Uma Abordagem de Aprendizagem Híbrida para Extração de Informação em Textos Semi-Estruturados

Eduardo F.A. Silva, Flávia A. Barros & Ricardo B. C. Prudêncio
Email: [efas, fab, rbcp]@cin.ufpe.br
Centro de Informática - Universidade Federal de Pernambuco
Cx. Postal 7851 – CEP 50732-970 Recife (PE) - Brasil
Tel.: +55 81 32718430 Fax: +55 81 32718438

Abstract. *Information Extraction (IE) aims to extract from textual documents only the relevant data. This paper presents a hybrid approach to the problem of IE that combines text classification techniques and Hidden Markov Models (HMM). The IE system generates an initial output using text classification techniques, which is refined by a HMM. The implemented prototype was used to extract information from bibliographic references, reaching a precision rate of 87,48% in a test set of 3000 references.*

Resumo. *A Extração de Informação (EI) tem por objetivo extrair de documentos textuais apenas os dados relevantes ao usuário. Este artigo propõe uma abordagem híbrida para EI que combina classificadores de texto e Modelos de Markov Escondidos (HMM). O classificador de texto gera uma saída inicial, que é refinada por meio de um HMM. O protótipo implementado foi usado na tarefa de EI em referências bibliográficas e apresentou uma precisão de até 87,48% para um conjunto de teste com 3000 referências.*

1. Introdução

A maior parte da informação armazenada nos repositórios digitais (e.g., a Web e as Bibliotecas Digitais) encontra-se na forma de documentos textuais. Considerando o imenso volume de informação disponível nesses repositórios, é de grande interesse a construção de sistemas capazes de selecionar automaticamente apenas os dados de interesse de um usuário, facilitando assim o acesso e a manipulação dessas informações.

A Extração de Informação (EI) tem por objetivo extrair, de documentos textuais, apenas os dados relevantes ao usuário [Kushmerick & Thomas 2003]. Sistemas de EI identificam trechos dos documentos que preenchem corretamente campos de um formulário de saída que determina os dados a serem extraídos. As abordagens comumente utilizadas na construção de sistemas de EI incluem o uso do processamento de linguagem natural (para tratar textos livres), a engenharia do conhecimento e a aprendizagem automática (mais adequadas a textos estruturados ou semi-estruturados).

Escolhemos como foco deste trabalho a EI a partir de textos semi-estruturados, amplamente disponíveis na Web. Dentre as diferentes abordagens de EI para tratar esse tipo de texto, destacamos o uso de algoritmos de aprendizagem para classificação de texto [Kushmerick et al 2001], por facilitarem a adaptação dos sistemas a novos domínios. Aqui, o documento é inicialmente dividido em fragmentos candidatos a preencher algum campo do formulário de saída. Em seguida, um algoritmo de aprendizagem determina a que campo do formulário cada fragmento corresponde. A

maior limitação desses sistemas é realizarem uma classificação local independente para cada fragmento, perdendo informações estruturais importantes do documento.

Propomos aqui uma abordagem de aprendizagem híbrida, original para EI, que combina classificadores de texto com os Modelos de Markov Escondidos (*Hidden Markov Models* - HMM) [Rabiner & Juang 1986]. Nesta abordagem, um algoritmo de aprendizagem gera uma classificação inicial dos fragmentos do texto de entrada, que é refinada por meio de um HMM, gerando uma nova classificação ótima global para todos os fragmentos do texto.

Como estudo de caso, implementamos um protótipo para o domínio de referências bibliográficas, que se apresentam como textos semi-estruturados com uma grande variação na sua estrutura, o que torna bastante difícil a EI neste domínio [Borkar et al 2001]. Nosso objetivo é extrair informações como autor, título, data de publicação, etc, a fim possibilitar a criação e a manutenção automática de grandes bases de dados sobre produção acadêmica/científica, muito úteis para a pesquisa científica.

Os diversos experimentos realizados para avaliar o protótipo revelaram um ganho consistente de desempenho com o uso do HMM, variando de 1,27 até 22,54 pontos percentuais, dependendo do classificador e do conjunto de características utilizado na geração da saída inicial. O melhor desempenho alcançado obteve precisão de 87,48% para um conjunto de 3000 referências de teste.

A próxima seção discute técnicas e sistemas para a EI. A seção 3 descreve a abordagem proposta, que combina um classificador de texto e um HMM. A seção 4 apresenta os experimentos realizados, e a seção 5 traz considerações finais sobre o trabalho, suas contribuições e indicação de trabalhos futuros.

2. Extração de Informação – Técnicas e Sistemas

Extração de Informação (EI) envolve povoar uma base de dados com valores automaticamente extraídos a partir de documentos textuais [Kushmerick & Thomas 2003]. Um sistema de EI tem por objetivo identificar trechos dos documentos que preenchem corretamente campos (*slots*) de um dado formulário (*template*) de saída, que determinam as informações que devem ser extraídas.

Na área de EI, os textos podem ser classificados como estruturados, semi-estruturados e não-estruturados (ou livres). Um texto *estruturado* segue um formato rígido (e.g., páginas HTML geradas a partir de bancos de dados), o que possibilita que a informação seja extraída usando regras baseadas em delimitadores e/ou na ocorrência de termos. Os *textos livres* contêm, basicamente, sentenças em alguma língua natural, o que inviabiliza a extração com base apenas em formatação. Textos *semi-estruturados*, por sua vez, apresentam algum grau de estruturação (e.g., referências bibliográficas), juntamente com irregularidades, como campos ausentes ou com valor nulo, variações na ordem dos dados, e ausências de delimitadores entre as informações a serem extraídas.

O tipo do texto sendo tratado exerce grande influência na construção do sistema de EI. Técnicas de Processamento de Linguagem Natural são comumente usadas para tratar *textos livres*, uma vez que são capazes de lidar com as irregularidades das línguas naturais [Appelt & Israel 1999]. Esses sistemas realizam uma análise sintática e semântica nem sempre possível em textos estruturados ou semi-estruturados, como é o caso de muitos dos textos (sites) na Web, que não seguem a gramática da língua.

Por outro lado, técnicas de Inteligência Artificial têm sido largamente usadas para EI em textos *estruturados* e *semi-estruturados*. Como exemplo, podemos citar os sistemas baseados em regras de extração construídas manualmente através de engenharia de conhecimento [Nunes & Barros 2000]. Embora apresentem bons resultados, esses sistemas requerem uma grande quantidade de trabalho manual e a existência de bons especialistas, não sendo facilmente adaptáveis a novos domínios.

Visando minimizar essas dificuldades, diferentes autores usam algoritmos de aprendizado de máquina para construir regras de extração de forma automática, a partir de um corpus de documentos etiquetados [Appelt & Israel 1999]. Segundo [Soderland 1999], a aprendizagem automática possibilita uma adaptação mais rápida e eficiente dos sistemas de EI para novos domínios de aplicação.

Destacamos inicialmente dois tipos de sistemas de aprendizagem utilizados na EI: os baseados em autômatos finitos [Kushmerick et al 1997], [Hsu & Dung 1998], [Kosala et al 2002] e os baseados em casamento de padrões [Soderland 1999],[Califf & Mooney 1999]. O primeiro tipo aprende regras de extração na forma de autômatos finitos que definem: (1) estados que "aceitam" os símbolos do texto que preenchem algum campo do formulário de saída, (2) os estados que apenas consomem os símbolos irrelevantes encontrados no texto, e (3) os símbolos que provocam as transições de estado. Já os sistemas baseados em casamento de padrões aprendem regras na forma de expressões regulares. Os dois tipos de sistemas estão intimamente relacionados, uma vez que as linguagens aceitas pelos autômatos finitos podem ser também descritas através de expressões regulares [Hopcroft & Ullman 1979].

Uma das grandes vantagens dos sistemas citados acima é que eles representam suas regras em linguagens simbólicas, de mais fácil interpretação. Contudo, esses sistemas requerem que as informações a serem extraídas apresentem padrões regulares ou delimitadores que determinem os campos que elas deverão preencher [Soderland 1999]. Dessa forma, eles são menos adequados para textos como as referências bibliográficas, que apresentam uma grande variação de estrutura [Borkar et al 2001].

Trabalhos mais recentes realizam EI através da aplicação de algoritmos de aprendizagem para classificação de textos [Kushmerick et al 2001], [Bouckaert 2002]. Inicialmente, tais sistemas dividem o texto de entrada em fragmentos candidatos a preencher algum campo do formulário de saída. Em seguida, algoritmos de aprendizagem classificam os fragmentos com base em suas características (e.g., número de palavras, presença de palavras específicas, letras capitalizadas, ...). A maior limitação desses sistemas é o fato de realizarem uma classificação local independente para cada fragmento, perdendo uma importante informação estrutural presente no documento.

Destacamos ainda a técnica de EI baseada nos Modelos de Markov Escondidos (HMM) [Kushmerick et al 2001], [Borkar et al 2001]. Nessa modelagem, um estado oculto é criado para cada campo de saída, e os símbolos emitidos pelos estados ocultos são definidos como os *tokens* do documento (i.e., palavras, números, pontuação, etc). Dada uma seqüência de *tokens*, o HMM determina os estados ocultos associados a cada um desses símbolos, ou seja, que campo de saída cada *token* deverá preencher. O HMM tem a vantagem de realizar uma classificação ótima para a seqüência completa de entrada. Por outro lado, ele não é capaz de fazer uso de múltiplas características dos *tokens* (por exemplo, formatação, tamanho e posição), como ocorre nos classificadores.

3. Um Sistema Híbrido para EI em Referências Bibliográficas

Este trabalho propõe uma abordagem híbrida para EI que combina as técnicas de classificação de texto e os HMM para EI a partir de textos semi-estruturados. Segundo Kushmerick et al (2001), essas técnicas são adequadas para tratar textos semi-estruturados, por serem capazes de lidar com variações na estrutura do texto. Além disso, ambas usam algoritmos de aprendizagem convencionais, o que não é o caso de muitos dos sistemas baseados em autômatos finitos e casamento de padrões, que usam algoritmos complexos e específicos para EI.

A idéia básica é fazer uma extração inicial de dados usando um classificador de texto, e em seguida refiná-la por meio de um HMM. Ao combinar as duas técnicas, obtivemos os benefícios de ambas, e superamos suas limitações individuais. Os classificadores apresentam uma classificação ótima localmente (i.e., para cada fragmento), enquanto que os HMM oferecem uma classificação ótima global para todos os fragmentos de entrada. Os HMM, por sua vez, não tratam múltiplas características dos fragmentos, o que é superado pelo uso do classificador. Essa combinação é original na área, e revelou resultados experimentais muito satisfatórios (ver seção 4).

O problema escolhido como estudo de caso foi o de EI em referências bibliográficas, objetivando a criação automática de bases de dados sobre a produção dos pesquisadores, que podem servir como ferramentas úteis para a pesquisa e a comunicação científica. A partir de uma referência, podem ser extraídas diversas informações, como autor, título do trabalho, local e data de publicação (Figura 1).

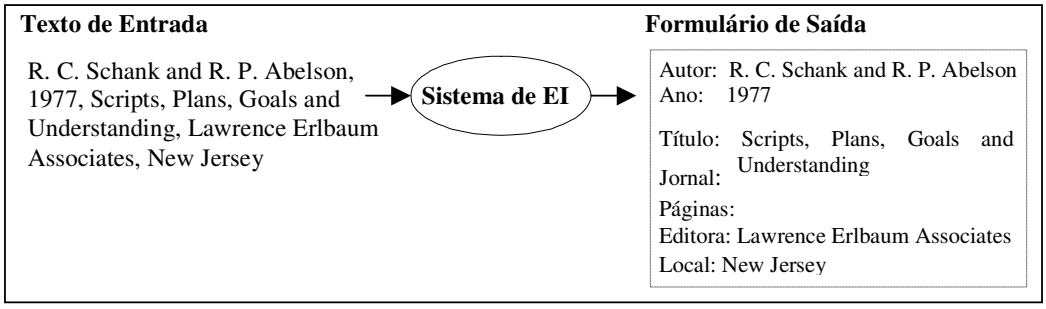

Figura 1. Exemplo de extração de informações em uma referência bibliográfica

Referências bibliográficas consistem em texto semi-estruturado cujo formato apresenta uma grande variação na sua estrutura, o que torna a extração de informação neste domínio uma tarefa bastante árdua [Borkar et al 2001]. Como exemplo, citamos: (1) *Ordem variável dos campos (e.g.,* autor pode ser o 1º ou o 2º campo); (2) *Campos ausentes (e.g.,* as páginas de um artigo são, muitas vezes, omitidas); (3) *Estilo telegráfico (e.g.,* páginas (pp)); (4) *Ausência de delimitadores precisos* (alguns delimitadores, como "," e "." podem aparecer no meio de um campo a ser extraído).

A seguir, apresentamos detalhes sobre as duas fases do processo de extração na abordagem proposta: extração inicial utilizando as técnicas de classificação para EI (seção 3.1), e refinamento da saída da fase 1 utilizando um HMM (seção 3.2).

3.1. Fase 1 – Extração Usando Técnicas de Classificação

O processo de extração de informação através dos classificadores realizado na fase 1 do sistema é dividido em 3 etapas, descritas a seguir.

1- *Fragmentação do texto de entrada*: o texto deve ser dividido em fragmentos candidatos a preencher algum campo do formulário de saída (figura 2). Para isso, foram utilizados os delimitadores do texto, como virgulas, pontuações e outros marcadores;

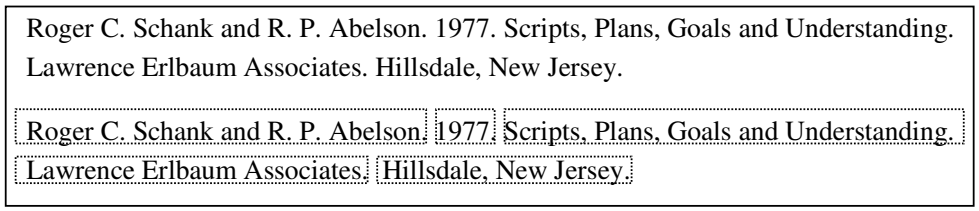

Figura 2. Exemplo simplificado de fragmentação de um texto de referência

2- *Extração de características*: para cada fragmento do texto, é criado um vetor de características, utilizado para a classificação do fragmento. Nesse trabalho, utilizamos três conjuntos distintos de características. Dois deles definem características específicas para o domínio de referências bibliográficas (o conjunto definido no sistema Prodext [Nunes & Barros 2000], e o definido por Bouckaert et al (2002)). Esses dois conjuntos foram definidos através de um processo de engenharia de conhecimento, e contêm características como presença de termos específicos (e.g., "vol", "conf." e "pp"), nomes de uma lista de editoras, datas em formato de ano, entre outras. O terceiro conjunto é formado por palavras extraídas diretamente do corpus de treinamento através do método de seleção de características *Information Gain* [Yang & Pedersen 1997].

3- *Classificação dos fragmentos*: as características de cada fragmento são apresentadas a um classificador, que decide que campo de saída o fragmento deve preencher, dentre 14 campos[1]: *Autor*, *Título*, *Instituição*, *Jornal*, *Veículo*, *Mês*, *Ano*, *Editor*, *Local*, *Editora*, *Volume*, *Número*, *Páginas e Outros*. O classificador é construído através de um processo de aprendizado que usa como exemplos de treinamento um conjunto de fragmentos etiquetados manualmente. Nessa etapa, testamos três classificadores, cada um representando uma família de algoritmos de aprendizado: o *Naive Bayes* [John & Langley 1995], o algoritmo PART (Rules) [Frank & Witten 1998] e um classificador baseado em instâncias, o kNN [Aha & Kibler 1991]. Todos foram implementados usando a API da ferramenta WEKA [Witten & Frank 1999].

3.2. Fase 2 - Refinamento dos Resultados com um HMM

A fase 1 realiza a classificação de cada fragmento de forma independente dos demais. Contudo, as informações a serem extraídas das referências aparecem em uma certa ordem que, apesar de não ser fixa, pode ser usada no processo de extração, a fim de se obter uma classificação global ótima dos seus fragmentos. A classificação da fase 1 serve como entrada para o HMM, que gera uma classificação mais refinada para os fragmentos com o objetivo de corrigir os eventuais erros de classificação inicial.

Nossa abordagem tem forte relação com a técnica de *Stacking* [Wolpert 1992], que consiste em treinar um novo classificador a partir da saída de outros classificadores, como uma forma de meta-aprendizagem. No nosso caso, entretanto, a idéia não é combinar a saída de vários classificadores, mas sim usar o HMM para refinar a classificação realizada por um único classificador para os diversos fragmentos do texto.

[1] Campos comumente definidos pelo formato BibTex, usado para referências bibliográficas em documentos LATEX. Mais detalhes em: http://www.ecst.csuchico.edu/~jacobsd/bib/formats/bibtex.html

Os HMM foram escolhidos por serem capazes de classificar uma seqüência inteira de símbolos de entrada (fragmentos de texto), buscando maximizar o acerto global.

Um HMM é um autômato finito probabilístico que consiste em: (1) Um conjunto de estados ocultos *S*; (2) Uma probabilidade de transição *Pr[s'/s]* entre os estados ocultos $s \in S$ e $s' \in S$; (3) Um conjunto de símbolos *T* emitidos pelos estados ocultos; e (4) Uma distribuição de probabilidade *Pr[t/s]* de emissão de cada símbolo $t \in T$ para cada estado escondido $s \in S$. O algoritmo Viterbi [Rabiner & Juang 1986] é usado no processo de classificação, retornando a seqüência de estados ocultos com maior probabilidade de ter emitido cada seqüência de símbolos de entrada.

Neste trabalho, a estrutura do HMM foi definida com um estado oculto para cada campo do formulário de saída. Todos os estados ocultos foram conectados entre si, uma vez que a determinação da classe de cada campo do formulário pode estar relacionada com a classe dos demais campos.

O HMM recebe como seqüência de símbolos de entrada as classes associadas aos fragmentos do texto na fase 1, e retorna a seqüência mais provável de campos do formulário de saída (i.e., estados ocultos) para a seqüência dada. A Figura 3 mostra um HMM simplificado, contendo 3 símbolos, identificados pelo prefixo "S_", e 3 estados ocultos, identificados pelo nome do campo ao qual eles estão associados.

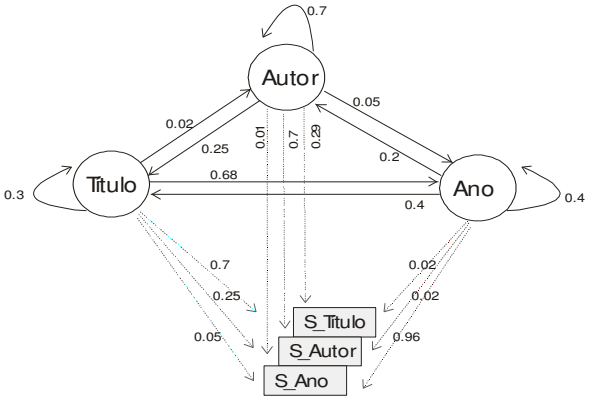

Figura 3. Exemplo simplificado de um HMM usado na Fase 2 do sistema proposto

Cada exemplo de treinamento do HMM consiste em uma lista de pares formados por um símbolo e o estado oculto que o emitiu, ou seja, pela classe encontrada na fase 1 para o fragmento e a classe real à qual ele pertence. No exemplo 1 da figura 4, o segundo fragmento foi classificado pela fase 1 como *Jornal*, porém sua classe real é *Título*. No exemplo 2, todos os fragmentos foram classificados corretamente, e no exemplo 3, o terceiro fragmento foi classificado como *Autor*, ao invés de *Editor*.

1- { (s_autor ,autor), (s_ jornal, título), (s_veículo, veículo), (s_ano, ano)}
2- { (s_autor, autor), (s_ título, título), (s_ano, ano), (s_local, local)}
3- { (s_autor, autor), (s_título, título), (s_autor, editor), (s_ano, ano)}

Figura 4. Exemplos de seqüências de treinamento do HMM

A probabilidade *Pr[s'/s]* de transição entre os estados ocultos e a probabilidade de *Pr[t/s]* de emissão dos símbolos pelos estados foram estimadas partir das seqüências de treinamento utilizando-se as equações definidas por Borkar et al (2001):

$$\Pr(s',s) = \frac{\text{Número de transições do estado } s' \text{ para o } s}{\text{Número total de transições a partir do estado } s'} \qquad \Pr(t,s) = \frac{\text{Número de emissões do símbolo } t \text{ pelo estado } s}{\text{Número total de símblos emitidos pelo estado } s}$$

4. Experimentos e Resultados Obtidos

Foi utilizada para treinamento e teste do sistema a coleção *Bibliography on computational linguistics, systemic and functional linguistics, Artificial intelligence and general linguistics*[2], que contém 6 mil referências bibliográficas com *tags* que indicam a classe à qual cada fragmento do texto pertence. O número médio de campos por referência é de 6,22, mostrando que nem sempre os 14 campos do formulário de saída estão presentes nas referências (sendo os mais freqüentes *Autor, Título* e *Ano*). O conjunto de referências foi dividido igualmente em 3000 referências para treinamento do sistema, e 3000 referências para avaliação do seu desempenho.

4.1 Descrição dos Experimentos

Os experimentos realizados avaliaram o desempenho obtido pelo sistema com e sem o uso do HMM, variando ainda os seguintes fatores: (1) o conjunto de características; e (2) o classificador utilizado na fase 1. Como dito na seção 3.2, os classificadores testados foram o *Naive Bayes*, *PART (Rules)* e o *kNN*.

Foram testadas 6 combinações dos conjuntos de características citados na seção 3.1: (1) *Manual1* (20 características usadas no sistema Prodext); (2) *Manual2* (9 características definidas por Bouckaert (2002)); (3) *Automático* (100 palavras selecionadas do conjunto de treinamento); (4) *Manual1+ Manual2* (totalizando 27 características); (5) *Automático + Manual2* (totalizando 109 características); e (6) *Automático + Manual2 + Manual1* (totalizando 127 características).

Cada combinação dessas representa um nível diferente de esforço de um especialista na sua criação. A combinação *Manual1+Manual2*, por exemplo, representa o máximo de esforço do especialista, pois combina dois conjuntos manualmente definidos. Por outro lado, o conjunto *Automático+Manual2* é formado pela união de um conjunto automaticamente definido com um manualmente definido, representando um esforço intermediário. Destacamos ainda que a combinação de características manualmente definidas com outras automaticamente selecionadas não foi observada em nenhum dos trabalhos anteriores de EI para referências bibliográficas.

O desempenho do sistema foi avaliado com e sem o uso do HMM para cada combinação *conjunto de características X algoritmo de classificação*. A medida de avaliação usada foi a precisão, calculada como a quantidade de campos corretamente extraídos dividido pelo número total de campos presentes na referência.

4.2 Resultados obtidos

A Tabela 1 mostra a precisão média por referência obtida pelo sistema com todas as combinações de características e classificadores testados. Comparando os valores de precisão obtidos com e sem o HMM, verificamos um ganho de desempenho com o uso do HMM para todas as combinações, variando de 1,27 até 22,54 pontos percentuais. O melhor resultado foi de 87,48% de precisão, obtido com o conjunto *Automático+Manual2*, o classificador PART e o refinamento com o HMM.

[2] Acesso on-line em http://liinwww.ira.uka.de/bibliography/Ai/bateman.html.

Tabela 1: Resultados obtidos para corpus de teste com 3.000 referências.

Conj. Características	Classificador	Precisão sem HMM	Precisão com HMM	Diferença Precisão
Manual1	PART	72,17%	76,40%	4,22%
Manual1	Bayes	66,70%	74,72%	8,01%
Manual1	kNN	71,96%	76,28%	4,32%
Manual2	PART	73,48%	77,29%	3,80%
Manual2	Bayes	69,03%	77,27%	8,23%
Manual2	kNN	76,17%	81,16%	4,99%
Automático	PART	49,91%	72,45%	22,54%
Automático	Bayes	50,11%	68,25%	18,14%
Automático	kNN	51,47%	73,57%	22,10%
Manual1+Manual2	PART	81,99%	86,00%	4,00%
Manual1+Manual2	Bayes	71,89%	81,43%	9,54%
Manual1+Manual2	kNN	81,40%	83,21%	1,81%
Automático+Manual2	PART	83,74%	**87,48%**	3,75%
Automático+Manual2	Bayes	74,78%	83,46%	8,69%
Automático+Manual2	kNN	83,23%	84,85%	1,62%
Automático+Manual2+Manual1	PART	84,82%	87,36%	2,54%
Automático+Manual2+Manual1	Bayes	75,29%	84,20%	8,90%
Automático+Manual2+Manual1	kNN	83,89%	85,17%	1,27%

O desempenho do sistema variou significativamente dependendo do algoritmo usado na fase 1. Note que, para todos os conjuntos de características, há uma piora de desempenho médio para o classificador Naive Bayes, especialmente sem o uso de HMM (ver tabela 2). Contudo, pode-se observar que o uso do HMM compensa o baixo desempenho desse classificador, aproximando-o do obtido pelos outros classificadores. Assim, concluímos que a variabilidade do desempenho do sistema em relação à escolha do classificador da fase 1 é menor quando o HMM é usado na fase de refinamento.

Tabela 2: Média da precisão obtida pelos classificadores com e sem HMM

Classificador	Média da Precisão sem HMM	Média da Precisão com HMM
PART	74,35%	81,16%
Bayes	67,97%	78,22%
kNN	74,69%	80,71%

O desempenho do sistema também variou significativamente dependendo do conjunto de características usado na fase 1. O conjunto *Automático* apresentou o pior desempenho médio de precisão (ver tabela 3). No entanto, existe uma diferença menos acentuada entre o desempenho do sistema para o conjunto *Automático* e os demais conjuntos quando se utiliza o HMM. Isso mostra que o refinamento do HMM parece compensar o uso de um conjunto de características menos expressivo, e torna viável inclusive o uso de conjuntos automaticamente definidos (o que facilita a adaptação do sistema para outros domínios de EI).

Tabela 3: Média da precisão obtida pelos conjuntos de características com e sem HMM

Conjunto de características	Média da Precisão sem HMM	Média da Precisão com HMM
Manual1	70,27%	75,80%
Manual2	72,89%	78,57%
Automático	50,49%	71,42%
Manual1+Manual2	78,42%	83,54%
Automático+Manual2	80,58%	85,26%
Automático+Manual2+Manual1	81,33%	85,57%

Por fim, citamos alguns trabalhos relacionados. Nunes & Barros (2000) construíram o sistema Prodext, baseado em regras de produção. Bouckaert (2002) utilizou um classificador baseado em redes bayesianas, e Borkar et al (2001) criou o sistema DATAMOLD, baseado em Modelos de Marvok Escondidos. Temos ainda o CiteSeer [Bollacker et al 1998], que engloba todo o processo de criação de uma base de referências a partir de documentos on-line. Experimentos com alguns desses sistemas estão publicados na literatura. Contudo, não é possível comparar o desempenho desses sistemas ao nosso porque existe uma grande variação no corpus usado para testes.

6. Conclusões

Neste trabalho, propomos uma abordagem híbrida para EI, baseada na combinação de classificadores de texto e modelos HMM. Na abordagem proposta, um HMM é usado para refinar as respostas iniciais fornecidas pelo classificador para um dado texto de entrada. A abordagem proposta é original dentro da área de EI.

A principal contribuição deste trabalho foi combinar duas técnicas nunca antes combinadas em um mesmo sistema de EI. Os resultados dos experimentos realizados no domínio de referências bibliográficas indicam que o uso do HMM compensou os desempenhos ruins obtidos com classificadores e conjuntos de características menos adequados. Um bom desempenho pode ser obtido até mesmo com o uso de características definidas sem tanto esforço de especialistas.

Embora o protótipo tenha sido implementado para EI a partir de referências bibliográficas, ele pode ser facilmente adaptado para outros domínios de aplicação. Como trabalhos futuros, podemos destacar ainda a definição de outras estruturas de HMM (atualmente todos os estados estão conectados entre si), e o uso de aprendizagem na etapa de geração dos fragmentos.

Referências

Aha, D. & Kibler, D. (1991), Instance-based learning algorithms, *Machine Learning*, Vol. 6, pp. 37-66.

Appelt, D. E. & Israel, D. (1999), Introduction to Information Extraction Technology, *IJCAI-99 Tutorial*, Stockholm, Sweden.

Bollacker, K., Lawrence, S. & Giles, C. L. (1998), CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications, *Proceedings of the 2nd International Conference on Autonomous Agents*, pp. 116-123.

Borkar, V. R., Deshmukh, K., & Sarawagi, S. (2001), Automatic segmentation of text into structured records, *Proceedings of the ACM-SIGMOD Int'l Conference on Management of Data*, pp. 175-186.

Bouckaert, R. R. (2002), Low level information extraction: a bayesian network based approach. In *TextML.*

Califf, M.E. & Mooney, R.J. (1999), Relational learning of pattern-match rules for information extraction, *Proceedings of the 16th National Conf. on AI*, pp. 328-334.

Frank, E. & Witten, I.H. (1998), Generating accurate rule sets without global optimization, *Proc. of the 15th International Conf. on Machine Learning,* pp. 144-151.

Hopcroft, J.E. & Ullman, J.D. (1979), *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley Publishing Co., Reading Massachusetts.

Hsu, C. & Dung, M. (1998), Generating finite-state transducers for semistructured data extraction from the web, *Journal of Information Systems*, Vol. 23(8), pp. 521–538.

John, G., H. & Langley, P. (1995), Estimating continuous distributions in bayesian classifiers, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Mateo, pp. 338-345.

Kosala, R.J., den Bussche. V., Bruynooghe, M. & Blockeel, H. (2002), Information extraction in structured documents using tree automata induction, *Proc of the 6th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pp. 299-310.

Kushmerick, N. & Thomas, B. (2003), Adaptive information extraction: Core technologies for information agents, Lecture Notes in Computer Science, Vol. 2586, Springer, pp. 79-103.

Kushmerick, N., Weld, D.S. & Doorenbos, R. (1997), Wrapper induction for information extraction, *Proc. of the 15th Int'l Joint Conference on AI*, pp. 729–735.

Kushmerick, N., Johnston, E. & McGuinness, S. (2001), Information extraction by text classification, *IJCAI Workshop on Adaptive Text Extraction and Mining,* Seattle, WA.

Nunes, C. & Barros, F. A. (2000), ProdExt: a knowledge-based wrapper for extraction of technical and scientific production in Web pages, *Proc. of the International Joint Conference IBERAMIA-SBIA 2000 - Open Track*, pp. 106-115.

Rabiner, L. R. & Juang, B.H. (1986), An introduction to hidden Markov models, *IEEE ASSP Magazine*, Vol. 3(1), pp. 4-16.

Soderland, S. (1999), Learning information extraction rules for semi-structured and free text, *Machine Learning*, Vol. 34(1-3), pp. 233–272.

Witten, I.H., Frank, E. (1999), *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann.

Wolpert, D.H. (1992), Stacked Generalization, *Neural Networks*, Vol. 5, pp. 241-259.

Yang, Y. & Pedersen, J.O. (1997), A comparative study on feature selection methods in text categorization, *Proc. of the 14th International Conference on Machine Learning, ICML97*, pp. 412-420.