Extração de dados e metadados em textos semi-estruturados usando HMMs

Roberto Oliveira dos Santos[1], **Filipe de Sá Mesquita**[1],
Altigran Soares da Silva[1], **Eli Cortez C. Vilarinho**[1]

[1]Departamento de Ciência da Computação
Universidade Federal do Amazonas (UFAM)
Manaus – AM – Brasil

{ros,fsm,alti,eccv}@dcc.ufam.edu.br

Abstract. *The Web is abundant in pages containing implicit data items. In many cases, these data items occur in semi-structured texts without explicit delimiters and embedded within an implicit structure. In this paper, we present a novel approach for the extraction from semi-structured texts which is based on Hidden Markov Models (HMM). Distinctly from previous proposals in the literature that also use HMM, our approach emphasizes the extraction of metadata in addition to the extraction of data items themselves. Our approach consists of a nested structure of HMMs, in which a main HMM identifies implicit attributes in the text and a set of internal HMM, one for each attribute, identifies data and metadata. The HMM are generated from training using a fraction of the set of the texts from which data is to be extracted. Our experiments with classified ads taken from the Web demonstrate that the extraction process reaches quality levels superior to 0,97 using the F-measure, even if the fraction used for training is small.*

Keywords: Data Extraction, Metadata, HMM, Semi-structured text.

Resumo. *A Web é abundante em páginas que armazenam dados de forma implícita. Em muitos casos, estes dados estão presentes em textos semi-estruturados sem a presença de delimitadores explícitos e organizados em uma estrutura também implícita. Neste artigo apresentamos uma nova abordagem para extração em textos semi-estruturados baseada em Modelos de Markov Ocultos (Hidden Markov Models - HMM). Ao contrário de outros trabalhos baseados em HMM, nossa abordagem dá ênfase à extração de metadados além dos dados propriamente ditos. Esta abordagem consiste no uso de uma estrutura aninhada de HMMs, onde um HMM principal identifica os atributos no texto e HMMs internos, um para cada atributo, identificam os dados e metadados. Os HMMs são gerados a partir de um treinamento com uma fração de amostras da base a ser extraída. Nossos experimentos com anúncios de classificados retirados da Web mostram que o processo de extração alcança níveis de qualidade acima de 0,97 com a medida F, mesmo se esta fração de treinamento é pequena.*

Palavras-chave: Extração de dados, Metadados, HMM, Texto semi-estruturado.

1. Introdução

A Web pode ser considerada como sendo um grande repositório de dados, abrigando um volume crescente de dados implícitos em páginas HTML. Entretanto, na grande maioria dos casos estes dados são produzidos para consumo humano, o que torna tarefas como busca, consulta, manipulação e análise dos dados, difíceis de serem realizadas de forma automática por computadores. Por causa disto, extratores de dados ou *wrappers* são usualmente utilizados para identificar dados nas páginas e mapeá-los em bancos de dados estruturados, ou semi-estruturados, onde estes dados podem ser processados computacionalmente [Laender et al. 2002].

Muitas das páginas Web ricas em dados são geradas a partir de banco de dados estruturados, o que facilita o processo de extração dos dados nessas páginas. A despeito disso, é também comum que os dados presentes em páginas Web estejam dispostos em textos contínuos, onde não há delimitadores explícitos entre os dados, mas que escondem ainda assim uma estrutura implícita. Em [Laender et al. 2002], estes textos são chamados de *textos semi-estruturados*. Alguns exemplos deste tipo de texto são anúncios de classificados, endereços postais, referências bibliográficas, listas comerciais e currículos.

A Figura 1 apresenta dois exemplos de anúncios retirados de sites de classificados de imóveis na Web. Também são ilustrados os possíveis resultados de uma extração realizada sobre cada um deles. Observe que nos textos semi-estruturados os dados não ocorrem numa ordem fixa e muitos são opcionais. Porém, algumas pistas, deixadas para entendimento humano, podem ser utilizadas para descobrir a estrutura implicitamente presente e extrair os dados no texto. Algumas dessas pistas são: os tipos de dados (numérico, alfanumérico, outros), padrões na ordenação dos dados, delimitadores de texto (vírgula, dois pontos, outros) e *metadados*. Na Figura 1 os metadados estão identificados com asterisco (*). Consideramos como metadados, palavras-chave encontradas ao longo do texto que descrevem os dados, indicando a qual atributo eles devem ser associados. Por exemplo, no primeiro anúncio da Figura 1, observamos que a palavra-chave "qtos." indica intuitivamente que o valor "2" está associado ao atributo `QUARTO`. Portanto, consideramos que "qtos." é um metadado.

1. Bairro de Fátima. Casa térrea, c/boa localização, em bom estado de conservação, c/varanda, sl., 2 qtos., banh. soc., coz., área de serviço, quintal, garagem.	2. Ano Bom. Av. Major José Bento, ótima casa, c/ varanda, garagem, sl., 3 qtos. (ste.), banh. soc., copa, coz.
[BAIRRO: *Bairro* de Fátima.*] [TIPO: *Casa*] *térrea, c/boa localização, em bom estado de conservação, c/* [VARANDA: *varanda**], [SALA: *sl.**], [QUARTO: *2 qtos.**], [BANHEIRO: *banh.* soc.*], [COZINHA: *coz.**], [AREA: *área* de serviço*], [QUINTAL: *quintal**], [GARAGEM: *garagem**].	[BAIRRO: *Ano Bom.*] [RUA: *Av.* Major José Bento*], *ótima* [TIPO: *casa*], *c/* [VARANDA: *varanda**], [GARAGEM: *garagem**], [SALA: *sl.**], [QUARTO: *3 qtos.**] [SUITE: *(ste.*)*] [BANHEIRO: *banh.* soc.*], [COPA: *copa**], [COZINHA: *coz.**],

Figura 1. Textos semi-estruturados retirados de sites de classificado de imóveis e os possíveis resultados de um processo de extração

A extração de dados implícitos em texto semi-estruturado é um problema de grande relevância, tendo sido abordado por vários pesquisadores na literatura recente [Califf e Mooney 1999, Embley et al. 1999, Freitag 2000, Freitag e McCallum 2000, Borkar et al. 2001, Viola e Narasimhan 2005]. Neste artigo apresentamos uma abordagem para extração automática de dados e metadados baseada em Modelos de Markov Ocultos (Hidden Markov Models - HMM) [Rabiner 1989]. Um HMM é um modelo estocástico onde se assume que o sistema a ser modelado é um processo de Markov com parâmetros desconhecidos (ocultos) que podem ser estatisticamente estimados a partir de parâmetros observáveis.

A extração em textos semi-estruturados utilizando HMM foi proposta em outros trabalhos [Freitag e McCallum 2000, Borkar et al. 2001], entretanto, estes desconsideram a presença de metadados, tratando-os como dados. Um HMM possui duas características que nos permitem aplicá-lo satisfatoriamente ao problema de extração em textos semi-estruturados. A primeira é a identificação de padrões na ordem de ocorrência de elementos estruturados no texto, que o modelo realiza estimando uma probabilidade de transição de um elemento para outro. Por exemplo, em endereços postais é mais provável que o nome da rua seja seguido do número da residência, do que o contrário. Ao longo do artigo chamamos esses elementos estruturados de *atributos*. A segunda é a definição de um vocabulário para os atributos, considerando a probabilidade de cada termo ocorrer para um atributo específico. Dessa forma, dado o exemplo de classificado de imóveis da Figura 1, o modelo é apto a identificar que "Fátima" é um valor do atributo `BAIRRO`, ao invés de identificá-lo como valor de `RUA`, por exemplo, uma vez que "Fátima" ocorre mais freqüentemente no primeiro caso.

Formalmente, consideramos que um texto semi-estruturado é um conjunto de atributos $TS = \{A_1, A_2, \ldots, A_n\}$, onde cada atributo é formado por um par $A_i = \{D, M\}$ de dados $D = \{d_1, d_2, \ldots, d_k\}$ e metadados $M = \{m_1, m_2, \ldots, m_l\}$, onde os metadados em M e os dados em D são opcionais. Quando os dados estão ausentes, $D = \emptyset$, consideramos que o dado implícito é VERDADEIRO caso o metadado ocorra no texto. Portanto, a tarefa de extração de dados e metadados pode ser considerada como um processo que identifica o conjunto de atributos no texto semi-estruturado TS. No entanto, observamos que há termos no texto que não se encaixam em nenhum atributo, ou que não se quer extrair. Por exemplo, os termos em "c/boa localização, em bom estado de conservação" do primeiro anúncio da Figura 1. Em casos como este, propomos a criação de uma "área" especial que recebe estes termos, chamados de *termos extras*. Embora não sejam associados a atributos, estes termos são extraídos para que possam ser utilizados em aplicações diversas que envolvem, por exemplo, busca baseada em palavras-chave.

Ao contrário de trabalhos anteriores na literatura [Embley et al. 1999, Freitag e McCallum 2000, Borkar et al. 2001, Viola e Narasimhan 2005], consideramos que a identificação e extração de metadados em textos semi-estruturados têm grande importância. Isso ocorre por três motivos: (1) metadados ajudam a identificar os dados com maior precisão, principalmente para valores numéricos, (2) os metadados no texto não pertencem ao domínio do atributo destino, devendo ser extraídos separadamente dos dados. Por exemplo, assumindo que o domínio do atributo `QUARTO` é numérico, não podemos considerar "2 qtos." como um valor deste atributo. Por outro lado, identificando "2" como dado e "qtos." como metadado, é possível associar o valor `2` ao

atributo QUARTO. Ainda, (3) o conjunto de metadados extraído pode ser utilizado em várias aplicações que envolvem busca baseada em palavras-chave, ajudando a identificar automaticamente quando os termos da consulta do usuário são valores (dados) ou atributos (metadados) do banco de dados alvo da busca.

Neste trabalho propomos um modelo de extração em dois níveis que utiliza uma estrutura aninhada de HMMs, onde um HMM *principal* identifica os atributos no texto $TS = \{A_1, A_2, \ldots, A_n\}$ e um conjunto de HMMs *internos*, uma para cada atributo, identificam os dados e metadados $A_i = \{D, M\}$. Dessa forma, a probabilidade de transição entre atributos é calculada no primeiro nível pelo HMM principal, e a identificação de dados e metadados é realizada no segundo nível pelos HMMs internos de cada atributo. Os HMMs de ambos os níveis são gerados a partir de um processo de aprendizado supervisionado, onde é fornecido um conjunto de amostras de texto manualmente estruturado para treinamento.

Embora seja esperado que a extração de dados tenha maior eficácia com a presença de metadados, o processo de identificação de metadados no texto não é um problema simples. No contexto de anúncio de classificados, encontramos várias instâncias de metadados que indicam o mesmo atributo, como "quarto", "qto", "qtos", "dorm" e "dormitório". Observamos ainda que numa base de treinamento satisfatória deve-se priorizar a identificação de tais instâncias, ao invés de cobrir grande parte das instâncias de dados, como é comum em extração de textos. Isto significa que o modelo proposto neste trabalho necessita, em princípio, de menos exemplos durante a fase de treinamento que outro modelos para extração de dados, conforme apresentado em nossos experimentos. Por exemplo, não é necessário dar exemplos de todos os possíveis valores do atributo COR (azul, verde, amarelo, entre outros), desde que haja no texto um metadado "cor" que descreva tais valores. Conseqüentemente, uma vez que o número esperado de instâncias distintas de metadados é muito menor do que o número de instâncias de dados, podemos alcançar bons resultados com um esforço manual mínimo na fase de treinamento.

De fato, experimentos apresentados neste artigo mostram que treinando com um número de exemplos que é cerca de 20% do número de textos semi-estruturados a serem extraídos, o modelo foi capaz de extrair dados e metadados com precisão média maior que 97%. Além disso, usando medida F, que combina precisão e revocação, a tarefa de extrair corretamente os atributos presentes em cada texto obteve valores médios acima de 0,97 para todas as bases usadas no experimento. Esta proporção de exemplos para treinamento é consideravelmente menor que a proporção apresentada nos trabalhos anteriores, como em [Embley et al. 1999],[Freitag e McCallum 2000],[Borkar et al. 2001] e [Viola e Narasimhan 2005].

As principais contribuições deste artigo são listadas a seguir. (1) Apresentamos uma nova formulação para o problema de extração em texto semi-estruturado a qual considera a extração dos metadados existentes no texto, além dos dados propriamente ditos. Trabalhos anteriores na literatura não lidam com a separação de dados e metadados, o que pode implicar em imprecisão na extração; (2) Propomos uma nova abordagem para este problema, baseada em dois níveis de HMM, que generaliza abordagens anteriormente propostas [Freitag e McCallum 2000, Borkar et al. 2001], pois, embora considere a presença de metadados, pode lidar com situações onde somente dados estão disponíveis para extração; (3) Implementamos a abordagem proposta e executamos experimentos so-

bre conjuntos de anúncios de classificados coletados da Web. Estes experimentos mostram que a abordagem atinge níveis elevados de eficácia mesmo treinando com poucos exemplos, o que pode ser explicado em parte pela ênfase que é dada à identificação de metadados.

O restante deste artigo está organizado da seguinte maneira. Na Seção 2, revisamos os trabalhos relacionados. Na Seção 3, discutimos em detalhes o uso de HMM para extração de dados. Nossa abordagem baseada em dois níveis de HMMs para extração de dados e metadados é apresentado na Seção 4. Apresentamos os experimentos conduzidos e a análise dos resultados na Seção 5. Finalmente, as nossas conclusões e direções para trabalhos futuros são apresentadas na Seção 6.

2. Trabalhos Relacionados

De acordo com [Laender et al. 2002], as técnicas propostas na literatura para extração de dados são direcionadas a pelo menos um de dois tipos de páginas Web: as que contêm dados semi-estruturados e as que contêm texto semi-estruturado. No primeiro caso, os dados estão formatados para serem reconhecidos individualmente, portanto as técnicas para extração de dados semi-estruturados utilizam a estrutura sintática das páginas (marcações HTML) para extrair os dados. No segundo caso, as páginas apresentam os dados em textos contínuos onde a estrutura pode ser apenas inferida. Uma discussão mais completa sobre técnicas de extração para os dois casos é apresentada em [Laender et al. 2002].

Diferentes abordagens têm sido apresentadas para o problema de extração em textos semi-estruturados. O uso de ontologias para extração de dados é proposto em [Embley et al. 1999]. Neste trabalho, um especialista cria manualmente um modelo ontológico que descreve um domínio de aplicação específico. A partir desse modelo, são geradas regras de extração, ou expressões regulares, que são usadas para identificar dados e palavras-chave no texto semi-estruturado. O conceito de palavras-chave é similar ao conceito de metadados adotado em nosso trabalho. Contudo, a identificação de palavras-chave é feita manualmente. Além disso, elas são utilizadas exclusivamente para ajudar na extração de dados. No nosso caso, o problema consiste em identificar instâncias de metadados, independentemente se foram previamente definidas no treinamento ou não, e extraí-las para uso futuro.

RAPIER [Califf e Mooney 1999] é uma ferramenta de extração que gera expressões regulares a partir de treinamento supervisionado. Dados documentos-modelo ou *templates*, cujos dados de interesse são demarcados manualmente, a ferramenta gera um conjunto de regras de extração através de um algoritmo baseado em lógica indutiva. As regras utilizam como evidências para extração as palavras e as classes gramaticais (substantivo, adjetivo, entre outros) das palavras que circundam os dados. Esta abordagem considera uma estrutura gramatical fixa no texto, o que não ocorre em nosso caso.

Recentemente, alguns métodos de extração baseados em modelos estatísticos têm atraído interesse de pesquisadores. Em [Viola e Narasimhan 2005] o domínio de aplicação é descrito por *gramáticas livres de contexto*, que são criadas manualmente a fim de rotular cada termo de um texto semi-estruturado. Ao fim do processo de rotulação, é obtida uma árvore gramatical, onde os nós folhas são os termos rotulados e os nós não-folha são os rótulos. Este trabalho faz uma comparação com os modelos de Mar-

kov condicionais, atingindo melhores resultados nos experimentos realizados. Contudo, assume-se que nos textos semi-estruturados a serem extraídos há apenas dados, desconsiderando a presença de metadados e termos extra ao longo do texto.

A aplicação de HMMs para o problema de extração de dados tem sido abordada em alguns trabalhos. Em [Freitag e McCallum 2000] são gerados HMMs independentes para cada atributo a ser extraído. O método proposto encontra boas estruturas para os HMMs através de um processo de otimização estocástico. Como neste modelo os HMMs são independentes, há o risco do mesmo segmento de texto ser rotulado mais de uma vez. Este risco é evitado pela ferramenta DATAMOLD [Borkar et al. 2001], que utiliza um modelo de HMMs aninhados para segmentação de textos. O problema de segmentação consiste em dividir o texto em segmentos estruturados e rotular cada segmento adequadamente. Os elementos estruturados (rótulos) são modelados como estados do HMM mais externo, e cada estado representa um HMM interno que faz a identificação dos dados, levando em consideração os tipos de termos (números, palavras e delimitadores) e a seqüência em que eles ocorrem.

O problema de extração de dados em texto semi-estruturado abordado pelos trabalhos acima citados pode ser visto como um caso específico do problema de extração de dados e metadados implícitos em texto semi-estruturado formulado no presente artigo. Nestes trabalhos, é desconsiderada a ocorrência de metadados no texto a ser extraído. Contudo, muitos textos semi-estruturados disponíveis na Web são ricos em metadados, como é o caso de anúncios de classificados. Este fato aumenta a importância do problema de extração de metadados.

3. Extração em Textos Semi-estruturados com HMM

Nesta seção apresentaremos as definições clássicas dos Modelos de Markov Ocultos (HMM) e o algoritmo baseado em programação dinâmica que permite a extração de dados em texto semi-estruturado. Tais definições são necessárias para simplificar a apresentação do modelo de extração de dados e metadados em dois níveis discutido na Seção 4.

3.1. Modelo HMM clássico

Um HMM é um autômato finito probabilístico, onde, como é usual, os vértices são chamados de *estados* e as arestas são as transições entre os estados. Para cada aresta é associada uma *probabilidade de transição*. O autômato consome uma seqüência finita de *símbolos*, ou *observações*, levando em consideração as probabilidades de transição de um estado para outro e as *probabilidades de emissão*, ou seja, a probabilidade de um determinado símbolo ser emitido por um estado específico. Na Figura 2 é ilustrado graficamente um exemplo de HMM.

Definição 1 *Formalmente, um HMM é formado pelo seguintes elementos: um conjunto de estados* $E = \{e_0, e_1, e_2, \ldots, e_n, e_{n+1}\}$ *de tamanho* $N = n + 2$*, onde o estado* e_0 *corresponde ao estado* `INÍCIO` *e* e_{n+1} *corresponde a* `FIM`*; um conjunto de símbolos* $S = \{s_1, s_2, \ldots, s_m\}$ *de tamanho* M*; uma matriz de probabilidades de transição entre os estados* $A[N, N]$ *onde a probabilidade de transição do estado* e_i *para o estado* e_j *é dada por* $A[i, j]$*; e uma matriz de probabilidades de emissão de símbolos* $B[N, M]$*), onde a probabilidade de um símbolo* s_k *ser emitido pelo estado* e_j *é dada por* $B[j, k]$*.*

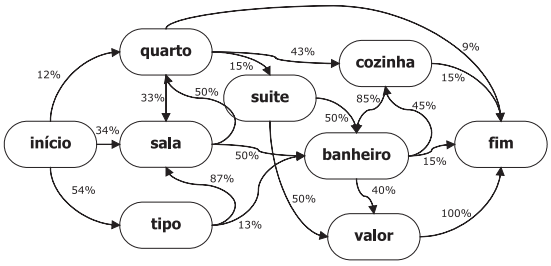

Figura 2. Exemplo de um HMM com estados e probabilidades de transições.

Os elementos de um HMM são obtidos na fase de treinamento, onde são informados os estados E, o dicionário de símbolos S e seqüências de pares $\langle e, s \rangle$, onde o estado e emite o símbolo s. Através destas seqüências pode-se estimar as probabilidades de transição e emissão, definindo os valores das matrizes A e B, respectivamente. No modelo clássico de HMM [Rabiner 1989], são propostas duas funções para o cálculo desses valores. A função que calcula a probabilidade de transição entre os estados e_i e e_j é definida a seguir:

$$A[i,j] \quad \frac{t_{ij}}{tout_i} \tag{1}$$

onde t_{ij} é o número de transições do estado e_i para o estado e_j e $tout_i$ é o total de transições saindo do estado e_i. A probabilidade de um símbolo s_k ser emitido por um estado e_j é definida a seguir:

$$B[j,k] \quad \frac{f_{jk}}{n_j} \tag{2}$$

onde f_{jk} é o número de vezes que o símbolo s_k foi emitido por e_j e n_j é número total de símbolos emitidos pelo estado e_j.

No contexto de extração de dados, os atributos são considerados como os estados de um HMM e os dados são considerados como os símbolos. Dessa maneira, dada uma seqüência de símbolos $O = o_1, o_2, \ldots, o_p$, o processo de extração de dados consiste em encontrar a seqüência de estados $\langle h_1, h_2, \ldots, h_p \rangle$ mais provável, onde o símbolo o_i é emitido pelo estado h_i, associando, portanto, um atributo para cada dado.

Uma solução simples para encontrar a seqüência de estados mais provável seria calcular a probabilidade de todas as seqüências possíveis. Tal solução é de ordem $O(x^N)$, onde x é o tamanho da entrada, tornando-a impraticável. Contudo, existe um algoritmo baseado em programação dinâmica, chamado de algoritmo de Viterbi [Rabiner 1989], que encontra a seqüência de estados mais provável num custo de $O(xN^2)$. Este algoritmo é discutido a seguir.

Algoritmo de Viterbi

O Algoritmo de Viterbi (AV) é um algoritmo baseado em programação dinâmica usado para encontrar a seqüência de estados mais provável, ou seja, o melhor caminho no HMM. O algoritmo recebe como entrada uma seqüência de observações $O = o_1, o_2, ..., o_T$ e um HMM de acordo com a Definição 1.

Para encontrar a melhor seqüência de estados $Q = q_1, q_2, ..., q_T$, dada a seqüência O, define-se a função recursiva $v(i, j)$ que retorna a probabilidade do melhor caminho

levando em consideração os i primeiros símbolos da seqüência O e terminando no estado q_j. Sejam q_0 e q_{T+1} os estados especiais INÍCIO e FIM, respectivamente. A função $v(i, j)$ é definida a seguir:

$$v(i,j) \quad \begin{cases} 1, & \textit{se } i = j = 0 \\ 0, & \textit{se } i = 0 \textit{ e } j \neq 0 \\ B[j, sb(o_i)] \cdot \max\limits_{1 \leq k \leq N} \{v(i-1,k) \cdot A[k,j]\}, & \textit{se } i > 0 \end{cases} \tag{3}$$

onde $sb(o_i)$ é uma função que retorna o índice do símbolo o_i na matriz B. Portanto, a probabilidade do melhor caminho, é dada por:

$$P_{T+1} = \max_{1 \leq k \leq N} \{v(T,k) \cdot A[k, T+1]\} \tag{4}$$

Intuitivamente, a função $v(i, j)$ encontra o melhor caminho $Q = \{q_1, q_2, \ldots, q_i\}$ levando em consideração a probabilidade do estado q_i emitir o símbolo o_i, a probabilidade do melhor caminho $\{q_1, q_2, \ldots, q_{i-1}\}$ até o símbolo o_{i-1} e a probabilidade de transição de q_{i-1} para q_i, conforme a definição do terceiro caso da recursão ($i > 0$). Os dois primeiros casos asseguram que apenas a probabilidade de transição do estado INÍCIO para q_1 será considerada como probabilidade inicial. O algoritmo de Viterbi resolve a Equação 3 usando programação dinâmica e constrói a seqüência de estados Q armazenando o estado que compõe o melhor caminho parcial a cada recorrência.

Ao final do processo, cada símbolo o_i da seqüência de entrada está associado a um estado q_i do HMM e, portanto, associado também ao atributo correspondente a este estado.

4. Modelo de Extração em Dois Níveis

O problema de extrair dados e metadados de um atributo implícito num texto semi-estruturado envolve dois problemas: (1) delimitar os atributos implícitos no texto, e (2) identificar os dados e metadados deste atributo. Entretanto, no modelo clássico de HMM, os estados podem emitir apenas um símbolo a cada instante. Essa restrição impede que os problemas (1) e (2) sejam tratados apropriadamente no modelo clássico.

Como alternativa, propomos uma estrutura de HMMs aninhados em dois níveis conforme ilustrado na Figura 3. Neste modelo o HMM principal é responsável por tratar o problema (1), modelando os atributos como estados, sem lidar diretamente com os símbolos, e os HMMs internos são responsáveis por tratar o problema (2), consumindo os símbolos e identificando-os como dados numéricos ($\mathcal{N}$), dados alfanuméricos ($\mathcal{W}$) ou metadados ($\mathcal{M}$) do atributo. Portanto, cada estado do HMM principal possui um HMM interno próprio que lida com os símbolos, solucionando a restrição do modelo clássico.

A estratégia de extração em dois níveis utilizando HMMs aninhadas já foi utilizada anteriormente em [Borkar et al. 2001]. Porém naquele trabalho a função das HMMs internas é definir com mais precisão as probabilidades de transição entre os dados de um atributo, e não identificar dados e metadados. Nas seções seguintes descrevemos os componentes de nossa abordagem de extração.

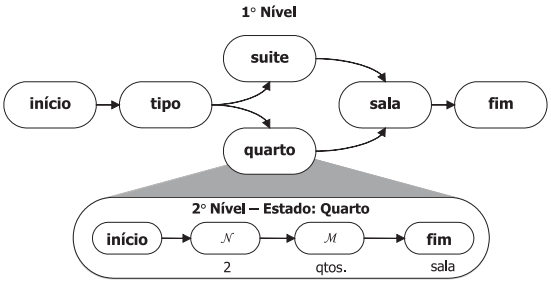

Figura 3. Exemplo de HMMs em dois níveis

4.1. Identificando os atributos no texto semi-estruturado

No primeiro nível do modelo, o HMM principal armazena as probabilidades de transição entre os estados, como no modelo clássico, porém as probabilidades de emissão são dadas pelos HMMs internos de cada estado, que encontram-se no segundo nível. Este fato exige uma alteração no Algoritmo de Viterbi, de forma a substituir a matriz de probabilidade de emissão B, pelo cálculo da probabilidade do caminho mais provável no HMM interno. Para simplificar a discussão, chamaremos de *AV principal* e *AV interno*, os algoritmos de Viterbi modificados para o primeiro e segundo nível do modelo, respectivamente.

Assim, todas as vezes que o AV principal necessita da probabilidade de um estado emitir um conjunto de observações, o algoritmo executa o AV para o HMM interno, que então consome os símbolos do atributo e encontra a seqüência de estados mais provável, retornando a probabilidade da seqüência e a quantidade de símbolos consumidos. Por exemplo, considerando o modelo da Figura 3, o caminho de estados mais provável da HMM interna de `QUARTO` é $\{\mathcal{N}, \mathcal{M}, \texttt{FIM}\}$, que identifica "2" como um dado numérico, "qtos." como metadado e "sala" como delimitador do atributo, conforme será explicado na seção seguinte, sendo apenas dois $(2, qtos)$ a quantidade de símbolos emitidos pelo atributo.

Os termos extras que não devem ser mapeados para o esquema formal de texto semi-estruturado $TS = \{A_1, A_2, \ldots, A_n\}$ são considerados como dados de um atributo chamado `EXTRA`. Tal atributo também fará parte do treinamento, sendo tratado como indistintamente como qualquer outro atributo.

4.2. Identificando dados e metadados nos atributos

No segundo nível do modelo, os HMMs internos devem ser capazes de identificar dados e metadados de um atributo num texto semi-estruturado. Além disso, o HMM deve consumir apenas os símbolos de seu atributo, identificando os símbolos de outros atributos e considerando-os como *delimitadores*, ou seja, pontos de parada para o AV interno. Para isto, assume-se na fase de treinamento que o primeiro símbolo s emitido pelo atributo A_i é um delimitador do atributo anterior A_{i-1}, considerando que s também foi emitido pelo estado interno `FIM` do estado e_{i-1}.

Para encontrar o caminho de estados mais provável no HMM interno, o AV necessita de apenas uma alteração na condição de parada. Além de finalizar o processamento quando toda a observação foi consumida, o algoritmo pode parar antes disto, no caso do

último estado do melhor caminho parcial ser o estado especial FIM. Isto indica que é mais provável que o último símbolo consumido seja de outro estado de primeiro nível, e por isso todas as observações restantes devem ser desconsideradas. Observe que a função do estado especial FIM na definição inicial era apenas contabilizar a probabilidade do um estado ser o último a emitir símbolos. Estendemos esta definição para os HMMs internos conforme explicado.

4.3. Treinamento do modelo em dois níveis

A fase de treinamento consiste em definir os estados da HMM principal, definir o conjunto de símbolos do dicionário dos HMMs internos, calcular as probabilidades de emissão dos símbolos nos HMMs internos e, finalmente, definir as probabilidades de transição em ambos os HMMs. Estas tarefas são realizadas automaticamente a partir dos dados de treinamento fornecidos. Em particular, os estados da HMM principal e o dicionário das HMMs internas são definidos, respectivamente, pelos estados e símbolos presentes nos dados de treinamento. As probabilidades de transição e emissão são calculadas conforme explicado a seguir.

Considere um conjunto de textos semi-estruturados cujos atributos estão rotulados manualmente como sendo a *base de treinamento*. Os termos nestes textos semi-estruturados também recebem um rótulo identificando-os como dados ou metadados. Formalmente, cada texto semi-estruturado $TS_i = \{A_1, A_2, \ldots, A_n\}$ da base de treinamento é um conjunto ordenado de atributos, onde cada atributo A_j representa os termos associados a este atributo. Além disso, cada termo em TS é um elemento do conjunto de dados $D_i = \{d_1, d_2, \ldots, d_k\}$ ou do conjunto de metadados $M_i = \{m_1, m_2, \ldots, m_l\}$. Por exemplo, o texto semi-estruturado "casa 2 qtos." poderia ser definido como $TS = \{$TIPO, QUARTO$\}$, $D = \{2\}$ e $M = \{$casa, qtos.,$\}$, onde TIPO=$\{$casa$\}$, QUARTO=$\{$2 qtos.$\}$. Uma vez definida a base de treinamento, podemos discutir o processo de construção dos HMMs de primeiro e segundo nível.

Definição 2 *Um HMM principal é formado por: um conjunto de estados* $E = \{e_0, e_1, e_2, \ldots, e_n, e_{n+1}\}$ *de tamanho* $N = n + 2$ *que representam os atributos presentes na base de treinamento, onde* e_0 *corresponde ao estado* INÍCIO *e* e_{n+1} *corresponde ao estado* FIM*; uma matriz de probabilidades de transição entre os estados* $A[N, N]$*, onde a probabilidade de transição do estado* e_i *para o estado* e_j *é dada por* $A[i, j]$*.*

Definição 3 *Um HMM interno é formado por: um conjunto de estados* $E' = \{$ INÍCIO, $\mathcal{M}, \mathcal{N}, \mathcal{W}$, FIM$\}$ *de tamanho* 5*. Um conjunto de símbolos* $S' = s_1, s_2, \ldots, s_h$ *de tamanho* M*. Uma matriz de probabilidades de transição entre os estados* $A'[5, 5]$*. Uma matriz de probabilidades de emissão* $B[5, M]$*. Nesta matriz a probabilidade de um símbolo* s_k *ser emitido* j*-ésimo estado é dada por* $B[j, k]$*.*

As probabilidades de transição das matrizes A e A' são calculadas pela Equação 1, utilizada no modelo clássico. Porém, o cálculo da probabilidade de emissão no modelo clássico, definido na Equação 2, penaliza severamente os estados que emitem muitos símbolos, favorecendo os que emitem poucos. Intuitivamente, consideramos que a probabilidade de emissão de um símbolo s por um estado e deve levar em consideração a freqüência com que e emite s, na base de treinamento, em comparação com a freqüência com que outros estados emitem o símbolo s. Dessa forma, a probabilidade de emissão é calculada pela função AF [Mesquita et al. 2006], que é utilizada num modelo de

estruturação de consultas sobre banco de dados, associando cada termo de uma consulta não-estruturada ao seu mais provável atributo no banco de dados. As probabilidades de emissão do símbolo s_k pelo j-ésimo estado do HMM interno são portanto definidas pela equação:

$$B[j,k] \quad AF(j,k) \quad \frac{f_{jk}}{n_k} \times \frac{f_{jk}}{max_j} \tag{5}$$

onde, f_{jk} é o número de vezes que o j-ésimo estado emite s_k, n_k é o total de vezes que s_k foi emitido e max_j é o número de emissões do símbolo mais freqüentemente emitido pelo j-ésimo estado.

4.4. Suavização do modelo

Mesmo que o modelo seja construído a partir de uma grande base de treinamento, ainda assim é inevitável que o treinamento seja insuficiente por dois motivos: (1) caso ocorra no texto semi-estruturado a ser extraído algum termo que não pertence ao dicionário de símbolos dos HMMs internos, a probabilidade do caminho mais provável será zero; e (2) o mesmo acontece quando uma transição entre os atributos implícitos no texto a ser extraído não foi definida pelos exemplos da base de treinamento.

A solução para os dois problemas é a suavização do modelo, que consiste em atribuir pequenas probabilidades para símbolos e transições não treinados. Para solucionar o primeiro caso, um símbolo especial s_0 é acrescentado aos dicionários dos HMMs internos. Na matriz de probabilidade de emissão, o valor $B[j,0]$ corresponde a probabilidade do estado e_j emitir um símbolo desconhecido. Tal probabilidade é definida a seguir.

$$B[j,0] = \frac{1}{b \cdot (max_e - f_j + 1)} \tag{6}$$

onde b é uma constante, max_e é o número de emissões do estado que emitiu mais símbolos considerando todos os HMM internos e f_j é o total de símbolos emitidos pelo j-ésimo estado do HMM interno. Intuitivamente, esta equação considera que os estados que emitem mais símbolos no treinamento tem maior probabilidade de emitir um símbolo desconhecido. A constante b (nos experimentos $b = 1000$) deve ser definida tal que $b > (n_k \times max_j)$ para todo k e j, (ver Equação 5) de forma que a probabilidade de um símbolo desconhecido seja sempre menor que a probabilidade de um símbolo presente no dicionário.

A solução para o segundo caso consiste na associação de uma probabilidade mínima para todas as transições da HMM principal, para que haja sempre uma transição entre dois atributos. Porém, ao final da fase de treinamento, a probabilidade calculada pela Equação 1 é a associada às transições que ocorrem na base de treinamento, de forma que estas tem sempre maior probabilidade que as transições que não ocorrem. A probabilidade mínima é definida pela equação a seguir.

$$A[i,j] \quad \frac{1}{a \times (max_t - tin_i + 1)} \tag{7}$$

onde a é uma constante (nos experimentos $a = 100$), max_t é o número de transições saindo de estados distintos para o estado que possui mais transições distintas para si e tin_i é o número de transições para o estado e_i. Informalmente, esta equação considera que a probabilidade de uma transição de e_i para e_j é maior, quando o estado e_j recebe mais transições de estados diferentes na fase de treinamento.

5. Experimentos e Resultados

Nesta seção apresentamos e analisamos os experimentos realizados com o objetivo de verificar a eficácia do método proposto. Para isto, utilizamos anúncios de classificados de imóveis disponíveis na Web, cujos sites estão listados na Tabela 1. Os textos dos anúncios foram extraídos de páginas coletadas de sete sites distintos, através de um processo automático cuja descrição omitimos por considerá-la fora do escopo deste artigo.

Nr.	Web Site	URL
1	*Diário Uol*	http://www.diarioon.com.br/classificados/
2	*Primeira Mão*	http://www.primeiramao.com.br/bancodeimoveis/
3	*Classificados JC*	http://classificadosjc.uol.com.br/
4	*Aranha Web*	http://gratis.aranhaweb.com.br/
5	*Folha Online*	http://classificados.folha.uol.com.br/classificados
6	*Jornal Classificados*	http://www.jornalclassificados.com.br/jornalclassificados.com
7	*Classificados Manaus*	http://www.classificadosmanaus.com.br/imoveis/

Tabela 1. Web sites utilizados nos experimentos.

Para cada um dos sites, foram escolhidos 20 anúncios considerados representativos, ou seja, que descreviam bem os atributos, dados e metadados, de forma a compor a base de treinamento para cada site. A partir destas bases, os modelos de extração em dois níveis foram treinados de forma independente, gerando um modelo para cada site. Em seguida, foram selecionados aleatoriamente 100 anúncios de imóveis de cada site. Sobre cada um destes conjuntos de anúncios foi executado o processo de extração utilizando os HMMs correspondentes. Note que o número de anúncios usados como exemplo corresponde a 20% do total de anúncios usados nos experimentos. Como apresentado a seguir, este número foi suficiente para se alcançar resultados de alta qualidade na extração.

Para avaliação dos experimentos, os resultados da extração foram analisados manualmente com o objetivo de verificar a eficácia do processo em três granularidades: (1) dados e metadados, (2) atributos e (3) anúncios como um todo. Nas seções seguintes, apresentamos e discutimos os resultados desta análise para as três granularidades.

Deve ser notado que, como nenhuma das abordagens anteriormente propostas para extração de texto semi-estruturado trata a questão da extração de metadados, não nos foi possível apresentar uma comparação empírica direta com estas abordagens. No entanto, destacamos que o nível de precisão alcançado é superior aos resultados destas abordagens, usando parâmetros experimentais semelhantes.

5.1. Identificação de Dados e Metadados

Para mostrar a qualidade da identificação de dados e metadados, mostramos na Tabela 2 os resultados da análise de extração com respeito à identificação correta dos termos de um anúncio como sendo um dado ou um metadado.

	1	*2*	*3*	*4*	*5*	*6*	*7*	*Média*
Dados	98,52%	97,59%	97,21%	97,22%	96,63%	98,61%	98,04%	97,67%
Metadados	99,12%	96,49%	98,43%	98,09%	98,90%	99,01%	99,51%	98,54%
Extra	99,01%	97,03%	99,20%	99,93%	98,92%	97,34%	98,16%	98,40%
Combinado	98,93%	97,06%	98,46%	98,11%	98,41%	98,25%	98,45%	98,25%

Tabela 2. Médias por site dos valores percentuais da corretude na identificação de dados e metadados nos anúncios.

Nesta tabela, as colunas rotuladas de 1 a 7 correspondem aos Web sites descritos na Tabela 1, nesta mesma ordem. Na linha rotulada “Dados” indicamos o percentual de

acertos para os termos que foram identificados como dado, e na linha rotulada "Metadados" indicamos o percentual de acertos para os termos que foram identificados como metadado. A linha "Extra" indica o percentual de termos extras, que não podem ser identificados como dados ou metadados de algum outro atributo, conforme explicado anteriormente. Esta linha em particular busca mostrar a capacidade do nosso método em identificar corretamente termos considerados como "ruído" para o processo de extração. Finalmente, a linha "Combinado" considera a eficácia combinada na identificação de dados, metadados e termos extra. Os percentuais apresentados nesta linha diminuem sempre que qualquer termo tiver sido incorretamente identificado.

Deve ser notado que os resultados na Tabela 2 refletem o desempenho dos HMMs internos na identificação dos dados e metadados presentes nos atributos dos anúncios. Estes resultados mostram que o HMM interno é bastante nesta tarefa dados e metadados, atingindo precisão média acima de 97% em todos os casos.

5.2. Identificação de atributos

Os resultados obtidos para a granularidade de atributo são apresentados na Tabela 3. Novamente, nesta tabela as colunas correspondem aos Web sites descritos na Tabela 1. Cada uma das linhas corresponde a um atributo modelado na fase de treinamento. Por uma questão de economia de espaço, somente são mostrados nesta tabela os atributos comuns a todos os sites. Para os demais atributos, ou seja, aqueles encontrados nos anúncios de apenas alguns dos sites, apresentamos somente os valores médios na linha "Outros". Ao todo, foram usados 29 atributos, sendo apenas 12 comuns a todos os sites.

Atributo	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*Média*
QUARTO	0,995	0,994	0,993	0,976	1,0	0,997	0,994	0,993
SALA	0,980	1,00	0,979	0,987	0,985	0,994	0,977	0,986
FONE	0,992	0,995	0,998	0,998	0,981	0,993	0,998	0,994
VALOR	0,994	0,975	1,00	0,976	0,977	0,989	0,997	0,987
COZINHA	1,00	1,00	1,00	0,990	1,00	0,988	1,00	0,997
BANHEIRO	0,997	0,987	0,979	0,990	1,00	0,970	0,980	0,986
SUITE	1,00	1,00	0,991	0,985	1,00	1,00	1,00	0,996
TIPO	0,964	0,974	1,00	0,959	0,978	0,949	1,00	0,975
COPA	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
GARAGEM	1,00	0,987	1,00	0,981	0,983	0,981	1,00	0,990
PISCINA	0,952	1,00	1,00	1,00	1,00	1,00	1,00	0,993
VAGA	1,00	0,968	1,00	0,952	1,00	0,968	0,985	0,982
Outros	0,994	0,987	0,998	0,986	0,989	0,962	0,995	0,987

Tabela 3. Médias por site dos valores da medida F, em percentual, para a delimitação dos atributos dos anúncios.

Para esta análise, utilizamos a *medida F* [Baeza-Yates e Ribeiro-Neto 1999] (*F-measure*), bastante utilizada em trabalhos na área de Recuperação de Informação. Seja A_i um atributo, S_i o conjunto de termos que compõem A_i e T_i o conjunto de termos identificados como pertencentes a A_i e que, além disso, foram corretamente identificados como um dado ou um metadado. A medida F é definida como:

$$F_i = \frac{2(R_i.P_i)}{(R_i + P_i)}, \quad \text{onde } R_i = \frac{|S_i \cap T_i|}{|S_i|} \quad \text{e} \quad P_i = \frac{|S_i \cap T_i|}{|T_i|} \tag{8}$$

onde R_i e P_i são medidas chamadas respectivamente de *revocação* e *precisão*.

Os resultados apresentados na Tabela 3 correspondem à média dos valores da medida F de todos os atributos dos anúncios de cada site. Estes resultados dizem respeito

principalmente à eficácia da estrutura aninhada proposta, onde o HMM principal é responsável por identificar corretamente os atributos e os HMMs internos são responsáveis em delimitar corretamente os termos de cada atributo. Nota-se novamente um excelente resultado, com valores médios acima de 0,97 para todos os atributos.

5.3. Extração de Anúncios

Para avaliar os resultados da aplicação de nosso método sobre os anúncios, apresentamos na Tabela 4 a análise dos resultados da extração de cada anúncio tomado como uma unidade (ou seja, como um registro ou tupla) sob duas perspectivas.

F	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*Média*
Média Atributos	0,994	0,988	0,998	0,984	0,993	0,993	0,992	0,992
Anúncios	0,992	0,986	0,994	0,976	0,985	0,988	0,990	0,987

Tabela 4. Médias por site dos valores da medida F, em percentual, para extrações de anúncios.

A primeira perspectiva é apresentada na linha "Média Atributos", onde temos a média dos resultados obtidos para cada anúncio. Este resultado individual consiste na média dos valores da medida F para os atributos de cada anúncio. A medida F é calculada como descrito anteriormente pela Equação 8. A segunda perspectiva é apresenta na linha "Anúncios", onde os resultados equivalem a média dos valores da medida F calculados para cada anúncio de um site. Neste caso, a medida F também é calculada pela Equação 8, observado o seguinte: seja A_i um anúncio, S_i é o conjunto de *atributos* que compõem A_i e T_i é o conjunto de *atributos* extraídos pelos HMMs de dois níveis como pertencentes ao *anúncio* A_i.

Note que, enquanto a primeira perspectiva está relacionada à distribuição dos possíveis erros entre os atributos de um mesmo anúncio, a segunda perspectiva está relacionada ao comportamento do método para os anúncios em si. Em ambos os casos, temos novamente excelentes resultados com médias acima de 0,98.

6. Conclusões e Trabalhos Futuros

Apresentamos neste artigo uma nova formulação para o problema de extração em texto semi-estruturado, a qual considera a extração dos metadados existentes no texto e não somente os dados propriamente ditos. Para lidar com este problema, propusemos uma nova abordagem que é baseada em Modelos de Markov Ocultos (Hidden Markov Models - HMM). Esta abordagem utiliza uma estrutura aninhada de HMMs, onde um HMM principal identifica os atributos implícitos ocorrendo no texto alvo, e um conjunto de HMMs internos, uma para cada atributo, identifica os dados e metadados de cada atributo individualmente. A abordagem proposta foi implementada, e com esta implementação foram executados experimentos sobre sete conjuntos distintos de anúncios de classificados de imóveis coletados da Web. Os experimentos mostram uma precisão média superior a 97% na identificação de dados e metadados dos atributos, realizada pelos HMMs internos. A qualidade da extração de atributos, que é realizada pelos HMMs principais, foi em média superior a 0,97, usando a medida F. Estes resultados são refletidos na eficácia da extração de anúncios, que obteve média acima de 0,98, também usando medida F. Salienta-se que estes excelentes resultados foram obtidos mesmo usando poucos exemplos no treinamento, sendo 20% do número de anúncios extraídos. Isso pode ser explicado em parte pela ênfase que é dada à identificação de metadados no nosso método.

A estratégia de suavização apresentada na Seção 4.4 é adotada para flexibilizar os parâmetros do modelo e permitir a generalização dos exemplos apresentados de um site. No entanto, não está ainda claro se esta estratégia também permite a extração em um site, a partir de exemplos de outros sites do mesmo domínio. A investigação desta questão será objeto de um dos nossos trabalhos futuros. Como outros trabalhos futuros temos a aplicação do modelo proposto em outros tipos de texto semi-estruturado e a investigação de maneiras para aumentar a precisão na detecção de metadados considerando que as instâncias destes metadados ocorrem com grande frequência nos textos na forma de palavras-chave similares.

Agradecimentos

Este trabalho é parcialmente financiado pelos Projetos Tamanduá (MCT/FINEP/CT-INFO-Grade-01/2004), Gerindo (MCT/CNPq/CT-INFO 552.087/02-5) e SIRIAA (CNPq/ CT-Amazônia 55.3126/2005-9). Roberto Oliveira e Filipe Mesquita são bolsistas da CAPES e Altigran Silva é bolsista de produtividade do CNPq (Proc. 303032/2004-9). O presente trabalho foi realizado com o apoio do UOL (www.uol.com.br), através do Programa UOL Bolsa Pesquisa, processo número 20060520151215a.

Referências

Baeza-Yates, R. A. e Ribeiro-Neto, B. A. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley.

Borkar, V., Deshmukh, K., e Sarawagi, S. (2001). Automatic segmentation of text into structured records. *SIGMOD Record*, 30(2):175–186.

Califf, M. E. e Mooney, R. J. (1999). Relational learning of pattern-match rules for information extraction. In *Proceedings of the NCAI and Conference on IAAI*, páginas 328–334.

Embley, D. W. et al. (1999). Conceptual-model-based data extraction from multiple-record web pages. *Data & Knowledge Engineering*, 31(3):227–251.

Freitag, D. (2000). Machine learning for information extraction in informal domains. *Machine Learning*, 39(2/3):169–202.

Freitag, D. e McCallum, A. (2000). Information extraction with hmm structures learned by stochastic optimization. In *Proceedings of the NCAI and Conference IAAI*, páginas 584–589.

Laender, A. H. F. et al. (2002). A brief survey of web data extraction tools. *SIGMOD Record*, 31(2):84–93.

Mesquita, F. et al. (2006). LABRADOR: Efficiently publishing relational databases on the web by using keyword-based query interfaces. Em preparação.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recongnition. *Proceedings of the IEEE*, 77(2):257–286.

Viola, P. e Narasimhan, M. (2005). Learning to extract information from semi-structured text using a discriminative context free grammar. In *Proceedings of the ACM Conference on Research and development in information retrieval*, páginas 330–337.