Geração automática de metadados para documentos técnicos científicos[1]

Bruno Viana Rezende
brunore@dcc.ufmg.br
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais

Marcello Peixoto Bax
bax@ufmg.br
www.bax.com.br
Escola de Ciência da Informação
Universidade Federal de Minas Gerais

Resumo
É sabido que a época atual tem como uma de suas características a explosão de informações disponíveis, principalmente nas redes de comunicação. Um domínio em que o volume de informações já é gigantesco, e vem crescendo a cada dia, é a área científica. Uma das formas de se facilitar a recuperação de informações é promover a utilização de metadados descritivos de recursos informacionais disponíveis na Web. Este documento trata da especificação de uma ferramenta para extração de informação e geração automática de metadados descritivos para documentos técnicos científicos disponíveis na Web. Esta ferramenta comporá mais tarde a arquitetura de uma biblioteca digital auto-evolutiva, caracterizada principalmente por uma rede de citações autônoma.

[1] Trabalho apresentado no XII ENDOCOM, XXV Congresso Anual em Ciência da Comunicação, Salvador/BA, 05. setembro.2002.

Introdução

É sabido que a época atual tem como uma de suas características a explosão de informações disponíveis, principalmente nas redes de comunicação. Fala-se da sociedade da informação e do conhecimento, bem como de uma nova economia. Um domínio em que o volume de informações já é gigantesco, e vem crescendo a cada dia, é a área científica [Figura 1].

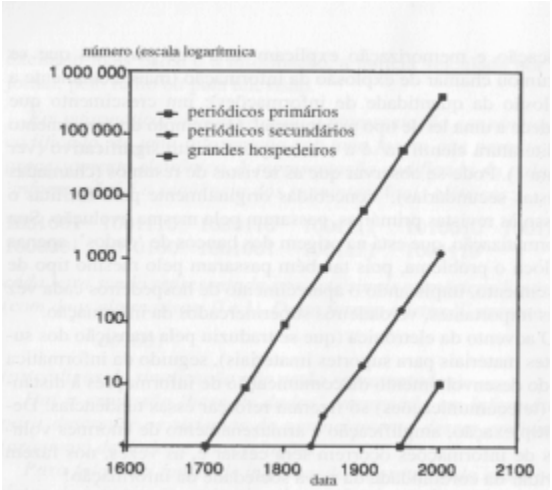

Figura 1. Crescimento das literaturas primária e secundária e dos grandes hospedeiros de bancos de informações

Figura 1 - Fonte: A Ciência da Informação, Yves-François, Le Coadic [4].

Na sociedade da informação, a diminuição do tempo de inovação é vital. O processo tradicional de divulgação científica que engloba a pesquisa, redação dos resultados, avaliação pelos pares, divulgação e assimilação de conhecimentos, apesar de ser confiável é demorado. Bush [3], citado por [14] exemplifica o fato: "As leis da genética de Mendel ficaram perdidas por uma geração porque sua publicação não alcançou os que seriam capazes de entendê-las e de estendê-las".

Um passo em direção à diminuição do tempo de inovação é fazer com que a produção científica chegue às mãos daqueles que dela necessitam. A internet se consolidou como valiosa fonte de referências bibliográficas para pesquisadores e estudantes atenderem suas necessidades de informação. Apesar disso, ainda há muito que ser feito, principalmente no que se refere à organização do acervo disponível. A recuperação precisa de documentos ainda não é regra geral. Entre as causas de tal deficiência, pode-se citar o volume de informações disponíveis e a dispersão dos documentos técnicos e científicos na internet [9].

Segundo o Livro Verde para a Sociedade da Informação no Brasil [15]:

> ***"Muito do desenvolvimento de um país depende da capacidade de organização de suas instituições no tocante aos acervos de informações. O fato de os conteúdos estarem sempre sendo produzidos e armazenados de forma descentralizada e dispersa obriga a um enorme esforço para reunir e incorporá-los como serviços e produtos. Daí a importância de se desenvolverem interfaces que possibilitem ao cidadão uma interação fácil, com meios de acesso facilitados pela descrição dos conteúdos dos documentos eletrônicos em arquiteturas de metadados."***

É sobre esta visão que se baseia o projeto que engloba o presente trabalho, qual seja, registrar de forma sistematizada e automatizada a produção científica brasileira, com o objetivo de atender as necessidades de informação da sociedade [2].

Uma possível solução para o registro sistemático da produção científica brasileira é a criação de uma biblioteca digital autônoma que obtenha na Internet os conteúdos científicos e que disponibilize estes conteúdos de forma gratuita, simples e precisa.

A obtenção e a descrição de conteúdos são algumas das necessidades que devem ser atendidas para o funcionamento satisfatório de uma biblioteca digital. Tais questões podem ser

resolvidas tanto manual quanto automaticamente. Ambas abordagens possuem vantagens e desvantagens. Embora a abordagem manual possua uma baixa taxa de erro quanto à descrição e alta confiabilidade e relevância dos documentos obtidos, provavelmente avaliados por pessoas qualificadas, o custo para sua manutenção é alto, já que é necessário mão de obra intensiva e bastante especializada. Outra desvantagem é a relativa lentidão na incorporação de novos documentos ao seu acervo.

Já a abordagem automática tem como vantagens o seu baixo custo de manutenção e a agilidade na incorporação de novos documentos ao acervo. As principais desvantagens, nesta abordagem, são a taxa de erro na precisão da descrição de documentos, relativamente alta, e a confiabilidade dos documentos obtidos, visto que o julgamento da qualidade de um documento é uma tarefa difícil de ser delegada a máquinas.

A partir destas considerações, verifica-se que a descrição automática de documentos, com baixa taxa de erros, é de grande valia para uma biblioteca digital.

O trabalho aqui apresentado trata da construção de uma ferramenta que realiza a extração automática de metadados de artigos científicos brasileiros.

Este documento se divide da seguinte maneira, na Seção 2 é apresentado o problema, na seção 3 são descritos alguns conceitos utilizados na construção da ferramenta; na Seção 4 será descrita a abordagem dada ao problema; na Seção 5 é apresentado o experimento realizado; na Seção 6 são apresentados trabalhos futuros e na Seção 7 são apresentadas as referências bibliográficas.

Especificação do problema

A pesquisa objetiva o desenvolvimento de uma ferramenta que extraia metadados descritivos a partir de documentos científicos automaticamente. Até o momento considera-se apenas arquivos no formato PDF (*Portable Document Format*). A tradução de arquivos no formato PDF para Plain Text [17], formato de entrada da ferramenta, foi feita utilizando o programa pdftotext, disponível no pacote XPDF [11].

Os metadados extraídos pela ferramenta são: Título, Autores, Resumo, Palavras Chave e Referências bibliográficas.

Os metadados descritivos utilizados serão aqueles que fazem parte do padrão *Dublin Core* [5]. Os metadados serão representados utilizando *RDF (Resource Description Framework)* [12] e expressados em *XML (eXtensible Markup Language)* [7]. A descrição e relação destas tecnologias é feita na seção 3.

Referencial Teórico: Metadados e Modelo de Markov

2 XML – Extended Markup Language

***XML* é uma linguagem de marcação utilizada para a descrição de dados que é derivada da linguagem *SGML* (*Standard Generalized Markup Language*) [16], linguagem utilizada para a descrição de outras linguagens [1]. *XML* foi criada para auxiliar a troca de documentos eletrônicos na *Web*. A descrição de dados em *XML* é restrita ao nível sintático, não há a priori qualquer significado nas marcações de *XML*. Uma aplicação que processe um documento *XML* deve conhecer o significado de cada *tag* do documento, já que não é definido qualquer conjunto de *tags* a priori. Cada aplicação irá definir seu próprio conjunto de *tags* bem como a semântica destes.**

3 RDF – Resource Description Framework

RDF é um "*framework*" para metadados cujo objetivo é promover a interoperabilidade entre aplicações que compartilham informações entendíveis por máquinas/programas na Web [8]. RDF não define qualquer semântica nem faz qualquer pressuposto sobre qualquer domínio de conhecimento, já que deve ser um mecanismo de descrição neutro que sirva para descrever recursos de qualquer campo de conhecimento [12].
RDF é composto por três tipos de objetos: Recursos, Propriedades e Tripla.
Um recurso é o que será descrito por uma expressão RDF. Todo recurso é identificado por um *URI* (*Uniform Resource Identifier*, incluindo aí o *Uniform Resource Locator - URL*).
Uma propriedade é qualquer característica utilizada para descrever um recurso. Em RDF um domínio de conhecimentos é definido via um *RDF Schema* [13]. É em um *RDF Schema*, portanto, que é definida a semântica e as características de uma propriedade. Uma aplicação que crie metadados em RDF e outra que utilize estes metadados devem utilizar o mesmo *Schema* para um funcionamento adequado.
Uma tripla é um recurso uma propriedade e o valor da propriedade para um recurso. Uma tripla possui a seguinte forma <sujeito, predicado, objeto>. O significado de uma tripla pode ser resumido como o recurso (sujeito) possui a propriedade (predicado) com este valor (objeto). Um objeto pode ser tanto um outro recurso quanto um tipo primitivo definido por XML. Por exemplo, a tripla <"http://www.dcc.ufmg.br/~brunore", "criador", "Bruno Rezende"> teria o significado: Bruno Rezende é o criador da página http://www.dcc.ufmg.br/~brunore. É importante notar que um recurso pode ter mais que um valor para uma dada propriedade. Por exemplo, suponha que o indivíduo X e o indivíduo Y tenham construído a página http://pagina.com.br/. A existência das duas triplas <"criador", "indivíduo X"> e <"http://pagina.com.br/", "criador", "indivíduo Y"> em um documento RDF não seria errônea.
Todas triplas representam um grafo direcionado que vai do nodo sujeito para o nodo objeto e o arco tem o valor do predicado. Um recurso é representado por uma elipse enquanto um terminal é representado por um retângulo. As triplas acima seriam representadas por:

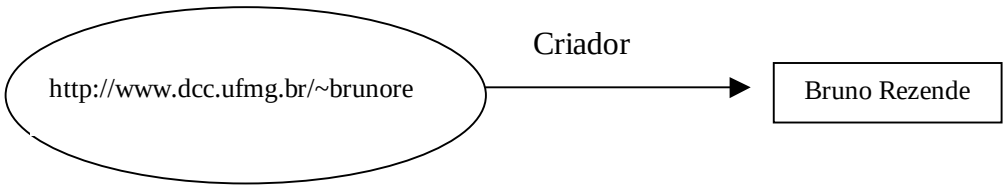

Figura 3

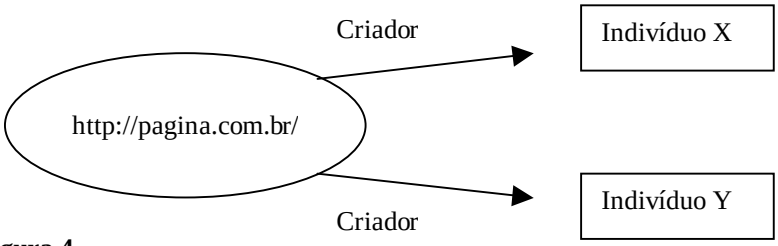

Figura 4

A sintaxe de RDF pode ser expressa em XML da forma especificada em [12].

4 Dublin Core

O *Dublin Core Metadata Elements Set* (DCMES) é uma ontologia criada pela *Dublin Core Metadata Initiative* (DCMI) para a descrição de recursos *Web* [5]. DCMES provê um vocabulário semântico para descrever as propriedades básicas de um recurso, por exemplo, o autor (*Creator*) e o título (*Title*). O objetivo de ter sido criado um padrão para a descrição de recursos é possibilitar o desenvolvimento de agentes inteligentes de software para a recuperação de informações na Web. Existem diversas razões para Dublin Core ser utilizado na descrição de recursos [10]:

Simplicidade**: O conjunto de elementos é simples de ser entendido, podendo ser usado tanto por leigos quanto por especialistas em descrição de recursos.**

Interoperabilidade semântica**: Os elementos podem ser utilizados para descrever recursos de diversas áreas de conhecimento. Tal facilidade permite que sejam realizadas pesquisas sem se importar com especificidade dos diversos campos de conhecimento.**

Consenso internacional**: Dublin Core vem sido utilizado em projetos em cerca de 20 países.**

Facilidade de extensão**: Dublin Core foi desenvolvido tendo em vista facilitar comunidades de diversas áreas do conhecimento estender seu conjunto básico de metadados.**

O conjunto de metadados Dublin Core é formado por quinze elementos: Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage, Rights. A descrição de cada elemento pode ser encontrada em [6].

Todos os elementos relacionados são opcionais ao descrever um recurso, além disso, não há restrição ao número de vezes que cada elemento pode ser utilizado ao descrever um recurso bem como ao tamanho do valor de cada elemento.

5 Hidden Markov Model

Um modelo de Markov é uma estrutura utilizada para modelar situações em que haja ocorrência de padrões. Um processo de Markov é um processo que varia de estado para estado dependendo apenas de seus *n* últimos estados. O processo é chamado de ordem *n* se *n* é o número de estados que afetam a escolha do próximo estado. Um processo de Markov de primeira ordem é um processo em que a escolha de um estado é feita tendo como base apenas o último estado.

A transição de um estado para outro em um processo de Markov é feita probabilisticamente, existindo para tanto uma matriz de transições A. Exemplo, suponha um processo de Markov de ordem 1 com a seguinte matriz de transições A:

	1	2	3
1	0.5	0.3	0.2
2	0.15	0.4	0.45
3	0.2	0.7	0.1

A probabilidade de se chegar ao estado 2 estando no estado 1 é 0.3, ou seja, P(2|1) = 0.3.

Em um processo de Markov é necessário também um vetor de probabilidades inicial (π) que define qual a probabilidade de ocorrência de um estado no tempo 0. Por exemplo, o vetor (0.04, 0.58, 0.38) representa que a probabilidade do estado inicial do sistema ser o estado 1 é 0.04, estado inicial ser o estado 2 é 0.58 e o estado inicial ser o estado 3 é 0.38. Resumindo, um processo de Markov de primeira ordem é definido por:

- Conjunto de estados C;
- Matriz de Transição entre estados A;
- Vetor de probabilidade inicial π;

Por exemplo, considere o problema de previsão do tempo. Vamos modela-lo da seguinte forma:

- **o clima de um dia pode ser determinado considerando apenas o clima do dia anterior, ou seja, é um modelo de Markov de primeira ordem;**
- **o clima de um dia pode ser apenas ensolarado, chuvoso ou nublado, ou seja possui apenas 3 estados.**

O modelo de Markov desse problema seria dado por:
C = (ENSOLARADO, CHUVOSO, NUBLADO)
A =

Ontem \ Hoje	ENSOLARADO	CHUVOSO	NUBLADO
ENSOLARADO	0.5	0.3	0.2
CHUVOSO	0.1	0.7	0.2
NUBLADO	0.3	0.3	0.4

π = (0.4, 0.3, 0.3)

Existem problemas em que um modelo de Markov não é suficiente para a determinação de padrões. Por exemplo, problemas em que os estados do modelo não são diretamente observáveis, os únicos dados a que se tem acesso são as aparições de fenômenos relacionados com um estado oculto.

Vejamos um exemplo:
Suponha um indivíduo que não tenha como determinar como está o tempo hoje diretamente, ou seja, olhando e vendo se está chuvoso, nublado ou ensolarado. Ao invés disso, o indivíduo só pode determinar o tempo verificando se o ambiente está abafado ou não.

Este problema pode ser modelado utilizando uma estrutura chamada Hidden Markov Model (HMM - Modelo Oculto de Markov). Um HMM é um modelo que possui um conjunto oculto de estados que estão, de alguma forma, relacionados com um conjunto de símbolos observáveis. No exemplo citado, o conjunto oculto de estados seria o tempo no dia (Ensolarado, Chuvoso e Nublado) e o conjunto de símbolos observáveis seriam as percepções do indivíduo (Abafado e Não-Abafado).
Um HMM é, portanto, definido por:

- Conjunto de estados ocultos C;
- Matriz de Transição entre estados ocultos A;

- Vetor de probabilidades iniciais de estados ocultos π;
- Conjunto de símbolos observáveis CO;
- Matriz B de co-ocorrência de C e CO.

O exemplo anterior modelado como um HMM seria:
C, π e A iguais;
C0= (ABAFADO, NÃO-ABAFADO)
B=

	ABAFADO	NÃO ABAFADO
ENSOLARADO	0.4	0.6
CHUVOSO	0.4	0.1
NUBLADO	0.2	0.3

Ou seja, neste exemplo, a probabilidade do dia estar ensolarado estando abafado é igual a 0.4, P(ENSOLARADO|ABAFADO) = 0.4.
Existem 3 problemas principais quando se trata de *Hidden Markov Models*:

1. **Dado uma seqüência de observações e um HMM, determinar qual o conjunto de estados ocultos que mais provavelmente gerou as observações;**
2. **Dado uma seqüência de observações e um conjunto de HMMs, determinar qual HMM tem a maior probabilidade de ter gerado a seqüência;**
3. **Dado uma seqüência de observações, gerar o HMM com probabilidade máxima de ter gerado esta seqüência.**

Neste artigo, apenas o primeiro problema é de interesse. Existe uma solução eficiente para o problema, o algoritmo de Viterbi [18].

Abordagem ao problema

6 Utilização de Dublin Core, RDF e XML na ferramenta

Os metadados gerados pela ferramenta fazem parte do conjunto de metadados *Dublin Core*. Abaixo segue a lista dos metadados gerados bem como os dados gerados para cada um deles.

Elemento	Dados Extraídos
Title	Título do artigo.
Creator	Autor do artigo. Para cada autor identificado será gerado um metadado.
Subject	Conjunto de palavras chaves separadas por ‘;’
Description	Resumo do artigo (abstract).
Relation	Citações feitas no artigo.
Source	Documento utilizado para gerar este arquivo.

Tabela 2

A ferramenta se limitará a geração destes metadados por serem os mais relevantes para uma dada consulta e para o entendimento do conteúdo do recurso.

A ontologia *Dublin Core* é definida pelo *RDF Schema* encontrado em *http://purl.org/dc/elements/1.1/*. Os metadados criados serão expressos em RDF utilizando a ontologia especificada pelo *Schema* http://purl.org/dc/elements/1.1/, a sintaxe dos metadados estarão de acordo com a sintaxe de *XML*. Por exemplo, suponha o recurso http://localizacao/recurso.pdf. Este é convertido para texto e é submetido à ferramenta que consegue extrair as informações exibidas na tabela abaixo:

Elemento	Dados Extraídos
Title	Um título qualquer
Creator	Um criador qualquer
Subject	Palavra 1, palavra 2, palavra 3
Description	Uma descrição qualquer
Relation	Citação 1, Citação 2, Citação 3
Source	http://localizacao/recurso.pdf

Representação em forma de grafo:

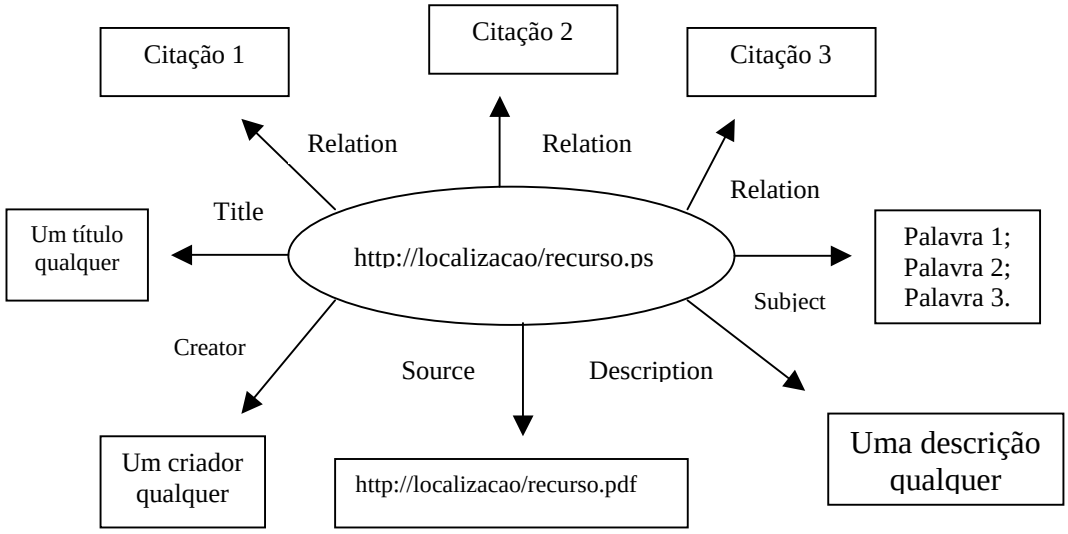

Figura 5: Representação da tabela descritiva de http://localizacao/recurso.pdf como grafo.

O texto abaixo é a descrição em RDF da figura 5, que constituirá a saída da ferramenta:

7 Extração dos metadados

O formato de entrada da ferramenta é um arquivo PDF, mas a extração dos metadados é realizada sobre texto plano (*plain text*). A conversão do arquivo PDF para TXT é feita utilizando o programa pdftotext, disponível no pacote XPDF [11]. Tal transformação dificulta a extração, já que características relevantes da fonte de uma palavra, como por exemplo o tamanho ou estilo, são perdidas. Além disso a forma com que a transformação é realizada gera outros problemas que serão discutidos na seção 5.2.

Tendo realizado a conversão, a extração de metadados é dividida em duas abordagens, uma que utiliza casamento de padrões simples e outra que modela o problema utilizando um

```
<?xml version="1.0"?>
<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:dc="http://purl.org/dc/elements/1.1/">
      <rdf:Description about="http://localizacao/recurso.ps">
            <dc:source>
                  http://localizacao/recurso.pdf
            </dc:source>
            <dc:title>
                  Um título qualquer
            </dc:title>
            <dc:creator>
                  Um criador qualquer
            </dc:creator>
            <dc:subject>
                  Palavra 1; palavra 2; palavra 3
            </dc:subject>
            <dc:description>
                  Uma descrição qualquer
            </dc:description>
            <dc:relation>
                  Recurso 1
            </dc:relation>
            <dc:relation>
                  Recurso 2
            </dc:relation>
            <dc:relation>
                  Recurso 3
            </dc:relation>
      </rdf:Description>
</rdf:RDF>
```

***Hidden Markov Model*. A primeira abordagem é utilizada para a extração das referências bibliográficas, do resumo e das palavras-chave enquanto a segunda é utilizada para a extração do título e dos autores.**

Resumo, Palavras-chave e Referências Bibliográficas

Analisando a base de arquivos convertidos, foi verificada a existência de padrões simples que determinam a ocorrência de alguns dos metadados que se pretende extrair. Abaixo segue uma descrição de cada uma dessas características bem como a utilização das mesmas na extração dos metadados.

Resumo

1. **O resumo de um artigo vem sempre precedido da palavra "Resumo";**
2. **Na maioria das vezes, o resumo é seguido pelas palavras-chave e pela palavra "palavras-chave";**
3. **Na maioria das vezes, o resumo é seguido pelas palavras-chave e pela palavra "palavras-chave";**
4. **O comprimento médio do resumo é de 93 palavras.**

Estas características sugerem um algoritmo extremamente simples para a extração do resumo:

1. Na primeira página, procure pela palavra "resumo";
2. Procure pela primeira ocorrência da palavra "palavras-chave" após a palavra "resumo";
3. Se a palavra "palavras-chave" não for encontrada ou for encontrada à uma distância muito grande da palavra "resumo" (130 palavras, o que sugere que a palavra está em outro contexto, no meio do texto) então considere como resumo as 93 primeiras palavras subseqüentes à palavra "resumo";
4. Senão considere como o resumo todas as palavras entre as palavras "resumo" e "palavras-chave".

Palavras-chave:

1. **As palavras chave de vêm sempre precedidas da palavra "Palavras-chave";**
2. **Em sua maioria, as palavras-chave são separadas por ';' e às vezes por ',',**
3. **o conjunto de palavras-chave tem 6 sub-conjuntos e o conjunto de palavras-chavetermina com '.';**
4. **Cada sub-conjunto de palavras-chave é composto por no máximo 5 palavras (ex.: "redes neuronais; banco de dados; redes de computadores.").**

Estas características sugerem, um algoritmo simples para extração das palavras-chave:

1. Enquanto não forem obtidas palavras consideradas palavras-chave faça:
    2. Na primeira página procure pela ocorrência da palavra "palavras-chave";
    3. Procure pela primeira ocorrência do caractere '.' Após a palavra "palavras-chave";
    4. Se o caractere não existir, obtenha as 30 palavras subseqüentes à palavra palavras-chave e verifique quais podem ser consideradas palavras-chave;
    5. Senão, verifique quais as palavras entre a palavra "palavras-chave" e o caractere '.' podem ser consideradas palavras-chave.

A verificação de que o conjunto constitui as palavras-chave é feita de acordo com o

1. Separe o conjunto de palavras em sub-conjuntos, sendo cada sub-conjunto definido pelas palavras anteriores a uma pontuação ('.', ',' e ';');
2. Para cada sub-conjunto e enquanto a condição de parada não for atingida, faça
    3. Se o tamanho do sub-conjunto for menor ou igual a 5, considere o sub-conjunto como sendo parte do conjunto de palavras-chave;
    4. Senão, descarte os sub-conjuntos não processados e considere a condição de parada atingida.

algoritmo:

Referências bibliográficas:

1. **As referências bibliográficas vêm na maioria das vezes precedidas pelas palavras "referências bibliográficas";**
2. **As referências se encontram e sua maioria nas páginas finais do artigo;**
3. **Uma referência bibliográfica é iniciada pela seqüência fim de linha, número e um ponto (a partir de agora referida como FNP), pela seqüência fim de linha, colchete (FC) ou pela seqüência fim de linha e uma palavra cujo primeiro caractere é maiúsculo (FCM);**
4. **Na maioria das vezes um artigo termina com uma referência bibliográfica, apesar da existência de outros conjuntos de palavras que possuem as mesmas características.**

Estas características sugerem o seguinte algoritmo:

1. Procure pela última ocorrência das palavras "referências bibliográficas";
2. Enquanto não chegar ao fim do arquivo faça
    3. Procure pela seqüência FNP, FC ou FCM;
    4. Procure a próxima seqüência e considere-a como sendo o fim da última referência, caso não encontre, considere o fim do arquivo como sendo o fim da referência;
    5. Extraia o conjunto de palavras e verifique se é uma referência.

A verificação de que um conjunto de palavras constitui uma referência é feita apenas pelo tamanho do conjunto, conjuntos menores que 30 são considerados referência. Esta verificação é falha e devem ser desenvolvidos outros algoritmos para melhorar a qualidade da extração.

Autores e Título

A extração do título e dos autores de um artigo é mais complicada que a dos outros metadados, já que não foi verificada a existência de padrões seguros que pudessem ser utilizados. Sendo assim, modelou-se o problema como um *Hidden Markov Model* da seguinte forma:

- O conjunto de estados ocultos é formado pelas classes que aparecem na primeira página de um artigo, são elas: Título, Autores, Notas sobre autores, Resumo, Palavras-chave e Resto, onde Resto é tudo aquilo que não pertence às outras classes;

- Os símbolos observáveis são as palavras que ocorrem na primeira página.

Para obter qual foi a mais provável sequência de estados ocultos (classes) que gerou os símbolos observáveis (palavras) foi utilizado o algoritmo de Viterbi [18].
Tendo o conjunto de estados ocultos e os símbolos observáveis, foram criadas as matrizes A e B e o vetor π. Abaixo segue uma breve descrição do significado de cada matriz e do vetor:
Matriz A**: Matriz com a probabilidade de se ir para uma classe estando em outra. As linhas representam o estado atual e as colunas o estado seguinte. Por exemplo, a probabilidade da classe Resumo vir depois da classe Autores é definida pelo termo A[AUTORES][RESUMO].**
Matriz B**: Matriz com a probabilidade de uma palavra aparecer em uma classe. As linhas representam o estado e as colunas as palavras. Por exemplo, a probabilidade da palavra José aparecer na classe Autores é definida pelo termo B[AUTORES][JOSÉ].**
Vetor π**: Vetor com as probabilidades de uma classe iniciar um artigo. Por exemplo, a probabilidade de um artigo ser iniciado com o título é definida por π[TÍTULO].**
A determinação dos valores das matrizes e do vetor foi feita da seguinte maneira:

- **Foi realizada a conversão de um conjunto de 50 artigos no formato PDF para TXT;**
- **Manualmente foram identificadas as classes que a serem extraídas;**
- **Utilizando os arquivos marcados com as informações desejadas, foram calculados os valores das matrizes utilizando os algoritmos descritos abaixo.**

Matriz A:

1. Para cada classe (linha)
2. Armazene em uma matriz auxiliar as transições para outras classes e para ela mesma, por exemplo: se em um artigo após a classe título veio a classe autores então aux[título][autores]= aux[título][autores]+1;
3. Tendo sido processados todos artigos, determinar o número total de transições da classe, ou seja, NumTransições[título] = aux[título][título]+aux[título][autores]+... + aux[título][título];
4. Armazenar em A a "probabilidade" de transição de uma classe para outra, ou seja,

A[título][título] = aux[título][título]/NumTransições[título]
A[título][autores] = aux[título][autores]/NumTransições[título]
...
A[título][resto] = aux[título][resto]/NumTransições[título]
...
A[resto][título] = aux[resto][título]/NumTransições[resto]

Vetor π:

1. Para cada classe
2. Armazene em um vetor auxiliar o número de vezes em que foi a primeira classe a aparecer em um artigo, por exemplo, se a classe Título foi a primeira a aparecer em um artigo então aux[título]=aux[título]+1;
3. Armazene a "probabilidade" da classe ser a primeira a aparecer em um artigo, ou seja,

π[título] = aux[título]/numArtigos

π[autores]=aux[autores]/numArtigos

...

π[resto]=aux[resto]/numArtigos

A[título][José] = aux[título][José]/NumAparições[José]

A[autores][José] = aux[autores][José]/NumAparições[José]

...

A[resto][José] = aux[resto][José]/NumAparições[José]

...

A[resto][yyz] = aux[resto][yyz]/NumAparições[yyz]

Matriz B:

Experimento

8 Testes

O teste da ferramenta consiste em submeter um conjunto de artigos para a extração e verificar quantas classes foram extraídas corretamente. O conjunto é formado por 70 artigos não utilizados na construção do HMM.

9 Resultados

	Precisão	Revocação
Título	41,2%	8,4%
Autores	99%	61,7%
Resumo	100%	100%
Palavras-chave	98,6%	93,6%
Referências	84,7%	89,9%

Naqueles metadados recuperados utilizando casamento de padrões simples (Resumo, Palavras-chave, Referências), o resultado pode ser considerado muito bom. No caso dos metadados extraídos utilizando o Hidden Markov Model (Título e Autores), o resultado não correspondeu ao esperado. Apesar da alta precisão das recuperações de autores a revocação é baixa, o esperado era uma revocação de 90% ou mais. No caso do Título tanto a precisão quanto a revocação foram bem abaixo do esperado, o que demonstra ser necessário uma forma de contornar o problema.

A causa do baixo resultado no caso do Título pode ser atribuída à conversão realizada do arquivo PDF para Plain Text. O programa *pdftotext*, utilizado para a conversão, não mantém um padrão na conversão dos arquivos. A posição do título

no arquivo TXT gerado é aleatória, pode ser tanto no início da primeira página quanto no fim ou entre uma coluna e outra.

No caso dos autores, a revocação é baixa devido à alta ocorrência de palavras desconhecidas tanto em seu corpo quanto no corpo de outras classes. Tal fato, gera uma "indecisão" do algoritmo de Viterbi para identificar a qual classe pertence uma determinada palavra desconhecida.

Trabalhos futuros

No trabalho apresentado restaram questões que devem ser resolvidas. Possíveis abordagens para soluciona-las incluem:

Aprimorar conversão de arquivos PDF para Plain Text**: Um dos problemas encontrados no reconhecimento do título de artigos foi o posicionamento aleatório de suas palavras no arquivo TXT gerado, uma das formas de melhorar este problema é realizar uma conversão em que o título venha nas primeiras linhas do arquivo convertido;**

Acrescentar informações de Currículos Lattes [19]**: Currículos Lattes são currículos de pesquisadores brasileiros disponíveis na Internet. Existem informações contidas nestes currículos úteis para o problema em questão, como por exemplo nomes de autores e títulos de artigos;**

Utilizar dados das referências bibliográficas na construção do Hidden Markov Model**: As referências bibliográficas possuem, assim como os Currículos Lattes, informações úteis para o problema, como por exemplo, títulos de artigos e nomes de autores.**

Referências Bibliográficas

[1] BAX, M. P. - Introdução às linguagens de marcas – Ciência da Informação, Brasília, v. 30, n. 1, p. 32-38, jan./abr. 2001

[2] BAX, M. P.; KURAMOTO, H. Pesquisa e Desenvolvimento de Ferramentas para a Captura, Tratamento e Recuperação de Literatura Científica na Web

[3] BUSH, Vannevar. As we may think. The Atlantic Monthly, Jul.1945. Disponível em: http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm

[4] COADIC, Y. L.. A Ciência da Informação, Cap. 1, p. 8, ISBN 85-85637-08-0

[5] Dublin Core Metadata Initiative (DCMI) - http://dublincore.org/

[6] Dublin Core Data Set elements - http://dublincore.org/documents/dces/

[7] Extensible Markup Language (XML) - http://www.w3.org/XML/

[8] Frequently Asked Questions about RDF - http://www.w3.org/RDF/FAQ

[9] GILES, C. L. & BOLLACKER, K. D. & LAWRENCE, S. – Indexing and Retrieval of Scientific Literature - Eighth International Conference on Information and Knowledge Management, CIKM 99, Kansas City, Missouri, November 2–6, pp. 139–146, 1999.

[10] HILLMANN, D. Using Dublin Core – http://dublincore.org/documents/2001/04/12/usageguide/

[11] NOONBURG, D. B. XPDF, disponível na web: http://www.foolabs.com/xpdf/

[12] Resource Description Framework (RDF) Model and Syntax Specification - http://www.w3.org/TR/REC-rdf-syntax

[13] Resource Description Framework(RDF) Schema Specification - http://www.w3.org/TR/1998/WD-rdf-schema/

[14] SENA, N. K. Open archives: caminho alternativo para a comunicação científica. Ciência da Informação, Brasília, v. 29, n. 3, p. 71-78, set./dez. 2000

[15] Sociedade da Informação no Brasil – Livro verde, Cap. 5, p. 60

[16] Standard Generalized Markup Language - ISO 8879

[17] The Text/Plain Format Parameter - http://www.rfc-editor.org/rfc/rfc2646.txt

[18] VITERBI, A. J. 1967. Error bounds for convolutional codes and an asymtotically optimum decoding algorithm. IEEE Transactions on Information Theory IT-13:260--267.

[19] Plataforma Lattes, disponível em: http://www.cnpq.br/plataformalattes/curriculolattes/