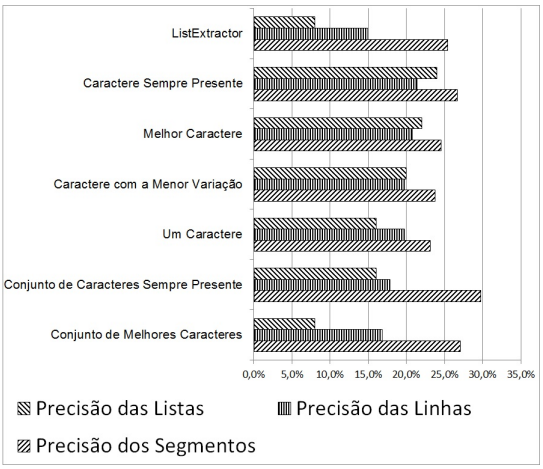
Já o desempenho geral dos métodos da coleção 'WT10G' foi bastante inferior se comparados às demais coleções. Nenhum dos métodos obteve uma precisão maior do



(a) Wikipedia



(b) Listas10



(c) WT10G

Figura 2. Resultados de Precisão Obtidos sobre Três Coleções de Listas Reais da *Web*

que 30% na segmentação de listas inteiras. Como os dados não pertencem a um domínio específico, onde padrões de exibição costumam ser adotados, os caracteres usados na separação não seguiam um formato bem comportado. Mesmo assim, percebe-se que o desempenho em todos os métodos propostos no artigo superam a técnica baseado em padrões textuais.

5. Conclusões

Este artigo apresentou métodos estatísticos que realizam a segmentação de listas. A segmentação é realizada com base na frequência de ocorrência de um conjunto de caracteres especiais que costumam ser usados para fazer a separação de conteúdo. O objetivo do trabalho foi verificar se os métodos propostos podem ser usados dentro de uma arquitetura de segmentação proposta na literatura([Elmeleegy et al. 2009]), como substituto de um de seus métodos internos que analisa a presença de padrões textuais no texto.

Foram realizados experimentos sobre coleções de listas reais extraídas da *Web*. Os resultados mostram que os métodos propostos têm um desempenho igual ou su-

perior ao método baseado em padrões textuais quando as listas utilizam caracteres de separação. O próximo passo é empregar os métodos propostos dentro da arquitetura de [Elmeleegy et al. 2009] afim de analisar como a segmentação se comporta.

Outra possibilidade de trabalho futuro envolve descobrir dinamicamente qual método é mais eficaz na segmentação de uma lista. Para atingir esse objetivo, pretende-se analisar as evidências exploradas pelas regras, na busca por correlações que indiquem a melhor forma de realizar a segmentação para cada caso específico.

Referências

Arasu, A. and Garcia-Molina, H. (2003). Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 337–348, New York, NY, USA. ACM.

Crescenzi, V., Mecca, G., and Merialdo, P. (2001). Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 109–118, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Elmeleegy, H., Madhavan, J., and Halevy, A. (2009). Harvesting relational tables from lists on the web. *Proceedings of the VLDB Endowment*, 2(1):1078–1089.

Franklin, M., Halevy, A., and Maier, D. (2005). From databases to dataspaces: a new abstraction for information management. *SIGMOD Rec.*, 34(4):27–33.

Liu, B., Grossman, R., and Zhai, Y. (2003). Mining data records in web pages. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 601–606, New York, NY, USA. ACM.

Machanavajjhala, A., Iyer, A. S., Bohannon, P., and Merugu, S. (2011). *Collective extraction from heterogeneous web lists*. ACM Press.

Wang, J. and Lochovsky, F. H. (2003). Data extraction and label assignment for web databases. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 187–196, New York, NY, USA. ACM.

Zhai, Y. and Liu, B. (2005). Web data extraction based on partial tree alignment. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 76–85, New York, NY, USA. ACM.

Zhao, H., Meng, W., and Yu, C. (2007). Mining templates from search result records of search engines. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 884–893, New York, NY, USA. ACM.

Zhu, J., Nie, Z., Wen, J.-R., Zhang, B., and Ma, W.-Y. (2006). Simultaneous record detection and attribute labeling in web data extraction. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 494–503, New York, NY, USA. ACM.