Um Método Probabilístico para o Preenchimento Automático de Formulários Web a partir de Textos Ricos em Dados

Guilherme Alves Toda[1], **Altigran Soares da Silva (Orientador)**[1]

[1]Departamento de Ciência da Computação – Universidade Federal do Amazonas (UFAM)
Manaus-AM, Brasil
{gat,alti}@dcc.ufam.edu.br

Resumo. *Apresentamos neste artigo um novo método para preencher automaticamente formulários Web utilizando como entrada textos ricos em dados (por exemplo, um anúncio). A partir de tal entrada, nosso método identifica e extrai automaticamente dados de interesse nela contidos e os utiliza para preencher os campos apropriados do formulário. Para essa tarefa, utilizamos o conhecimento obtido a partir de valores utilizados previamente pelos usuários para preencher os formulários. Nosso método, chamado de* iForm*, utiliza características relacionadas ao conteúdo e ao estilo desses valores, que são combinadas através de uma Rede Bayesiana. Mostramos através de experimentos que o iForm atinge resultados superiores na comparação com o método que representa o estado da arte para o problema.*

Palavras-chave: *Formulários Web, Extração de Dados, Recuperação de Informação, Aprendizagem de Máquina.*

1. Introdução

Existem na Web várias aplicações onde usuários provêm dados para serem adicionadas a um banco de dados visando um processamento futuro. Este é o caso de bancos de currículos (p.ex., *CATHO*), bibliotecas digitais (p.ex., *DBLP*, *NDLDT*), sites de leilão (p.ex., *eBay*) e de eventos (p.ex., *DBWORLD*). A solução mais comum nesses casos é disponibilizar formulários compostos por vários campos de entrada de dados, como caixas de texto, caixas de seleção, listas de seleção, caixas de marcação, etc. Diferente dos formulários de busca, os formulários Web de entrada de dados normalmente apresentam um grande número de campos. Isso torna a tarefa de preenchimento por parte dos usuários trabalhosa e sujeita a erros. Este é um problema frequente em sites de comércio eletrônico como *eBay*, *amazon.com* e *TodaOferta.com* que utilizam formulários para usuários registrarem ofertas. Nesses sites podem existir diferentes formulários dependendo do produto oferecido, tornando ainda mais complicado o problema.

Neste artigo, propusemos, implementamos e avaliamos um novo método para resolver o problema de preencher automaticamente formulários de entrada de dados na Web. Nosso método, chamado de *iForm*, consiste em selecionar automaticamente segmentos de um texto rico em dados fornecido por um usuário, e preencher o formulário associando estes segmentos aos campos apropriados. Estes segmentos são identificados a partir do aprendizado de características relacionadas ao conteúdo e ao estilo dos valores previamente submetidos para cada campo, utilizando-se para isso um arcabouço probabilístico. Os usuários podem então verificar o preenchimento, realizar correções e então prosseguir com a sua submissão. Feito isso, os novos valores atribuídos a cada campo

são armazenados e usados como uma evidência extra quando novos textos de entrada são fornecidos.

Por se basear nas características dos campos do formulário e não dos textos de entrada, nosso método é bastante flexível, podendo lidar com textos de diversos estilos e estruturas. Para desenvolvedores, utilizar nosso método não requer nenhum esforço extra além de construir o formulário. Através da experimentação em um conjunto de dados reais, mostramos que nosso método é viável e efetivo, superando o estado da arte [T. Kristjansson et. al. 2004] encontrado na literatura para o problema.

O trabalho de mestrado aqui apresentado deu origem a um pôster apresentado em 2009 na principal conferência mundial da área de Web, a *International World Wide Web Conference* [G. A. Toda et. al. 2009] e a um artigo completo a ser apresentado em setembro de 2011 na principal conferência mundial da área de Bancos da Dados, a *Very Large Data Bases (VLDB) Conference*. O artigo já foi publicado em 2010 no periódico associado à conferência [G. A. Toda et. al. 2010].

2. Trabalhos Relacionados

Várias abordagens propostas na literatura recente têm tratado o problema de prover alternativas mais intuitivas que às interfaces baseadas em formulário para usuários interagirem com bancos de dados na Web. Essas soluções vão desde a estruturação de consultas baseadas em palavras-chave [F. Mesquita et. al. 2007] até o processamento de consultas em linguagem natural [Al-Muhammed and Embley 2007].

Especificamente para o problema de preenchimento automático de formulários Web, Kristjansson et. al. [T. Kristjansson et. al. 2004] propuseram um método baseado no modelo CRF (Condition Random Fields) [J. D. Lafferty et. al. 2001], atualmente o estado da arte em extração de dados de fontes textuais. O CRF se baseia em características do contexto em que ocorrem os valores a serem usados no preenchimento. Entre estas características estão o posicionamento e o sequenciamento dos valores, as quais são aprendidas por meio de um treino prévio sobre um conjunto representativo de textos de entrada manualmente rotulados. Na dissertação, este método foi experimentalmente comparado com o iForm.

3. O Método iForm

O método iForm consiste em estimar a probabilidade de um campo do formulário para cada segmento extraído do texto. Considere um texto de entrada I, composto por $N > 0$ símbolos (palavras). Seja S_{ab} um segmento, ou seja, uma sequência de símbolos em I que inclui os símbolos $t_a, t_{a+1}, \ldots, t_{b-1}, t_b$ ($0 < a \leq b \leq N$). Consideramos S_{ab} como um valor adequado para o campo F se a probabilidade do campo dado esse segmento é maior que um limiar ϵ. Para isso, consideramos que os segmentos tem um tamanho máximo L[1].

O princípio por trás de nosso método é utilizar as características dos valores usados anteriormente para preencher cada campo do formulário quando um novo texto é dado como entrada. Consideramos dois tipos de características: (1) os valores em si e os símbolos que compõem esses valores, as quais chamamos de *características de conteúdo*; e (2) o estilo, ou seja, o uso de letras maiúsculas ou minúsculas, pontuação, etc., que cha-

[1]Em nossos experimentos L não é maior que 10

mamos de *característica de estilo*. Vale ressaltar que nenhuma característica derivada do texto de entrada foi considerada em nosso método.

Com relação às características de conteúdo, consideramos que um segmento de texto pode ser usado para preencher um campo se seu conteúdo é similar ao conteúdo usado previamente para preencher esse campo. Para isso, consideramos tanto os símbolos que compõem o segmento quanto o segmento como um todo. Essas características são capturadas através de funções de probabilidade como descrito a seguir.

Calculamos a probabilidade de um campo F_j do formulário $\mathcal{F}$ dado um segmento S_{ab} com base nos símbolos que compõem o segmento utilizando a Eq.1,

$$TAF(F_j, S_{ab}) = \eta \sum_{\tau \in tokens(S_{ab})} \frac{\texttt{freq}(\tau, F_j)}{\sum_{F_i \in \mathcal{F}} \texttt{freq}(\tau, F_i)} \quad (1) \qquad \eta = \frac{1}{k + |avg(F_j) - k|} \quad (2)$$

onde a função $tokens(S_{ab})$ retorna o conjunto de símbolos em S_{ab}, F_j é um campo em $\mathcal{F}$ e $\texttt{freq}(\tau, F_i)$ retorna a frequência do símbolo τ em um campo F_i, considerando os valores submetidos anteriormente pelo usuário do formulário $\mathcal{F}$. Assim, calculamos a probabilidade entre cada símbolo do segmento e um campo F_j dividindo a frequência de τ nos valores submetidos pelos usuários a F_j pela frequência de τ em todos os campos do formulário. A intuição por trás dessa modelagem é que quanto mais concentradas forem as ocorrências anteriores de um símbolo em um campo, maior é a probabilidade do campo ser relacionado a tal símbolo. A constante η é um fator de normalização cujo valor é dado pela Eq.2 onde k é o número de símbolos em S_{ab} e $avg(F_j)$ é o número médio de símbolos nos valores submetidos como entrada para o campo F_j em interações anteriores do usuário com o formulário. Assim, segmentos com tamanho menor que o número médio de símbolos do campo são penalizados. Isso faz com que valores extraídos apresentem um tamanho compatível com os valores típicos de seus respectivos campos.

Calculamos também a probabilidade do segmento S_{ab} representar um valor de F_j com base na frequência do segmento inteiro nos campos, ao invés de comparar a frequência dos símbolos, dividindo a frequência de S_{ab} nos valores submetidos pelos usuários a F_j pela frequência de S_{ab} em todos os campos do formulário, conforme a equação abaixo.

$$VAF(F_j, S_{ab}) = \frac{\texttt{freq}(S_{ab}, F_j)}{\sum_{F_i \in \mathcal{F}} \texttt{freq}(S_{ab}, F_i)} \quad (3)$$

Utilizamos $TAF(F_j, S_{ab})$ e $VAF(F_j, S_{ab})$ para estimar a probabilidade de F_j dado S_{ab}, com base no *conteúdo* dos valores conhecidos de F. Esta probabilidade é combinada com a probabilidade de F_j dado S_{ab} com base nas características de estilo, descritas a seguir.

Com relação às características de conteúdo, consideramos como relevante para determinar quando um segmento pode ser usado para preencher um campo, a similaridade entre a forma como são escritos os símbolos do segmento e a forma como são escritos os valores submetidos ao campo. Seja SV_j o conjunto de valores submetidos anteriormente ao campo F_j. Para tratar as características relacionadas ao estilo, utilizamos um *Hidden Markov Model (HMM)*[V. Borkar et. al. 2001] $SM(F_j)$, chamado de *Modelo de Estilo de Valores* (MEV), que captura o estilo de escrita dos valores em SV_j. Para gerar um MEV, primeiro separamos cada valor de SV_j em símbolos (palavras) com

base nos espaços entre eles. Em seguida, utilizando uma taxonomia similar à proposta em [V. Borkar et. al. 2001], codificamos esses valores como sequências de máscaras, representadas por expressões regulares. Chamaremos de *Sequências de Máscaras de Símbolo* (SMS).

Um grafo representando o MEV $SM(F_j)$ é gerado com todas as SMS encontradas em valores submetidos anteriormente ao campo F_j. Em $SM(F_j)$, cada nó é representado por uma *máscara de símbolo* que ocorre em SV_j. As arestas são formadas por pares ordenados $\langle n_x, n_y \rangle$ tais que o nó n_x é seguido por n_y em alguma sequência de máscara de símbolos codificada nos valores SV_j. Assim, identificamos cada sequência de máscara de símbolos como um *caminho* em $SM(F_j)$. Além disso, o nó inicial do grafo contém arestas para os primeiros valores representados no grafo e o nó final é ligado por arestas que denotam a probabilidade de serem as últimas máscaras de seus respectivos valores.

Usando a abordagem de *probabilidade máxima* [V. Borkar et. al. 2001], o *peso* de cada aresta em $SM(F_j)$ é calculado como:

$$w(SM(F_j), n_x, n_y) = \frac{\text{\# de pares } \langle n_x, n_y \rangle \text{ em } SM(F_j)}{\text{\# de pares } \langle n_x, n_z \rangle, \forall n_z \in SM(F_j)}. \quad (4)$$

Portanto, o *peso* de uma aresta ligada aos nós n_x e n_y indica a probabilidade de uma *máscara de símbolo* n_x ocorrer seguida de uma máscara n_y.

Para realizar o casamento de um segmento S_{ab} com os valores de SV_j de acordo com o seu estilo de escrita, primeiro codificamos S_{ab} em uma SMS utilizando a taxonomia apresentada acima. Então, computamos a probabilidade da SMS de S_{ab} representar um *caminho* no grafo $SM(F_j)$, como:

$$style(SM(F_j), S_{ab}) = \frac{\sum_{\langle n_x, n_y \rangle \in caminho(S_{ab})} w(SM(F_j), n_x, n_y)}{\text{\# de arestas em } caminho(S_{ab})} \quad (5)$$

onde $caminho(S_{ab})$ retorna o caminho percorrido no MEV $SM(F_j)$ associado ao segmento S_{ab} correspondente à SMS de S_{ab}.

Modelamos a computação da probabilidade do campo F_j dado o segmento S_{ab}, $P(F_j|S_{ab})$, por meio de um modelo de Rede de Crença Bayesiana que é usada para combinar $TAF(F_j, S_{ab})$, $VAF(F_j, S_{ab})$ e $style(SM(F_j), S_{ab})$. Por questões de espaço, omitimos aqui os detalhes sobre esta rede, os quais são discutidos em profundidade no texto da dissertação.

Seja C_j o conjunto de segmentos S_{ab} tal que $P(F_j|S_{ab})$ é maior que o limiar ϵ. Dizemos que C_j é um conjunto de *valores candidatos* para o campo F_j. O objetivo é encontrar um mapeamento $\mathcal{M}$ entre os valores candidatos e os campos no formulário tal que a probabilidade agregada tenha o valor máximo, satisfazendo as seguintes condições: (1) apenas um segmento é associado a cada campo e (2) os segmentos selecionados não sofrem sobreposição, isto é, nenhum símbolo t_i pode ser utilizado para preencher dois ou mais campos. Isto pode ser feito seguindo o procedimento de duas fases descrito a seguir.

Na primeira fase, iniciamos pelo cálculo dos valores candidatos para cada campo F_j, baseado apenas nas características relacionadas ao conteúdo. Seja $\mathcal{I}$ um conjunto composto pela união dos conjuntos dos valores candidatos C_j para todos os campos F_j.

Referimos a $\mathcal{I}$ como o *mapeamento inicial*, o qual contém os pares segmento-campo $\langle S_{ab}, F_j \rangle$. Assumimos que dois pares em $\mathcal{I}$ são *conflitantes* se eles violam alguma das condições acima.

Para encontrar a solução ótima precisamos encontrar todos os possíveis subconjuntos – um número exponencial. Na prática, utilizamos uma heurística simples para encontrar um solução aproximada. Primeiro, extraímos o par com a maior probabilidade de $\mathcal{I}$ e verificamos se ela apresenta conflito com algum par em $\mathcal{I}$. Se o par não for conflitante, o adicionamos ao mapeamento final. Repetimos esse processo até que todos os pares em $\mathcal{I}$ forem extraídos. Isso marca o fim da primeira fase.

Na segunda fase, se algum campo continua não mapeado por um segmento, utilizamos as probabilidades derivadas das características relacionadas ao estilo para tentar encontrar outras associações e assim calcular a probabilidade de cada campo dado um segmento. Então repetimos o processo de mapeamento, agora considerando apenas os pares dos segmentos e os campos que não foram mapeados na primeira fase.

4. Experimentos

Para verificar a eficácia do iForm, foram executados diversos experimentos com o método utilizando dados reais de vários domínios diferentes. Neste artigo nos limitamos a apresentar os resultados de um destes experimentos, onde fazemos uma avaliação comparativa com o método proposto em [T. Kristjansson et. al. 2004]. Como já mencionado, este método é baseado no modelo CRF [T. Kristjansson et. al. 2004], estado da arte em extração dados. Implementamos esse método a partir de uma publicação genérica do CRF [2] que utiliza as mesmas características descritas em [J. D. Lafferty et. al. 2001]: características de dicionário, função de cálculo da média das palavras e características de transição. Utilizamos para o experimentos um conjunto de 150 anúncios de emprego disponíveis na coleção RISE[3], onde os segmentos a serem extraídos se encontram manualmente rotulados. Separamos aleatoriamente um subconjunto de 100 destes anúncios para treinar o CRF e simular submissões anteriores no iForm. Um outro subconjunto disjunto de 50 anúncios foi usado para executar os testes. Esse processo foi repetido 5 vezes. Esses resultados estão na Tabela 1 mostrando os valores da medida-F por campo, considerando a média das 5 execuções.

Campo	iForm	CRF	Campo	iForm	CRF
Estado	**0,85**	0,81	*Formação Desejada*	**0,57**	0,37
Cidade	**0,70**	0,65	*Aplicação*	**0,82**	0,37
Linguagem	**0,84**	0,69	*Área*	0,18	**0,23**
País	0,77	**0,87**	*Empregador*	**0,44**	0,22
Formação Necessária	0,31	**0,59**	*Empresa*	**0,41**	0,17
Plataforma	**0,47**	0,38	*Salário*	0,22	**0,25**

Tabela 1. Comparação de resultados por campo

O iForm obteve valores de medida-F superiores em nove campos, enquanto que o método baseado em CRF superou o iForm em apenas quatro campos. Estes valores estão indicados na tabela pelos números em negrito. A baixa qualidade dos resultados

[2] `http://crf.sourceforge.net/`
[3] `http://www.isi.edu/info-agents/RISE/index.html`

obtidos com CRF pode ser explicada pelo fato dos segmentos a serem extraídos de textos de entrada típicos, como anúncios de emprego, podem não aparecer em um contexto regular, o que é um requisito importante para o CRF. Para o caso do iForm, esse contexto é menos importante, pois nosso método depende apenas das características relacionados aos *campos* ao invés de depender das características dos textos de entrada. Além disso, o iForm foi desenvolvido para explorar essas características relacionadas ao campo das submissões anteriores. Note-se que para aplicar o CRF a esse problema, um treinamento intensivo dos dados de exemplos representativos de texto de entrada é necessário.

5. Conclusões

Neste artigo apresentamos um método chamado iForm para o problema de preenchimento automático de formulários Web a partir de segmentos extraídos automaticamente de textos ricos em dados. Realizamos uma avaliação experimental que mostram a eficácia do iForm. Mostramos que o método obtêm melhores resultados que um método baseado no modelo CRF, o estado da arte em extração de dados, sendo mais flexível e exigindo menos intervenção por parte do usuário.

Como trabalhos futuros, pretendemos investigar o problema de ajudar usuários a encontrar entre um grande conjuntos de formulários, quais deles são mais apropriados para serem preenchidos dado um texto de entrada. Esse problema necessita de uma solução eficiente e escalável para o cenário da Web. Nesse cenário, pretendemos estender nosso método para lidar também com formulários de consulta. Atualmente, o iForm assume que os valores do texto de entrada são indicações positivas para preencher o formulário. Assim, pretendemos estender nosso método para também lidar com eventuais indicações negativas nos textos de entradas.

Referências

Al-Muhammed, M. and Embley, D. W. (2007). Ontology-based constraint recognition for free-form service requests. In *Proc. of the 23rd Intl. Conf. on Data Engineering*, pages 366–375.

F. Mesquita et. al. (2007). LABRADOR: Efficiently publishing relational databases on the web by using keyword-based query interfaces. *Inform. Proc. and Management*, 43(4):983–1004.

G. A. Toda et. al. (2009). Automatically filling form-based web interfaces with free text inputs. In *Proc. of the 18th Intl. Conf. on World wide web*, pages 1163–1164.

G. A. Toda et. al. (2010). A probabilistic approach for automatically filling form-based web interfaces. *Proc. VLDB Endow.*, 4(3):151–160.

J. D. Lafferty et. al. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. 8th Intl. Conf. on Mach. Learning*, pages 282–289.

T. Kristjansson et. al. (2004). Interactive information extraction with constrained conditional random fields. In *Proc. of the 19th Nat. Conf. on Artifical intelligence*, pages 412–418.

V. Borkar et. al. (2001). Automatic Segmentation of Text into Structured Records. In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pages 175–186.