José Alberto Grossi Júnior

Análise Comparativa de Ferramentas de Extração de Metadados em Artigos Científicos

Belo Horizonte/MG, Brasil

Mai 2015, v-0.6.1

José Alberto Grossi Júnior

Análise Comparativa de Ferramentas de Extração de Metadados em Artigos Científicos

Dissertação de Mestrado apresentada à coordenação do PPGCI/UFMG com o objetivo de obtenção de título de Mestre em Ciência da Informação.

Universidade Federal de Minas Gerais – UFMG

Escola de Ciência da Informação

Programa de Pós-Graduação em Ciência da Informação

Orientador: Marcello Peixoto Bax

Coorientador: Renato Rocha Souza

Belo Horizonte/MG, Brasil

Mai 2015, v-0.6.1

José Alberto Grossi Júnior
Análise Comparativa de Ferramentas de Extração de Metadados em Artigos Científicos/ José Alberto Grossi Júnior. – Belo Horizonte/MG, Brasil, Mai 2015, v-0.6.1-
71 p. : il. (algumas color.) ; 30 cm.

Orientador: Marcello Peixoto Bax
Coorientador: Renato Rocha Souza

Dissertação (Mestrado) – Universidade Federal de Minas Gerais – UFMG
Escola de Ciência da Informação
Programa de Pós-Graduação em Ciência da Informação, Mai 2015, v-0.6.1.

1. Extração de metadados 2. Metadados. I. Artigos científicos. II. Extração de informação. III. Ciência da Informação. IV. Análise Comparativa de Ferramentas de Extração de Metadados em Artigos Científicos

CDU 02:141:005.7

José Alberto Grossi Júnior

Análise Comparativa de Ferramentas de Extração de Metadados em Artigos Científicos

Dissertação de Mestrado apresentada à coordenação do PPGCI/UFMG com o objetivo de obtenção de título de Mestre em Ciência da Informação.

Trabalho aprovado. Belo Horizonte/MG, Brasil, 26 de maio de 2015:

Marcello Peixoto Bax
Orientador

Renato Rocha Souza
Coorientador

Beatriz Valadares Cendón
Professora Convidada

Renata M. Abrantes Baracho Porto
Professora Convidada

Gercina Ângela B. de Oliveira Lima
Professora Suplente

Belo Horizonte/MG, Brasil
Mai 2015, v-0.6.1

Resumo

São inúmeras as ferramentas para extração de metadados em artigos científicos, tendo cada uma sua particularidade, tecnologia e técnicas utilizadas. Porém, com a crescente produção científica e a grande variedade de editoras, eventos e congressos, um grande número de artigos permanece sem uma extração de metadados eficaz, o que dificulta a disseminação de conhecimento e principalmente a pesquisa eletrônica destes documentos.

Este trabalho realiza um teste com algumas ferramentas pré-selecionadas com um conjunto pré-determinado de artigos, que abrange diversas áreas do conhecimento, diversos eventos e formatos visuais diferentes. Estes testes são realizados em ambientes pré-configurados de acordo com a necessidade tecnológica de cada ferramenta, permitindo que todos os artigos tenham seus metadados extraídos por cada uma delas e seus resultados comparados individualmente.

Desta forma, com base nos resultados apresentados, pode-se identificar o comportamento de cada uma das ferramentas perante à extração de metadados, suas falhas, onde são necessários ajustes e onde se obtém um maior sucesso na extração. Além disso, é apresentado também o Índice de Confiabilidade, onde cada ferramenta recebe uma nota com base nos resultados obtidos na extração de metadados pela seleção de artigos realizada.

Palavras-chaves: artigos científicos, extração de metadados, extração de dados em artigos.

Abstract

Currently we can find numerous tools to extract metadata from scientific papers, each one with your particularity, technology and techniques. However, with the crescent scientific production and the numerous publishers, events and conferences, a large part of papers still remain without an effective metadata extraction, hindering the knowledge dissemination e mainly the electronic search for these documents.

The present work makes tests with pre selected tools with a set of scientific papers, covering different areas of knowledge, different events and layouts. These tests were made inside custom environments according the technologies each tool needs, allowing all papers to be tested and their metadata extracted, comparing results one by one.

Thereby, according the presented results, we can identify the behavior of each tool related to the metadata extraction, where it failed, where adjusts are needed and where it has success on the extraction. Moreover, we also present the Reliability Index, a grade received by each tool based on the metadata extraction results using the selected scientific papers.

Palavras-chaves: scientific papers, metadata extraction, data extraction in scientific papers.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

PDF	Portable Document Format
IEEE	Institute of Electrical and Electronics Engineers
RSL	Revisão Sistemática de Literatura
ACM	
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
XML	eXtensible Markup Language
SVM	Support Vector Machines
HMM	Hidden Markov Models
CRF	Conditional Random Fields
URL	Uniform Resource Locators
HTML	HyperText Markup Language
DCMI	Dublin Core Metadata Initiative
SVM	Support Vector Machines
DOI	Digital Object Identifier
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
UFMG	Universidade Federal de Minas Gerais

Sumário

1 Introdução

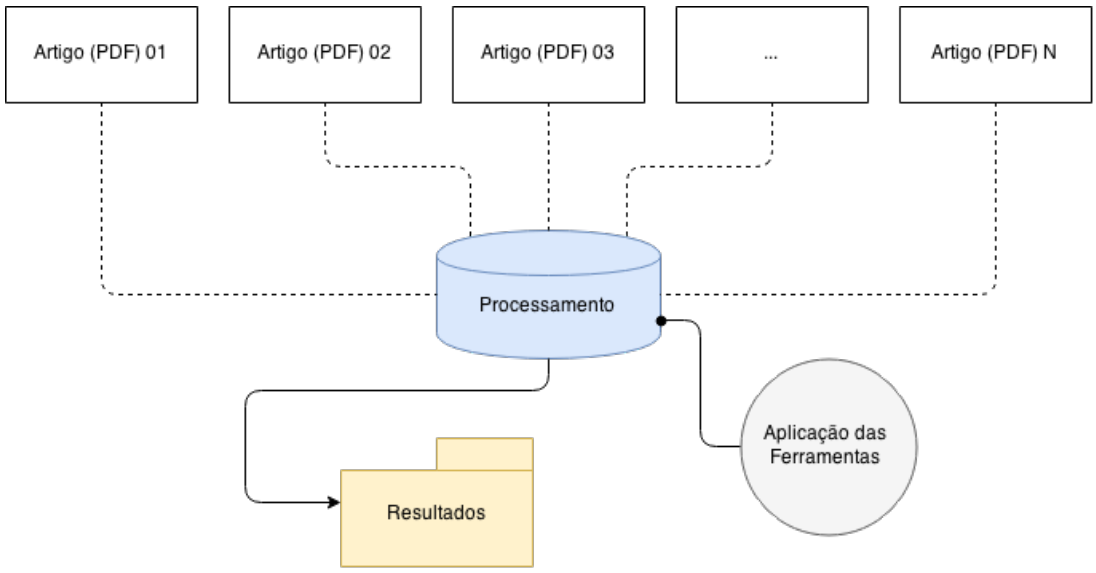
Em virtude da grande produção científica existente nos dias atuais é cada vez mais necessária a utilização de ferramentas automatizadas precisas na extração de informações, o que contribui com uma melhor organização dos artigos científicos e facilita a apresentação de resultados de busca mais rápidos e eficientes.

A pesquisa aqui realizada situa-se no campo da extração de metadados principalmente via *machine learning*. Este trabalho aborda as principais ferramentas encontradas atualmente. Diversas ferramentas e técnicas para extração de metadados em artigos podem ser encontradas na Internet. Algumas são propriedade de universidades ou instituições privadas, o que dificulta a análise. Algumas ferramentas não permitem que testes automatizados sejam feitos, visto que não permitem acesso ao código fonte ou não podem ser utilizadas via linha de comando, dificultando a análise dos resultados.

De modo geral, as ferramentas de extração são focadas em *layouts* (disposição visual) pré-definidos, geralmente seguindo modelos de revistas e encontros científicos, que possuem um padrão visual já estabelecido, como é o caso do IEEE (*Institute of Electrical and Electronics Engineers*), por exemplo, que serve de referência para diversos outros eventos da área da Ciência da Computação, tomando seu *layout*, a princípio, como base.

Porém, existem diversos outros eventos que possuem *layouts* de artigos fora de padrão que necessitam de adaptações por parte das ferramentas para serem analisados e catalogados de maneira automática.

Figura 1 – Processo de Extração de Metadados



Fonte: O próprio autor

Algumas ferramentas são aparentemente muito eficazes para um certo grupo de artigos, já seguindo um padrão visual pré-determinado. Porém, para alguns *layouts* pouco comuns, de áreas do conhecimento diversas, espera-se que estas ferramentas não sejam tão eficazes, variando de acordo com a tecnologia utilizada e, principalmente, de acordo com o princípio teórico utilizado pelos seus autores, que será mais aprofundado nos próximos capítulos.

Como definido por (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012), *machine learning* permite uma forma de aprendizado com base em experiências passadas, através da utilização de dados coletados, que são analisados posteriormente seguindo padrões definidos.

Este tema é muito amplo e sua aplicabilidade é diversificada, permitindo a utilização de suas técnicas e teorias em diversas atividades, como classificação de textos e documentos, processamento de linguagem natural, reconhecimento de fala, detecção de fraudes, diagnósticos médicos e sistemas de recomendações, além de mecanismos de buscas e extração de informação, ponto principal de discussão deste trabalho.

O objetivo da pesquisa é comparar ferramentas de extração de metadados em artigos científicos, identificando também suas melhores aplicações, com base em um conjunto de documentos pré-selecionados para testes, dos mais diversos padrões e de diversas áreas do conhecimento.

Com efeito, esta identificação permite que resultados sejam comparados e confrontados, para decidir qual ferramenta é melhor utilizada para cada padrão visual, abrangendo um conjunto cada vez maior de dados e tendo resultados cada vez mais precisos.

Com base na diferenciação dos *layouts* de artigos científicos, este trabalho visa identificar também pontos em que as ferramentas de extração de metadados necessitam de adaptações por parte de seus usuários e desenvolvedores, garantindo assim, uma cobertura mais abrangente dos artigos científicos. Além disso, com base nos resultados coletados, pode-se identificar qual ferramenta é melhor aplicada para cada tipo distinto de metadado.

Acredita-se que os padrões de extração existentes hoje são, de maneira geral, insuficientes para suprir todos os *layouts* de artigos existentes, limitando a apenas uma pequena parcela destes, dentro de um padrão visual específico.

As formas de extração de metadados em artigos científicos são geralmente baseadas em *layouts*, ou seja, em pequenos pedaços de documentos onde certos dados devem estar presentes. Porém, em virtude da diversidade de materiais produzidos, dos mais diversos padrões visuais e áreas do conhecimento, este *layout* padrão não se mostra eficiente na abrangência total das necessidades da comunidade científica como um todo.

Sobre este aspecto, espera-se que certos artigos científicos não tenham seus metadados extraídos de maneira precisa por todas as ferramentas analisadas, uma vez que

adaptações no código seriam necessárias a fim de permitir que outros padrões visuais de artigos fossem também reconhecidos, aplicando então um dos fundamentos do *machine learning*, o aprendizado por repetição.

Por fim, espera-se que, com esta análise aprofundada das ferramentas selecionadas, as mais eficazes na extração de metadados sejam identificadas e comparadas, elevando então o ganho científico neste tema importante para o desenvolvimento tecnológico.

O documento é estruturado iniciando com uma breve introdução sobre o tema.

O segundo capítulo tem como base o referencial teórico feito através de um mapeamento de bases de dados e eventos científicos relevantes para a área. Neste capítulo são apresentados alguns conceitos básicos, além das técnicas mais utilizadas e as ferramentas mais comuns encontradas atualmente.

O terceiro capítulo apresenta a metodologia usada no trabalho, citando as ferramentas que serão testadas e principalmente como serão feitos os testes.

Posteriormente, no capítulo quarto faz-se a análise e apresentação dos resultados, explicando como os testes foram realizados, os ambientes de teste criados e os resultados coletados.

No quinto capítulo temos a discussão/conclusão, trabalhos futuros e considerações finais sobre o trabalho apresentado.

2 Referencial Teórico

Como o objetivo do trabalho tem foco no resultado prático, geralmente as bases relacionadas às áreas mais técnicas, como Ciência da Computação, devem ser bem focadas, como IEEE e ACM *(Association for Computing Machinery)*, por exemplo. Outras bases multidisciplinares também foram pesquisadas, através do site da CAPES, porém com caráter de apoio na fundamentação teórica.

A pesquisa deveria ser focada em técnicas e ferramentas para extração de metadados em artigos científicos, que permitissem testes com dados reais, fazendo assim com que os resultados obtidos pudessem ser comparados e confrontados para verificar então a eficácia deste processo de extração.

Como a pesquisa necessita de um resultado prático e eficaz, os critérios que serão adotados para demonstrar a relevância de uma ferramenta serão seus resultados, sua participação de mercado e sua maturidade. Cada ferramenta deverá ser testada juntamente com um grupo de artigos previamente selecionado. Estes artigos já possuirão seus metadados extraídos manualmente, permitindo então comparar os resultados obtidos com os resultados de cada uma das ferramentas. Portanto, os critérios utilizados serão de natureza explicitamente prática, com foco em resultados concretos, numericamente representados.

Visando abranger grande parte da literatura sobre o assunto foram mapeados alguns eventos para a pesquisa, a fim de encontrar trabalhos relevantes para a área. Sendo assim, foram mapeados os seguintes eventos:

- KDIR (International Conference of Knowledge Discovery and Information Retrieval);
- ICDAR (International Conference on Document Analysis and Recognition);
- PDCAT (International Conference on Parallel and Distributed Computing, Applications and Technologies);
- IAPR (International Workshop on Document Analysis Systems);
- ACM Conference on Digital Libraries.

Estes eventos foram analisados observando todas as citações e referências utilizadas em cada artigo publicado, objetivando encontrar também detalhes importantes para a pesquisa, sem perder o foco central do trabalho.

2.1 Metadados

2.1.1 Conceito de Metadado

Com base nas ideias apresentadas por (CATHRO, 1997), podemos definir inicialmente um metadado da seguinte maneira:

> *"[...] an element of metadata describes an information resource, or helps provide access to an information resource."*

Sobre este aspecto conclui-se que todo dado que agregue nova informação a um recurso pode ser considerado um metadado. Desta maneira, quanto mais metadados um recurso tiver, mais detalhado ele é, ou seja, mais dados sobre ele teremos. Com efeito, podemos simplificar ainda mais a definição de metadado com sendo "um conjunto de dados sobre um determinado recurso".

Para exemplificar podemos citar, por exemplo, a utilização de pequenos pedaços de dados sobre um conjunto de livros, dentro de um ambiente de biblioteca, por exemplo, o que é considerado uma coleção de elementos de metadados, conforme pode ser confirmado em (CATHRO, 1997). Além disso, o mesmo autor cita como exemplos de metadados os dados coletados por mecanismos de busca no momento em que as páginas da Internet são indexadas e então armazenadas.

Ante o exposto, podemos criar e identificar diversos conceitos do que podemos chamar de metadado, porém, em virtude de sua enorme aplicação e significância, alguns detalhes dentro do escopo deste trabalho devem ser direcionados, visando identificar e explicar melhor os conceitos que serão aplicados nos testes das técnicas e ferramentas que aqui serão apresentadas.

2.1.2 Padrões de Metadados

Diante da infinidade de dados que podem estar atrelados a um determinado recurso, temos uma amplitude muito grande de características que podem ser definidas como sendo metadado. Por isso, foram definidos 15 (quinze) elementos de metadados para descreverem um recurso informacional, independente da disciplina em questão, estabelecendo então um padrão adotado em todo o mundo, o padrão "Dublin Core" (WEIBEL, 1997).

Este padrão se originou após uma série de encontros feitos desde 1995, unindo bibliotecários e pesquisadores digitais e de conteúdo, visando identificar padrões para se representar um recurso eletrônico, seja ele uma página na Internet, um texto ou até mesmo um documento em um determinado formato. O nome "Dublin" foi dado em virtude da primeira reunião do grupo, que foi realizada na cidade de Dublin, Ohio. Já o nome "core"

se deu em virtude dos elementos serem amplos e genéricos, sendo então utilizados para descrever uma grande variedade de recursos.

Os quinze elementos que fazem parte do padrão Dublin Core compartilham de um vasto conjunto de vocabulários de metadados, juntamente com diversas especificações técnicas, que são mantidas pela *Dublin Core Metadata Initiative* (DCMI), a agência responsável pela definição destes elementos.

Este padrão é utilizado nos dias atuais para se representar um recurso na Internet, por exemplo, podendo ser qualquer detalhe que possa ser identificado (KUNZE; BAKER, 2007). Com base em suas características, mecanismos de buscas podem indexar um recurso de maneira mais rápida e precisa, pois este é acompanhado de pequenas sinalizações sobre seu conteúdo, que são apresentados pelos metadados.

Para cada elemento descrito pelo padrão temos informações como:

- *label*, que é o texto para leitura e entendimento humano;
- *name*, que é usado para o processamento de máquina, ou seja, um identificador único que a máquina utiliza para reconhecimento.

Os elementos que fazem parte da versão 1.1 do padrão Dublin Core (KUNZE; BAKER, 2007) podem ser vistos no Anexo A deste trabalho.

Abaixo podemos ver um exemplo de código para utilização de metadados Dublin Core em uma página da Internet, onde iremos referenciar os elementos *creator*, *title* e *language*.

```
<meta name="DC.Creator" content="Jose␣Albeto␣Grossi␣Junior" >
<meta name="DC.Title" content="Análise Comparativa de
   Ferramentas de Extração de Metadados em Artigos
   Científicos" >
<meta name="DC.Language" content="pt_BR" >
```

Desta forma, a indexação do autor (*DC.Creator*) da página, bem como seu título (*DC.Title*) e idioma (*DC.Language*), são feitos de maneira bem simplificada e direta. Para páginas da Internet, com as informações todas em formato HTML, a utilização destes metadados perderia o sentido, visto que essas informações poderiam ser facilmente encontradas de outras formas, como a análise do próprio código. Porém, para documentos binários - em formato PDF ou Word, por exemplo - a utilização destes metadados é de suma importância, visto que permite que essas informações básicas sejam capturadas sem a necessidade de análise do conteúdo destes arquivos, o que traria complexidade ao processo de indexação.

2.1.3 Técnicas de Extração de Metadados

Algumas técnicas e algoritmos de extração de metadados são utilizadas em diversos projetos, de maneira a serem citadas em momentos onde exige-se uma precisão maior.

Estas técnicas se baseiam basicamente na classificação de dados com base nas suas representações escritas, tanto baseadas em padrões preestabelecidos ou até mesmo com base em um dicionário de palavras capaz de reconhecer ocorrências em diversas partes de um documento, o que agrega assertividade ao processo de extração.

Técnicas de *machine learning*, extração e classificação de dados utilizam, em sua grande maioria, dados de entrada previamente selecionados em forma semi-estruturada. Estes dados são fornecidos por diversas entidades/órgãos de certa forma estruturados para serem analisados, com alguma bagagem semântica já presente; são os chamados *datasets*. Algumas bases de dados já consolidadas no mercado fornecem estes conjunto de dados reais, com base nos *papers* que estão catalogados internamente, justamente para serem objetos de pesquisa posterior, a fim de facilitar o manuseio e seleção dos documentos de entrada para análise e desenvolvimento de novos projetos.

Com a utilização destes *datasets* o processo de teste fica muito mais fácil. Existem algumas técnicas que possuem melhor desempenhos ao se utilizar dos chamados *training sets*, que são por definição estes *datasets* sendo utilizados para treinar um determinado modelo, promovendo um padrão de extração com base em dados previamente informados.

2.1.3.1 Regular Expressions (RegEx)

Regular Expressions ou Expressões Regulares é uma técnica muito utilizada na computação para reconhecimento de padrões dentro de um conjunto de caracteres, encontrando combinações seguindo uma sequência definida.

Sua origem data-se de 1956 (KLEENE, 1956), sendo fundamentada pelo matemático Stephen Kleene, que deu origem à Teoria da Computação. Somente em 1968 que as expressões regulares ficaram conhecidas, através de Ken Thompson (THOMPSON, 1968), que incluiu a pesquisa de Kleen como funcionalidade dentro de um editor de textos, permitindo que padrões fossem encontrados dentro de arquivos.

Para a utilização de Expressões Regulares é necessário o fornecimento de um padrão, que será a base de busca em todo o conjunto de caracteres existente. Este padrão é representado por um conjunto de símbolos, que determina a forma desejada de reconhecimento. Assim, além de utilizar-se de caracteres específicos pode-se informar números, letras, dentre outros tipos de representação, correspondendo ao que se deseja reconhecer dentro do texto.

Existem variações de formas de representação de uma Expressão Regular, porém a maioria das linguagens de programação atualmente seguem o padrão POSIX - sob

responsabilidade do IEEE e do Open Group <http://opengroup.org/> -, que determina algumas regras para utilização desta técnica (GROUP, 2013).

Como exemplo, visando identificar somente os números existentes dentro da frase "Foram encontrados 4 passageiros dentro do veículo parado na BR262.", devemos escrever uma expressão regular para que estes números sejam identificados. Desta forma podemos representar o que desejamos buscar da seguinte forma: `/[0-9]+/`, onde identificamos:

- o uso do caractere `/`, que determina o começo e fim da expressão regular;
- a representação de algarismos utilizando `[0-9]`, que significa qualquer dígitos de 0 a 9, permitindo que os números sejam identificados dentro do texto;
- a existência de números com mais de um dígito, como é o caso de `262`, necessitando complementar o padrão para permitir um ou mais algarismos, o que é representado pelo caractere de repetição +, que significa exatamente "uma ou mais ocorrência".

Em diversas ocasiões é necessária a utilização de repetições, informando que aquele padrão pode ocorrer diversas vezes. Assim, temos as seguintes formas de representação para repetições:

- Como pôde ser observado no exemplo acima, utilizamos o caractere + para representar "uma ou mais ocorrências". No exemplo `[0-9]+` representamos qualquer número que possua um ou mais dígitos de 0 a 9.
- Já o caractere `?` (interrogação) pode ser utilizado para representar "nenhuma ou apenas uma ocorrência", ou seja, aquele padrão pode ou não existir, é opcional. Na expressão regular `[0-9]?` estamos informando que o dígito é opcional, ou seja, ele pode ou não estar presente no reconhecimento.
- Podemos utilizar o caractere `*` (asterisco) para representar "nenhum ou mais", ou seja, podemos ter nenhuma ou várias ocorrências daquele conjunto, indo do zero ao infinito.

Além das repetições, em alguns momentos necessitamos identificar um número exato de caracteres. Para identificar um ano de 4 (quatro) dígitos, necessitamos informar que o padrão deve ser identificado para apenas 4 dígitos, nem mais, nem menos. Desta forma temos as seguintes formas de representação:

- Para número de ocorrências exatos utilizamos da expressão regular `[0-9]{4}`, que exige que para identificação do padrão o número tenha exatamente 4 dígitos de 0 a 9.

Tabela 1 – Formas de representação de repetições em Expressões Regulares.

Forma de Representação	**Significado**
`?`	Nenhuma ou apenas uma ocorrência
`*`	Nenhuma ou várias ocorrências
`+`	Uma ou mais ocorrências
`{4}`	Exatamente 4 ocorrências
`{4,}`	No mínimo 4 ocorrências
`{,8}`	No máximo 8 ocorrências
`{4,8}`	No mínimo 4 e no máximo 8 ocorrências

- Podemos estipular quantidade mínima de repetições, como por exemplo `[0-9]{3,}`, que significa "qualquer número que possua no mínimo 3 dígitos", ou seja, os números 123, 481145, 9182 seriam identificados, mas 14 não, visto que possui apenas 2 dígitos.
- Podemos também estipular apenas a quantidade máxima desejada, como `[0-9]{,8}`, ou seja, somente os números que possuem até no máximo 8 dígitos. Neste caso, um número com 9 ou mais dígitos não entraria no reconhecimento de padrão.
- Por fim, tomando como base os dois últimos exemplos, podemos informar os valores mínimo e máximo ao mesmo tempo, como `[0-9]{4,8}`, ou seja, números que possuem no mínimo 4 dígitos e no máximo 8 dígitos.

Desta forma podemos consolidar as formas de repetição em Expressões Regulares com base na Tabela 1.

Além dos dígitos podemos representar também caracteres puros, utilizando-se também do operador `|` (chamado *pipe* em inglês), que representa alternância, ou seja, quando temos mais de uma opção. Tomemos a frase "O amor da vida de Ana Maria se chama Paulo". Para identificar o reconhecimento dos nomes próprios nesta frase podemos utilizar o padrão `/Ana Maria|Paulo/`, que quer dizer "Ana Maria ou Paulo".

Utilizando-se do caractere de alternância `|` ainda podemos utilizá-lo em conjunto com outros caracteres. Na expressão regular `/abaca(te|xi)/`, podemos identificar as palavras "abacate" ou "abacaxi", preservando os caracteres `abaca` e alternando entre as opções `te` ou `xi`. Neste caso utilizamos parênteses para representar grupos de caracteres. Caso utilizássemos o padrão sem os parênteses - `abacate|xi` - iríamos ser capaz de identificar apenas as palavras "abacate" ou a palavra "xi".

A aplicação de expressão regular é bastante variada, existindo diversas formas de representação de qualquer padrão necessário. Além dos detalhes já explicados acima, existem certos caracteres chamados "metacaracteres", que possuem significados definidos em uma expressão regular, assim como `+` e `?`, mas objetivando outras formas de representação.

- O ponto (`.`) possui um significado muito importante, sendo considerado "qualquer coisa". Assim, a expressão regular `/abaca../` permite que `abacaxi` e `abacate` também sejam identificados. Ela representa "qualquer coisa que comece com `abaca` e possui mais 2 caracteres quaisquer depois". Ou seja, além de identificar `abacate` e `abacaxi` ela permite identificar `abaca17` ou até mesmo `abaca s`, visto que espaço também é um caracteres e faz parte do reconhecimento de `..`

- Os colchetes - já visto anteriormente - representam um conjunto de caracteres únicos dentro de várias possibilidades. Isso quer dizer que em `/[abc]/` conseguimos identificar qualquer um dos caracteres `a`, `b` ou `c`. Assim, dentro da palavra `casa` conseguimos identificar a letra `c` e as letras `a`.

- O acento circunflexo (`^`) possui significado de negação quando presente dentro de colchetes. Com base no exemplo acima (`/[abc]/`), caso ele seja escrito como `/[^abc]/`, seu significado é exatamente o oposto, representando quaisquer caracteres exceto `a`, `b` e `c`. Além deste significado, ele é utilizado para representar o início de algum padrão no seu texto de origem. Na expressão regular `/^a.+/` temos o seguinte significado: "um texto que comece com a letra `a` seguido de qualquer caractere em qualquer quantidade". Desta forma, com esta expressão, conseguimos identificar `abcd`, `ab`, `abcdefghijk` e até mesmo `a9715263`. A única exigência, neste caso, é começar com a letra `a`, de maneira que no texto "as mulheres marcaram presença" seremos capaz de identificar toda a frase, mas já em "todas as mulheres marcaram presença" ela não identificará nada, visto que o texto começa com a letra `t`.

- Assim como o `^` representa o início de um texto o `$` indica o fim. A lógica é a mesma, se aplicando para todo o texto, e não apenas em ocorrências. Assim, para a expressão `/^a.+z$/` exige-se que o texto comece com a letra `a` e termine com a letra `z`. Nada além disso é permitido.

- Já os parênteses - `(` e `)` - representam grupos. Estes grupos, além de serem utilizados como no exemplo `/abaca(te|xi)/` podem ser utilizados para substituição de ocorrências, aumentando ainda mais a aplicação de expressão regulares. Como exemplo, dentro do texto "os estudantes adoram comer abacaxi depois do almoço", podemos substituir toda ocorrência de `abacaxi` por qualquer outra coisa, como `melão`. Para isso temos que formar um grupo com a palavra `abacaxi`, escrevendo a expressão regular da seguinte maneira: `/(abacaxi)/`. Assim, será reconhecida a palavra completa e esta poderá ser substituída por `melão`, ficando a frase "os estudantes adoram comer melão depois do almoço".

Em virtude da existência dos metacaracteres, caso seja necessária a representação de algum destes caracteres em sua forma pura, utiliza-se do caractere de escape `\` (barra

Tabela 2 – Classes do padrão POSIX.

POSIX	**Equivalência**	**Descrição**
`[:alnum:]`	`[A-Za-z0-9]`	Caracteres alfanuméricos
`[:alpha:]`	`[A-Za-z]`	Caracteres alfabéticos
`[:blank:]`	`[ \t]`	Espaço e tabulação (tab)
`[:cntrl:]`	`[\x00-\x1F\x7F]`	Caracteres de controle ASCII
`[:digit:]`	`[0-9]`	Dígitos
`[:graph:]`	`[\x21-\x7E]`	Caracteres visíveis
`[:lower:]`	`[a-z]`	Letras minúsculas
`[:print:]`	`[\x20-\x7E]`	Caracteres visíveis e espaço
`[:punct:]`	`[][!"#$%&'()*+,./:;<=>?@\^_`{|}~-]`	Caracteres de pontuação

invertida) antes do caractere desejado. Por exemplo, para escrever o símbolo +, literalmente, deve-se representá-lo por `\+`. Desta forma a expressão será interpretada como um "mais" e não como um caractere de repetição.

Assim como vimos a utilização de `0-9` para representação de dígitos, temos outras representações, tanto para indicar letras maiúsculas quanto minúsculas, utilizando de `A-Z` e `a-z`, respectivamente. Sendo assim, podemos identificar dígitos, letras minúsculas e maiúsculas com a expressão regular `/[A-Za-z0-9]+/`. Além disso, no padrão POSIX podemos representar conjuntos de letras e números de outras formas chamadas "classes", como pode ser verificado na Tabela 2.

No caso específico de extração de metadados a utilização de expressão regular é muito eficaz no reconhecimento de padrões, como é o caso do metadado e-mail, que possui um formato muito específico. Deste modo, podemos escrever uma expressão regular para identificar e-mails dentro de um texto a título de exemplo:

```
/[A-Za-z0-9-\.]{2,}@[A-Za-z0-9-]{2,}\.([A-Za-z0-9]{2,})+/
```

Assim, as aplicações de Expressões Regulares são infinitas, podendo ser capazes de identificar qualquer sequência que possa ser representada em forma de texto, como telefones, endereços, dentre outras.

2.1.3.2 Support Vector Machines (SVM)

Support Vector Machines (SVM) é uma técnica de *machine learning* que permite que um conjunto de dados seja analisado, possibilitando que padrões sejam então identificados, criando uma espécie de memória (VAPNIK; CORTES, 1995). Seu objetivo inicial era ser uma técnica de classificação de dados, por ser focada em reconhecimento de padrões através de análises matemáticas.

Esta técnica se baseia na redução de erros com base em um resultado de treinos consecutivos que permitem a criação de um padrão e estabelecem um aprendizado por parte da distância entre ocorrências (VAPNIK; CORTES, 1995). Todas as análises realizadas são mapeadas, permitindo que um registro histórico em forma matemática seja realizado, levando o algoritmo à possibilidade de diferenciação entre um resultado e outro, de forma numérica.

De acordo com Chieu (CHIEU; NG, 2002), sugere-se que a tarefa de extrair informação pode ser considerada um problema de classificação. Partindo deste pensamento foi que Han (HAN et al., 2003) decidiu utilizar técnicas de SVM para extração de metadados, utilizando das qualidades matemáticas do processo no reconhecimento de padrões, o que permitiu que novas descobertas fossem feitas, expandindo o estudo da extração de dados para um patamar mais amplo e elevado.

Várias ferramentas de extração se baseiam na utilização de SVM como técnica principal, visto sua eficiência no reconhecimento de padrões. Como descrito por (HAN et al., 2003) a utilização desta técnica é baseada na identificação de campos previamente selecionados no cabeçalho de um documento, do qual se deseja obter os metadados, por exemplo.

Esta técnica analisa diversos campos chamando-os de classes, e atribui a cada classe uma característica que a permite ser identificada. Deste modo, cada linha do cabeçalho do documento é classificada em uma ou mais classes, onde algumas fazem parte do padrão Dublin Core, detalhado na subseção 2.1.2.

Seymore et al. (SEYMORE; ROSENFIELD, 1999) definiu 15 (quinze) *tags* para esta definição do cabeçalho de um documento. Porém, destas 15 (quinze) *tags* definidas, somente 4 (quatro) correspondem ao padrão da Dublin Core e estão ilustradas na Tabela 3. Estas também foram as *tags* utilizadas por Han na extração de metadados dos cabeçalhos de artigos científicos (HAN et al., 2003).

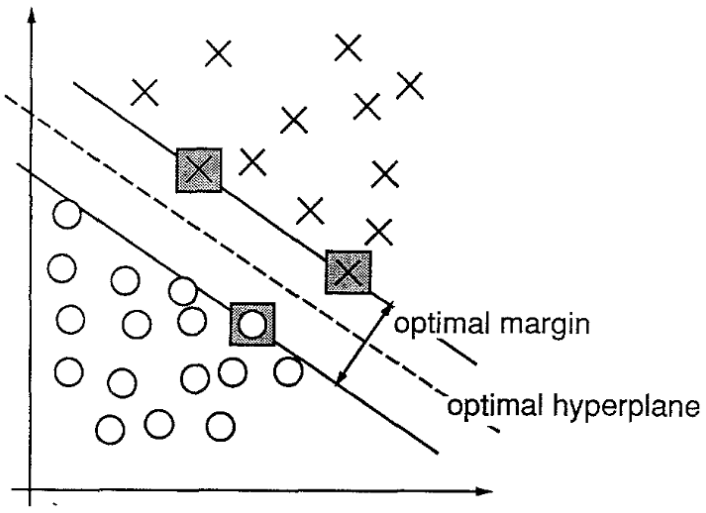
A ocorrência dos campos é mapeada em uma representação bidimensional, permitindo identificar visualmente os padrões encontrados na análise de cada classe. Desta forma, com a visualização do posicionamento de cada ocorrência é possível obter uma distância clara entre os pontos de reconhecimento, chamada pelo autor de *hyperplanes* (VAPNIK; CORTES, 1995). Por sua vez, estes pontos são marcados como sendo os "support vectors", que permitem que o *hyperplane* entre eles determine a divisão entre as classes de forma clara e eficaz, como pode ser visto na Tabela 2.1.3.2. Esta divisão permite então a distinção entre os metadados, diferenciando os elementos analisados pelo algoritmo.

Além desta análise são utilizadas comparações de palavras dentro de um contexto previamente selecionado. Assim foi criado um *cluster* de palavras comuns que facilita a identificação destas classes nos cabeçalhos analisados. Este *cluster* basicamente é composto

Tabela 3 – Relação de classes utilizadas e comparação com o padrão Dublin Core.

Classe (Tag)	**Referência Dublin Core**	**Descrição**
Title	Title	Título do artigo
Author	Creator	Nome do autor do documento
Affiliation		Afiliação do autor
Address		Endereço do autor
Note		Frases de reconhecimentos, *copyright*
Email		Endereço de e-mail do autor
Date		Data da publicação
Abstract Introdution	Description	A introdução do artigo
Phone		Telefone do autor
Keyword	Subject	As palavras-chaves do documento
Web		Endereço na Internet do autor
Degree		Associação com o grau acadêmico
Pubnum		Número da publicação do documento
Page		O final da página

Figura 2 – Distância representando a separação entre classes na técnica de SVM.



Fonte: (VAPNIK; CORTES, 1995)

de:

- Dicionário online padrão em sistemas Linux;
- 8.441 nomes e 19.613 sobrenomes;
- Sobrenomes chineses;
- Nome dos estados do Estados Unidos e das províncias canadenses;
- Nomes das cidades dos Estados Unidos;
- Nome dos países do mundo, de acordo com World Fact Book[1];
- Nome dos meses e suas respectivas abreviações.

Para cada uma das classes analisadas foram feitas correlações com o tipo de dado esperado, de maneira a permitir que endereços de e-mail, por exemplo, fossem extraídos com base em expressões regulares utilizadas em linguagens de programação.

Support Vector Machine é uma técnica conhecida principalmente por sua boa performance e habilidade para com grandes quantidades de dados, sendo por isso considerada uma boa solução para problemas de classificação. Por essa característica, sua principal funcionalidade é baseada na apresentação de um conjunto de opções, e a que mais se assemelha com a análise realizada é então classificada. Por isso, Han (HAN et al., 2003) decidiu utilizar este mesmo conceito de comparação para ser então aplicado na extração de dados, de modo a confrontar e comparar classes de metadados, a fim de identificá-los e então diferenciá-los.

Han também encontrou alguns desafios na diferenciação de campos, como é o caso dos múltiplos autores. Em alguns casos a diferenciação de autores, que fazem parte do mesmo campo, poderia estar inclusive em linhas ou grupos diferentes. Para isso foram utilizados alguns elementos para representar a separação dos nomes (*chunks*), como pontuações e a presença da palavra "and". Desta forma, os autores eram extraídos seguindo este padrão estipulado.

O resultado obtido pela utilização de SVM como técnica de extração de metadados foi bem relevante. O autor realizou uma comparação com a aplicação da técnica de Hidden Markov Models (HMM) - detalhada na subseção 2.1.3.3 - onde a SVM se mostrou mais eficaz na precisão da extração para algumas classes específicas, como é o caso dos títulos, autores e endereços, por exemplo. Para outras classes a utilização da técnica de HMM ainda demonstrou ser mais eficiente na extração de metadados.

[1] Disponível em <https://www.cia.gov/library/publications/the-world-factbook/index.html>

2.1.3.3 Hidden Markov Models (HMM)

A teoria básica de Markov foi conhecida próximo dos anos 80 por engenheiros e matemáticos, com grande aplicação inicialmente em processamento da fala, mas com vasta amplitude em outras áreas onde a descoberta de padrões pode ser aplicada (RABINER; JUANG, 1986).

O processo é baseado na identificação de modelos observáveis que representem e caracterizem a ocorrência de símbolos, ou seja, padrões, em determinados estados. Se um sinal foi observado ele pode ser utilizado para futuras referências, de acordo com o padrão.

Um exemplo prático citado por Rabiner e Juang (RABINER; JUANG, 1986) é o caso de uso do jogo "Cara e Coroa". Toma-se um observador em um quarto fechado com uma cortina totalmente isolada para outro cômodo qualquer. Este observador não consegue ver nada que acontece no outro cômodo, onde está presente uma outra pessoa jogando uma moeda pra cima, relatando sempre o resultado obtido (cara ou coroa). Neste caso o problema é construir um modelo Hidden Markov Model (HMM) para explicar ao observador a sequência dos resultados obtidos.

Neste exemplo, o primeiro caso é baseado tanto no estado de cada resultado (cara ou coroa) e em probabilidades matemáticas de ocorrência destes estados, neste caso, 0.5 (50%), ou seja, dois estados totalizando 100%. Assim desenha-se modelos onde os estados são representados com base nas inúmeras possibilidades existentes, levando inclusive em consideração a sequência dos últimos acontecimentos. Outra possibilidade seria a existência de duas moedas, o que daria ainda dois estados existentes, mas não em função da probabilidade de sair cara ou coroa, mas sim por serem consideradas duas moedas "justas", o que daria também uma probabilidade de 0.5 pra cada.

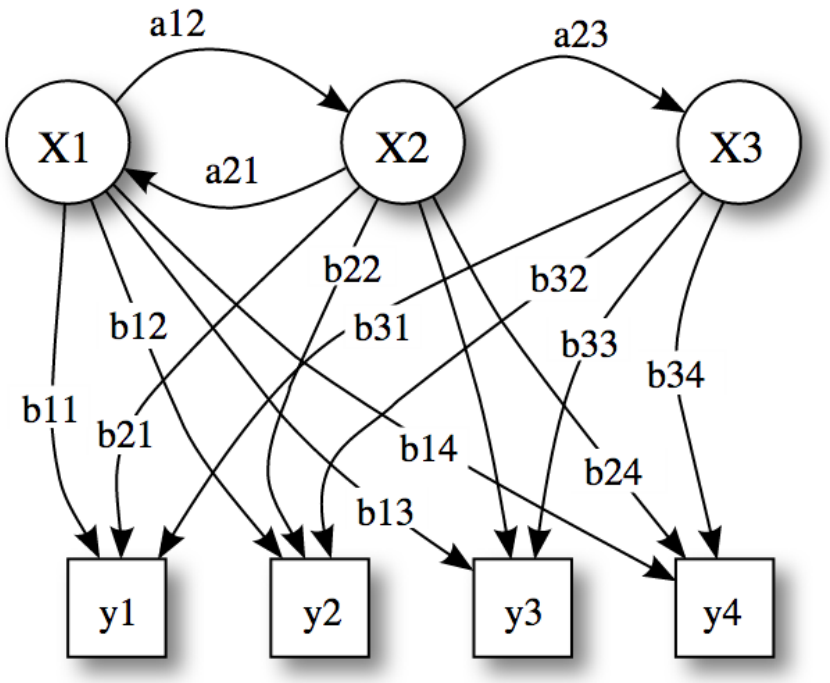
Neste último exemplo o grande detalhe do modelo é que este é oculto (*hidden*). Isso se deve ao fato de os dois estados, representados pelas duas moedas, serem totalmente independentes, o que não permite identificar qual moeda é a "justa" e então informar ao observador o resultado daquela rodada.

Por esta alteração de resultados e probabilidades, o fator decisivo na criação de cada modelo é a definição do número de estados que ele terá. Além disso, outro ponto que determina o sucesso do método é a utilização de um resultado anterior - *training dataset* -, ou seja, uma memória, um conjunto de informações pré-identificadas que permite ainda à associação dos estados e ocorrência dos símbolos (RABINER; JUANG, 1986).

Um HMM pode ser formado por um conjunto de elementos, que formarão toda a teoria e aplicação dos algoritmos dentro do processo:

1. Um número `N` de estados, onde `N` é um inteiro finito;
2. Um intervalo temporal `t`, que determina a entrada em um novo estado, através de

Figura 3 – Exemplo de modelo HMM, onde "X" são os estados, "Y" as observações possíveis, "A" as probabilidades de mudança de estado e "B" as saídas destas probabilidades.



Fonte: Wikipedia / Hidden Markov Model <http://goo.gl/pI3XUU>

uma transição de probabilidade entre eles, levando em consideração sempre o estado anterior;

3. Após cada transição o observador registra um `símbolo` de acordo com a distribuição de probabilidade, que por sua vez depende do estado atual do modelo.

A utilização dos resultados passados - *training dataset* - é muito importante para uma boa definição de um HMM, visto que permite adaptar os parâmetros do modelo para aquele conjunto de dados passados, que por sua vez fazem parte de um padrão já identificado.

Seguindo este padrão o HMM pode ser utilizado, por exemplo, para reconhecimento de palavras isoladas que, juntamente com a utilização de um vocabulário previamente selecionado permite a criação de modelos de reconhecimento. Cada palavra deste vocabulário seria um modelo HMM, permitindo que a palavra escolhida fosse a pertencente ao modelo com maior probabilidade encontrada.

Já no âmbito da extração da informação, o HMM pode ser aplicado conforme é apresentado por Seymore et al. (SEYMORE; ROSENFIELD, 1999), onde um modelo construído manualmente contendo múltiplos estados por campos (título, autor, etc), pode ser mais eficiente do que um modelo com somente um estado por campo.

Um dos pontos positivos deste modelo é que por ser baseado em estatística ele é muito bem empregado em problemas de linguagem natural, aliando os resultados positivos à excelente performance computacional. Como desvantagem desta técnica podemos citar o fato de, por ser baseada em estatística matemática, uma grande quantidade prévia de dados deve ser utilizada - a título de treino - para se obter padrões significativos para então ser aplicados de maneira final na criação do modelo.

Deste modo, para extração dos metadados, o HMM pode ser utilizado aplicando um marcador (*label*) em cada palavra do cabeçalho de um documento (artigo científico), relacionando cada palavra a uma classe, como título, autor, etc. Assim, uma opção seria a criação de um modelo com `N` estados, onde cada estado corresponderia a uma classe que se deseja extrair - por exemplo, o título do documento. Porém, no caso da existência de sequências ocultas (*hidden sequences*) - o que alteraria a seleção do estado seguinte - a utilização de vários estados para cada classe traria resultados melhores (SEYMORE; ROSENFIELD, 1999).

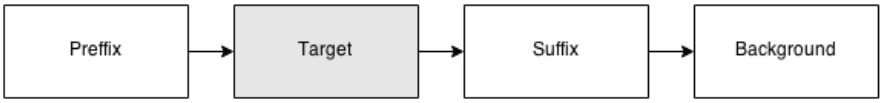
Para isso seria necessário entender melhor a estrutura do modelo de acordo com os dados de treino (*training dataset*), utilizando-se então de múltiplos estados para cada classe, o que traria resultados melhores. Deste modo, este *training dataset* permitiria a construção de um modelo chamado *maximally-specific model*. Neste modelo, cada palavra do *training dataset* seria associada com seu próprio estado, com transição para o estado correspondente à palavra seguinte (SEYMORE; ROSENFIELD, 1999).

Este modelo, segundo os autores, poderia ser usado como o ponto de partida de uma variedade de técnicas para junção de estados (*state merging techniques*). Desta forma os autores propõem duas formas de junção de estados que permitam a construção deste *maximally-specific model*:

1. ***Neighbor-merging:*** combina todos os estados que compartilham transições e possuem o mesmo nome de classe. Por exemplo, uma sequência de estados correspondentes ao título seriam juntados em apenas um estado. Assim, vários estados vizinhos com os mesmos nomes de classes seriam transformados em apenas um.

2. ***V-merging:*** combina quaisquer dois estados que possuem o mesmo nome de classe e compartilham transições “de” ou “para” um estado comum. Desde modo, ao invés de começar no estado inicial e decidir para qual estado correspondente ao título será feita a transição, esta técnica juntaria os estados “filhos” em um único estado, de maneira que somente uma transição do estado inicial para o estado de título iria existir. Desta forma, esta técnica poderia ser utilizada como modelo direto para a extração de dados, podendo também criar novas combinações de estados, implicando em uma futura melhora deste modelo.

Figura 4 – Estados utilizados por (ZHANG, 2001) em seu modelo HMM.



Fonte: O próprio autor.

Como o objetivo de Seymore et al. é extrair informações relevantes de cabeçalhos de artigos de Ciência da Computação, a área de análise nestes documentos se limita até o início da parte de introdução, ou ao final da primeira página, o que ocorrer primeiro.

O resumo (*abstract*) é extraído facilmente com a utilização de expressão regular. Algumas classes de palavras especiais são identificadas também através de expressão regular e então são transformadas em *tags* ou *tokens*, como `<EMAIL>` ou `<YEAR_NUMBER>`, por exemplo. Em todos estes casos, todos os acentos e informações de novas linhas (`n`) são removidos do texto.

Os resultados apontam como muito positiva a utilização de HMM para a extração de dados em cabeçalhos de artigos científicos. A precisão encontrada no experimento foi de 92,9% para todas as classes do cabeçalho e, mais especificamente, 97,2% para a extração dos autores. Também como resultado pode-se afirmar que modelos HMM com mais de um estado por classe são mais eficientes do que os que utilizam apenas um estado para cada classe analisada (SEYMORE; ROSENFIELD, 1999).

Uma outra utilização de HMM na extração de informação é descrita por (ZHANG, 2001), onde é realizado um experimento de extração de informação utilizando um conjunto de dados semi-estruturados, em formato HTML, contendo informações sobre restaurantes da cidade de Los Angeles.

Neste caso são estipulados quatro estados para o modelo: *Background*, *Prefix*, *Suffix* e *Target*. O *Target* é o estado responsável pela emissão do símbolo - chamado pelo autor de *token* - para o "campo-alvo". O *Preffix* e *Suffix* são estados que emitem símbolos que aparecem respectivamente antes e depois deste campo-alvo. Todos os demais símbolos são emitidos no estado *Background*. A relação entre os estados pode ser visualizada na Figura 4.

Os seguintes campos deveriam ser extraídos das informações dos restaurantes: *restaurant name* (nome do restaurante), *telephone number* (número de telefone), *hours* (horas) e *cuisine* (tipo de comida servida, como italiana, alemã, etc). Segundo Zhang (ZHANG, 2001) os campos *restaurant name* e *telephone number* deveriam ser mais fáceis de serem obtidos, visto que o nome do restaurante geralmente se encontra em destaque, com alguma diferenciação visual. Já o telefone possui um formato numérico, que permite mais facilmente uma identificação de um padrão. O campo *hours* também não foi complicado,

embora se tenha uma variedade muito grande na disposição desta informação. Já o campo *cousine* foi mais difícil de ser extraído, visto a diversidade que existe na forma de um restaurante especificar e/ou representar sua cozinha, tanto na utilização de palavras diferentes quanto na própria identificação do estilo do restaurante por si só.

Como resultados esperados, os campos *restaurant name* e *telephone number* obtiveram muito êxito em sua extração, com resultados realmente consideráveis. Já os campos *hours* e *cousine* não tiveram resultados muito satisfatórios, o que pode ser facilmente explicado em função da característica dos HMMs de utilizar como modelo resultado de aprendizados anteriores, os *training datasets* ou simplesmente *training data*. Como nestes dois campos há uma grande possibilidade de representação diferente, os resultados não foram tão eficazes, o que poderia ser resolvido com uma alteração no modelo HMM que permitisse que as palavras identificadas como sendo do campo *cousine* fossem capturadas de maneira mais proveitosa, que estão, neste modelo sugerido, isoladas no estado *Background*.

Em função dos resultados obtidos, pode-se considerar a performance de HMM na extração de informação muito positiva, visto as possibilidades de variação do modelo, que permite um resultado mais acurado e próximo dos objetivos reais (ZHANG, 2001). Além disso, a utilização de resultados passados garante um aprendizado muito importante para que o modelo seja estabelecido, o que garante ainda mais um ganho de eficiência na aplicação desta técnica.

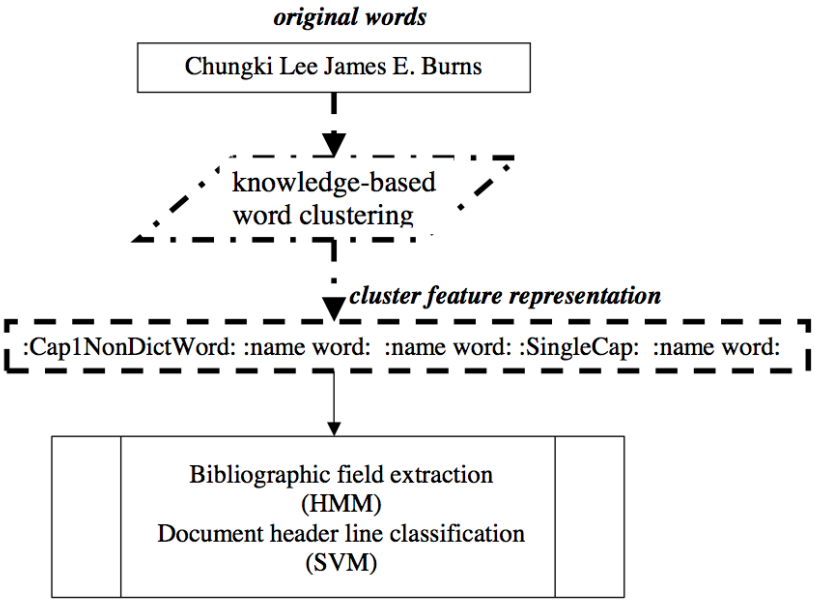
Além da utilização de HMM de forma natural, algumas variações de seu algoritmo são também citadas na literatura, como é o caso dos MEMMs (*Maximum Entropy Markov Models*) (BERGER; PIETRA; PIETRA, 1996). Nos MEMMs cada estado possui um modelo exponencial que utiliza as características de observação como entrada de dados (*input*) (LAFFERTY; MCCALLUM; PEREIRA, 2001). Estes modelos são baseados na técnica de HMM, se diferenciando na maneira como os estados se relacionam, bem como a relação entre suas transições, o que leva à citações de maneira independente por alguns autores, porém, com herança lógica dos conceitos de HMM.

2.1.3.4 Word Clustering

Técnicas de classificação de texto geralmente utilizam palavras extraídas como a principal fonte de recursos para a representação. Por outro lado, os *clusters* de palavras tem sido uma proposta eficaz para a redução da dimensionalidade e da dispersão, melhorando assim a performance desta classificação (HAN et al., 2005).

Basicamente o conceito de *clusters* compreende um conjunto de palavras formando um banco de dados de domínio (*domain database*), aliado a um conjunto de propriedades ortográficas de demais palavras dentro daquele contexto específico. A utilização destes *clusters*, juntamente com outras técnicas, tem mostrado um ganho de 6,6% na performance de classificação de elementos de um cabeçalho de artigo científico, e ainda 8,4% de ganho

Figura 5 – Workflow da extração de metadados usando *cluster* de palavras.



Fonte: (HAN et al., 2005)

de performance para a extração das referências destes documentos (HAN et al., 2005).

A utilização destes grupos de palavras demonstra uma relação entre textos semelhantes dentro de um determinado contexto, permitindo que a extração dos metadados possa ocorrer de maneira mais natural, com resultados mais eficazes.

Sendo assim, Han et al. apresentou uma ideia de um *cluster* de palavras para promover a extração de metadados de artigos da área de Ciência da Computação, indo de maneira relativamente contrária às propostas tradicionais, que baseiam, geralmente, apenas na ocorrência e estatísticas de palavras isoladas dentro do texto original.

Han et al. agrupou bases de dados de domínios diversos incluindo também propriedades ortográficas de palavras, com base em um conhecimento prévio de classes específicas, como autor, título, etc. Deste modo, palavras encontradas nos documentos vão sendo comparadas com palavras deste *cluster*, permitindo identificar, por grupos, características semelhantes de metadados. Para cada classe (metadado) cria-se um *cluster*, com suas palavras e propriedades ortográficas específicas.

Como exemplo, a palavra “Mary” faz parte do contexto de “nomes”. Portanto existe uma probabilidade maior de ela, juntamente com seu grupo de palavras ao redor, fazer parte da classe “autor”, por exemplo. Esta lógica é apresentada para outras classes, como “e-mail” por exemplo, que pode ser identificado, em sua grande maioria, com a presença do caractere “@”, o que levaria ao conceito de utilização de expressões regulares para encontrar padrões de ocorrências.

A utilização de *word clustering* tem demonstrado um custo computacional muito

baixo, o que pode ser considerado uma vantagem sobre demais técnicas computacionais (HAN et al., 2005).

Han et al. ainda utiliza da técnica de SVM (subseção 2.1.3.2) para classificação de linhas de um cabeçalho de um documento, tanto em função dos bons resultados obtidos quanto também pela boa performance apresentada. Deste modo, cada linha obtida se transforma em um vetor de palavras, que é comparado com seu respectivo *cluster*, melhorando os resultados de classificação quando da união destas duas técnicas em prol do mesmo objetivo.

Basicamente o *Word Clustering* se resume em 3 (três) etapas. A primeira etapa compreende a construção das bases de dados, como referenciado por Han (HAN et al., 2003), onde foram utilizadas também como bases externas os nomes apresentados em seu *cluster* (nomes de estados americanos, países, cidades, nomes de dicionários, nomes de pessoas, códigos postais, etc), de maneira a unir, não somente estas bases externas, mas também bases construídas dentro de um domínio específico, como palavras que pertencem a uma classe específica.

A segunda etapa é chamada de *Cluster Design*. Nesta etapa é onde os *clusters* são arquitetados, contemplando também as propriedades ortográficas das palavras, como se funcionassem, de maneira geral, como uma expressão regular de palavras, formando então este dicionário com base nas características apresentadas.

Já a terceira etapa é chamada de *Rule Design*, que resume-se basicamente na combinação de palavras em diferentes domínios, nas suas verificações ortográficas e na classificação delas no seu *cluster* correto. Por exemplo, nomes devem começar com a primeira letra maiúscula para então serem classificadas como pertencentes ao *cluster* "nomes".

Além disso, foi observada também a presença de certas palavras que faziam parte de diversos *clusters* ao mesmo tempo, o que permitiu que elas fossem classificadas em seu próprio grupo de palavras, tornando-se então independentes.

A representação através de *clusters*, utilizada por Han et al. (HAN et al., 2005), conseguiu reduzir um texto original de 11.223 palavras em um *cluster* de 588 elementos, permitindo ainda que ele fosse distribuído entre as classes distintas, o que tornou o processo muito menos trabalhoso mas, ao mesmo tempo, eficaz.

Como resultado, a utilização desta técnica de clusterização permitiu um ganho considerável de performance, além de contribuir para uma precisão maior dos resultados, visto que eles são apresentados nestas bases focadas em um domínio específico, perfazendo um contexto mais definido e com resultados mais garantidos.

Por outro lado a utilização desta técnica possui uma falha na semântica dos dados. No momento da classificação das classes, na separação e criação dos *clusters*, um dígito ou

conjunto deles é substituído pela identificação `:number:`. Isso faz com que ele se torne apenas um número qualquer, sem um contexto específico, ou seja, pode ser tanto uma referência a alguma página do documento ou até mesmo um mês de um ano, por exemplo.

2.1.3.5 Conditional Random Fields (CRFs)

CRFs é um framework proposto por Lafferty et al. (LAFFERTY; MCCALLUM; PEREIRA, 2001) criado para construir modelos probabilísticos e dados marcados em sequência (*label sequence data*), geralmente utilizados no reconhecimento de padrões e aprendizado de máquinas (*machine learning*).

Esta técnica oferece algumas vantagens ao se comparar com técnicas mais tradicionais, como HMM (subseção 2.1.3.3), se destacando a habilidade de diminuir pressupostos independentes feitos nestes modelos (LAFFERTY; MCCALLUM; PEREIRA, 2001).

Modelos baseados em HMM possuem uma fraqueza, que é o problema denominado *bias problem*. As transições de um dado estado somente competem entre elas, e não com todas as transições presentes no modelo. Elas são feitas com base probabilística de acordo com o estado inicial e a sequência de observação (*observation sequence*).

Desta forma, devido ao *bias problem*, em um caso extremo, um estado que tiver como opção de transição somente um outro estado, pode simplesmente ignorar a sequência de observação, o que traria efeitos contrários aos objetivos do processo inicial.

Laffarty et al. realiza comparações funcionais e práticas entre CRFs, HMM e MEMM (*Maximum Entropy Markov Models*), uma variação da técnica de HMM, detalhada na subseção 2.1.3.3. A diferença crítica entre CRFs e MEMMs é que MEMM utiliza como probabilidade de ocorrência de um próximo estado modelos exponenciais para cada um deles, enquanto a técnica de CRF possui um modelo exponencial único para toda a sequência de *labels*, com base na sequência de observação. Segundo Laffarty et al. (LAFFERTY; MCCALLUM; PEREIRA, 2001) pode pensar na CRF como um modelo de estado finito sem normalização das probabilidades de transição.

Uma das vantagens de se utilizar CRFs sobre HMM é que ela absorve todas as qualidades deste último, mas com a particularidade de resolver o *bias problem*. Outra grande vantagem ao ser comparada com HMMs e MEMMs é que a CRF possui resultados melhores quando a distribuição dos dados possui grande dependência do modelo, o que geralmente ocorre em casos mais práticos.

Um exemplo para entender o *bias problem* foi apresentado por Lafferty et al. (LAFFERTY; MCCALLUM; PEREIRA, 2001). Ele propõe um modelo cujo objetivo é distinguir duas palavras: `rib` e `rob`, tendo como sequência de observação as letras `r i b`. O problema é identificado quando uma das duas palavras é mais comum no *training set*, o que acarreta nas transições do estado inicial preferirem suas transições correspondentes,

o que acaba sempre na vitória da palavra relacionada aquele estado.

Visando a comprovação e reconhecimento da eficiência da técnica de CRF na extração de informação, foram aplicados dois tipos de experimentos:

1. Verificação direta do *bias problem*;
2. Geração de dados utilizando HMM aleatórios.

Os resultados apontam que HMMs superam MEMMs em virtude do *bias problem*. Por sua vez os CRFs superam os HMMs, sendo então considerada a CRF a melhor técnica a ser empregada com base no *training set* utilizado (LAFFERTY; MCCALLUM; PEREIRA, 2001). Outro ganho apresentado pode ser verificado ao agregar algumas características ortográficas à utilização de CRFs, aumentando o poder destes modelos condicionais.

De modo geral as CRFs utilizam a mesma lógica que modelos baseados em Markov (HMMs e MEMMs), se diferenciando nos aspectos probabilísticos para com as transições entre os estados, acarretando em resultados comparativamente melhores. Por corresponder a uma "máquina" de estado finito, a técnica é muito aplicável para a função de classificação sequencial, o que permite que ainda seja treinada para se obter os melhores resultados probabilísticos (PENG; MCCALLUM, 2004). Nas CRFs as transições de estado são também representadas como *features* por alguns autores.

Uma outra vantagem da técnica de CRFs - assim como dos modelos *maximum entropy* - é que eles facilmente proporcionam o uso de características arbitrárias dos dados de entrada. As CRFs são utilizadas também em marcação e classificação de dados sequenciais, com uma ordem determinada, como linguagem natural, sequências biológicas (como os genes) ou estados computacionais.

Sua aplicação na extração de metadados foi apresentada por Peng et al. (PENG; MCCALLUM, 2004), como uma maneira eficaz de extrair metadados em cabeçalhos e referências de artigos científicos. Deste modo, através da identificação destes padrões sequenciais pode-se determinar os tipos de dados existentes e então identificá-los, seguindo uma lógica/ordem pré-determinada.

Peng et al. apresenta resultados desta extração utilizando Conditional Random Fields (CRF) e aponta também algumas questões acerca da utilização testa técnica para este tipo de atividade. Os autores comparam as CRFs com técnicas de HMM (subseção 2.1.3.3) e SVM (subseção 2.1.3.2), mencionando que a forma de trabalhar com CRF aponta parte das vantagens destas duas técnicas, destacando a junção entre as sequências e características dependentes, mas ao mesmo tempo arbitrárias. Ainda assim, a utilização de CRFs para a tarefa de extração de metadados possui melhoras, segundo os autores, significativas, ao se comparar com as demais técnicas de SVM e HMM.

Os autores definem quatro diferentes transições de estado para diferentes classes, em ordem diferente das técnicas derivadas de Markov (HMM):

1. ***First-order:*** os dados de entrada são examinados no contexto de somente um estado;
2. ***First-order + transitions:*** são adicionados alguns parâmetros correspondentes às transições;
3. ***Second-order:*** as entradas são examinadas no contexto dos estados atual e anterior;
4. ***Third-order:*** os dados de entrada são examinados no contexto do estado atual e de dois estados anteriores.

Para a extração destes dados Peng et al. também considera como sendo o cabeçalho de um *paper* como sendo a parte inicial do documento até o início da introdução, ou somente a primeira página, o que ocorrer primeiro. Além disso, os autores consideram como os campos a serem analisados os mesmos 15 (quinze) que foram definidos anteriormente por Seymore et al. (SEYMORE; ROSENFIELD, 1999).

Do *dataset* para extração dos metadados de cabeçalhos utilizado foram analisados 935 (novecentos e trinta e cinco) documentos. Destes, 500 (quinhentos) foram utilizados para compor o *training set* e os outros 435 (quatrocentos e trinta e cinco) para fins de teste, unicamente. Já do *dataset* utilizado para extração das referências foram analisados 500 (quinhentos) documentos, dos quais 350 (trezentos e cinquenta) foram utilizados para *training set* e os demais 150 (cento e cinquenta) também para testes.

Para fins de resultados comparativos foram utilizadas três métricas:

1. ***Overall word accuracy:*** é uma métrica que utiliza a porcentagem de palavras das quais os nomes (*labels*) previstos são exatamente seus valores reais. Esta métrica favorece aqueles campos que possuem muitas palavras, como por exemplo a introdução (*abstract*);
2. ***Average F-measure (F1):*** esta métrica se baseia na exatidão das ocorrências, considerando tanto a precisão quanto a memória de todos os campos (*fields*). Esta métrica favorece campos com poucas palavras, visto sua característica mais importante focar na exatidão dos resultados.
3. ***Whole instance accuracy:*** nesta métrica uma "instância" é considerada como sendo todo um cabeçalho ou referência, de maneira integral. Desta forma esta métrica utiliza-se da porcentagem de instâncias das quais cada palavra é corretamente associada.

Tabela 4 – Resultados de extração para CRFs após análise do *dataset* com cabeçalhos (PENG; MCCALLUM, 2004).

	HMM		CRF		SVM	
Overall acc.	93.1%		**98.3%**		92.9%	
Instance acc.	4.13%		**73.3%**		-	
	acc.	F1	acc.	F1	acc.	F1
Title	98.2	82.2	99.7	**97.1**	98.9	96.5
Author	98.7	81.0	99.8	**97.5**	99.3	97.2
Affiliation	98.3	85.1	99.7	**97.0**	98.1	93.8
Address	99.1	84.8	99.7	**95.8**	99.1	94.7
Note	97.8	81.4	98.8	**91.2**	95.5	81.6
Email	99.9	92.5	99.9	**95.3**	99.6	91.7
Date	99.8	80.6	99.9	**95.0**	99.7	90.2
Abstract	97.1	98.0	99.6	**99.7**	97.5	93.8
Phone	99.8	53.8	99.9	**97.9**	99.9	92.4
Keyword	98.7	40.6	99.7	**88.8**	99.2	88.5
Web	99.9	68.6	99.9	**94.1**	99.9	92.4
Degree	99.5	68.8	99.8	**84.9**	99.5	70.1
Pubnum	99.8	64.2	99.9	86.6	99.9	**89.2**
Average F1		75.6		**93.9**		89.7

Conforme pode-se observar na Tabela 4 e Tabela 5 a utilização de CRFs para extração de metadados - tanto de cabeçalhos como de referências - teve resultados melhores do que a utilização de HMM (subseção 2.1.3.3), aumentando a performance em praticamente todos os campos, chegando a uma precisão de 98,3% (*overall accuracy*).

Pode-se observar também que a utilização de modelos HMM para precisão de palavras (campos com poucas palavras, onde a precisão é muito mais importante) é, de modo geral, pior do que quando utiliza-se de SVMs (coluna F1 da Tabela 4). Por outro lado, no campo *abstract* HHM possui performance bem melhor que quando utilizado SVM (98% contra 93,8%).

Com base nos resultados apresentados pode-se considerar que o trabalho de Peng (PENG; MCCALLUM, 2004) contribuiu para o estado da arte, melhorando a performance na extração de metadados em artigos científicos. Assim, a utilização de CRFs mostra-se muito eficaz por reduzir consideravelmente os erros encontrados, aumentando o sucesso da aplicação desta técnica neste contexto.

Tabela 5 – Resultados de extração para CRFs após análise do *dataset* com referências (PENG; MCCALLUM, 2004).

	HMM		CRF	
Overall acc.	85.1%		**95.37%**	
Instance acc.	10%		**77.33%**	
	acc.	F1	acc.	F1
Author	96.8	92.7	99.9	**99.4**
Booktitle	94.4	0.85	97.7	**93.7**
Date	99.7	96.9	99.8	**98.9**
Editor	98.9	70.8	99.5	**87.7**
Institution	98.5	72.3	99.7	**94.0**
Journal	96.6	67.7	99.1	**91.3**
Location	99.1	81.8	99.3	**87.2**
Note	99.2	50.9	99.7	**80.8**
Pages	98.1	72.9	99.9	**98.6**
Publisher	99.4	**79.2**	99.4	76.1
Tech	98.8	74.9	99.4	**86.7**
Title	92.2	87.2	98.9	**98.3**
Volume	98.6	75.8	99.9	**97.8**
Average F1		77.6%		**91.5%**

2.1.3.6 Outras Técnicas e Conceitos

2.2 Revisão de Estado da Arte na Extração de Metadados

Técnicas de *machine learning* não são novas, porém suas aplicações são inúmeras. Estas técnicas eram então utilizadas apenas para classificação de palavras e processamento de linguagem natural foram sendo utilizadas em diversas outras áreas, resolvendo problemas distintos e inovando em soluções, como é o caso da extração de informação.

O aperfeiçoamento da aplicação destas técnicas para a área de *information extraction* acarretou em um resultado muito positivo, melhorando cada dia mais a precisão dos resultados obtidos. Assim, algoritmos foram/são alterados e otimizados visando obter maiores performances e resultados cada vez melhores.

Pode-se perceber que cada técnica de classificação e/ou extração de informação possui suas particularidades e características diferentes. Assim, nota-se que cada uma possui uma melhor aplicação para extração de determinado campo, ou conjunto deles. Pequenas modificações são necessárias para que todo o contexto de um artigo científico seja mapeado com sucesso, sendo às vezes necessária a utilização de diversas técnicas em um único projeto.

Para isso, diversas comparações destas técnicas foram feitas. Algumas, inclusive já

mencionadas ao longo deste trabalho, foram feitas no surgimento de uma nova técnica, onde esta era comparada com técnicas anteriores, a fim de comprovar sua aplicabilidade e aumento de performance. Porém, este tipo de comparação torna-se tendenciosa, visto que os campos analisados, bem como os *datasets* utilizados, tendenciam para resultados positivos da nova técnica apresentada. Além disso, geralmente os *datasets* utilizados são focados em documentos da área de Ciência da Computação, e tende a seguir um padrão visual já estipulado pelos grandes eventos da área.

Em função deste problema alguns autores realizam comparações de técnicas, isoladamente ou com utilização de ferramentas, para analisar um grupo de documentos reais, objetivando um resultado mais próximo da realidade e, consequentemente, mais passível de erros.

Granitzer et al. (GRANITZER et al., 2012a) comparam o uso de Conditional Random Fields (subseção 2.1.3.5) e Support Vector Machines (subseção 2.1.3.2) na extração de metadados em artigos científicos, utilizando para tal *datasets* multidisciplinares, como *Mendeley* e *e-Prints*, que fazem parte de um grupo social de *datasets*, permitindo a contribuição global, unindo pesquisadores de diversas áreas do conhecimento, que constroem um repositório diversificado e recheado de novos desafios reais.

Em função da existências destes repositórios a informação fica cada vez mais descentralizada e, portanto, são necessários cada vez mais mecanismos inteligentes para garantir a alta qualidade dos metadados extraídos e armazenados de cada documento (GRANITZER et al., 2012a). A combinação destes mecanismos com pós-processamento inteligente contribui para o processo, elevando a qualidade do resultado final encontrado.

Visando um reconhecimento maior dos resultados obtidos Granitzer et al. realizam comparações dos resultados com a aplicação de três ferramentas: ParsCit (subseção 2.3.6), Mendeley Desktop (subseção 2.3.3) e sua própria ferramenta baseada em CRFs, no qual se referem como “Layout-based CRF”.

Este trabalho (GRANITZER et al., 2012a) é uma continuação do trabalho anteriormente realizado (GRANITZER et al., 2012b), onde as ferramentas Mendeley Desktop (subseção 2.3.3) e ParsCit (subseção 2.3.6) foram comparadas, utilizando, porém, um *dataset* menor. Desta vez foi incluída a técnica de Conditional Random Fields (CRF) na comparação que, de acordo com a literatura mencionada neste trabalho, possui resultados melhores do que a utilização de Hidden Markov Models (HMMs - subseção 2.1.3.3).

Foram analisadas 20.672 publicações do *dataset* Mendeley, abrangendo diversas áreas do conhecimento como ciência de modo geral, ciência da computação, biomedicina e física. Já no *dataset* e-Prints foram analisadas 2.452 publicações de áreas como física, medicina e diversas outras pertencentes ao IEEE, ligadas geralmente à área de computação.

Foi relatado que umas das etapas principais para um bom processamento e uma

extração mais precisa é o "pós-processamento". A aplicação de diversas análises (inclusive matemáticas) nos resultados extraídos garante um conjunto de dados de saída muito melhor e mais acurado. Segundo Granitzer et al. (GRANITZER et al., 2012a) estas tarefas são mais de responsabilidade do setor de engenharia, onde diversos detalhes devem ser observados e diversos algoritmos aplicados.

De modo geral, os resultados obtidos pelas ferramentas Mendeley e ParsCit foram considerados ruins para os grupos pertencentes à área médica (GRANITZER et al., 2012a). Porém estes resultados poderiam ser melhorados com um novo treino dos dados (*training set*), com aplicações específicas para aquela área do conhecimento, no caso, a medicina. Outro ponto interessante observado foi que a extração dos títulos ocorreu mais facilmente do que a extração dos autores, que possui sua performance muito variada em função da técnica que é utilizada.

Para os outros grupos, os resultados foram bem positivos, observando uma pequena diferença entre os números obtidos com as ferramentas Mendeley e ParsCit, que foram muito superiores aos resultados da implementação CRF dos autores. Isso se deve ao fato das duas ferramentas anteriores utilizarem de pós-processamento dos dados, o que garantiu aos resultados obtidos uma precisão muito maior (GRANITZER et al., 2012a).

De acordo com Gratnizer et al., mesmo a técnica de CRF possuindo melhores modelos de extração de informação, foi-se observado que a técnica de SVM utilizada pelo Mendeley Desktop supera a CRF no que tange extração de metadados (GRANITZER et al., 2012a).

A ferramenta ParsCit de modo geral não teve resultados melhores do que o modelo do Mendeley Desktop. Para a área de Ciência da Computação, mais especificamente na base de dados do IEEE, ambas as ferramentas tiveram excelentes resultados. Já na base de dados da ACM o Mendeley obteve melhores resultados do que o ParsCit (GRANITZER et al., 2012a).

2.3 Ferramentas de Extração de Metadados

Algumas ferramentas baseiam sua extração em padrões pré-definidos de maneira a identificar dados relevantes dentro de uma região específica dos artigos, facilitando a procura e consequentemente aumentando a velocidade no resultado.

Comentário: Citar trabalho do Prof. Renato Rocha

Estas ferramentas geralmente permitem uma variedade muito grande de layouts, embora nem todos já estejam previamente definidos. Geralmente novos layouts são inseridos

em novas versões ou até mesmo por contribuições das mais diversas, como é o caso dos projetos de código livre, os chamados projetos *open source*.

Abaixo segue uma relação das principais ferramentas relacionados à área de extração de metadados de artigos científicos, com informações sobre seu funcionamento e as técnicas que são utilizadas.

2.3.1 Cermine

Uma destas ferramentas é o recente CERMINE (TKACZYK et al., 2014), uma biblioteca *open source* desenvolvida na linguagem de programação Java que permite que sejam extraídos os metadados de artigos científicos em formato digital PDF, oferecendo ainda a possibilidade de cruzamento de dados por meio de referências e títulos, permitindo assim identificar citações bem como a relevância de um determinado documento.

O CERMINE ainda possui um mecanismo de aprendizagem da própria máquina, permitindo que, na medida que dados forem sendo alterados, ele consiga absorver os detalhes e permitir assim uma mudança de sua maneira de extrair os dados. Deste modo ele permite que seja adaptado para novos padrões de layouts, o que permite de maneira geral que uma grande gama de modelos seja então abrangida.

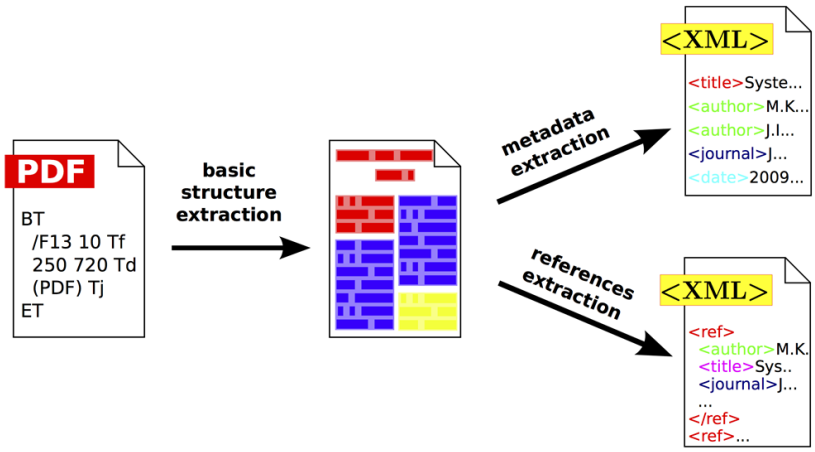
Seu grande diferencial em comparação com as demais ferramentas é que ele não somente extrai os metadados de um artigo, mas sim analisa todo o seu conteúdo, incluindo citações a outros documentos, que podem ser facilmente cruzados por meio de informações como título e autor(es).

Seu mecanismo considera somente arquivos PDF com texto gerado de maneira pura, sem a utilização de imagens. Basicamente ele considera regiões, linhas e páginas como pontos estratégicos para a extração de informações. As bases destas regiões possuem padrões que são utilizados juntamente com técnicas de SVM (HAN et al., 2003). Com base nisso ele condensa um layout onde as informações geralmente estão dispostas, permitindo assim que em um determinado local do arquivo esteja, provavelmente, o título e o nome dos autores.

Com estas regiões definidas o CERMINE extrai as informações com base em padrões preestabelecidos, de maneira a gerar então sua saída para os metadados e referências encontradas. A saída trabalhada pelo projeto é no formato XML, permitindo assim que possa ser compartilhado com outros sistemas por possuir uma leitura semântica e ao mesmo tempo fácil de ser interpretada pelas linguagens de programação. A Figura 6 demonstra como o processo de extração do CERMINE funciona.

Com o mapeamento definido ele identifica regiões de acordo com seu conteúdo, as quais ele chama de *zones*. Esta regiões são determinadas a fim de extrair as informações relevantes para cada uma, de maneira a separar, por exemplo, a área destinada aos

Figura 6 – CERMINE Extraction Workflow



Fonte: (TKACZYK et al., 2014)

metadados do arquivo. O CERMINE divide estas *zones* da seguinte maneira:

- **Metadata:** É a região mais ao alto do documento, onde obtém os metadados, que seriam o resumo, *bib_info*, tipo, título, afiliação, autores, datas, editores e palavras-chaves.
- **References:** Região responsável por identificas detalhes de referências que foram utilizadas no artigo, como título e autores, por exemplo.
- **Body:** O texto geral do artigo, incluindo equações, imagens e tabelas.
- **Other:** Outros detalhes menos significantes semanticamente, como número das páginas, dentro outros dados.

A extração das referências abrange também seus próprios metadados. Tanto no texto corrido (*Body*) quanto na lista de referências do artigo o *parser* do CERMINE analisa o conteúdo linha a linha, permitindo uma extração de dados mais eficaz. Das referências são extraídos os seguintes dados: autor, título, nome do *journal*, volume, *issue*, páginas, *publisher*, localização e o ano.

2.3.2 TeamBeam

Outra ferramenta de destaque é o TeamBeam (KERN et al., 2012), cuja base ideológica possui objetivos bem sociais, de contribuir com o compartilhamento de conhecimento. Basicamente o objetivo do projeto é extrair metadados de artigos científicos, porém focado apenas nestes, de maneira a extrair título, nome do *journal*, resumo e informações sobre os autores, como nome, endereço de e-mail e afiliações.

O projeto também é de código livre (*open source*) e é baseado na extração de pequenos blocos de texto. A manipulação dos arquivos PDF é feita pela biblioteca PDFBox [2], que fornece meios eficazes de extrair textos com base nas regiões desejadas.

O TeamBeam utiliza o algoritmo de *Maximum Entropy* (BERGER; PIETRA; PIETRA, 1996), que utiliza basicamente de tarefas de classificação sequencial como ferramenta principal para obtenção de padrões. A base deste algoritmo está na utilização de CRFs (subseção 2.1.3.5), principalmente no que diz respeito à extração dos metadados (PENG; MCCALLUM, 2004).

O processo de extração é feito basicamente em duas etapas. A primeira é a etapa de classificação de blocos de texto (*text block classification*), onde geralmente já é possível obter algum dado concreto como resultado. Nesta etapa o objetivo é associar certos blocos de texto a um dos seguintes marcadores: *Title Block*; *Sub-Title Block*; *Journal Block*; *Abstract Block*; *Author Block*; *E-Mail Block*; *Affiliation Block*; *Author-Mixed Block*; e *Other Block*.

Dependendo do layout do artigo alguns metadados podem vir divididos em blocos de texto diferentes, necessitando de um processamento posterior, como é o caso, geralmente, dos blocos com informações sobre os autores. Neste caso também é realizada a etapa de classificação de token (*token classification*), que se resume na classificação de palavras individualmente de acordo com um dos seguintes marcadores: *Given Name*; *Middle Name*; *Surname*; *Index*; *Separator*; *E-Mail*; *Affiliation-Start*; *Affiliation*; e *Other*.

Kern et al. defendem a ideia de excelentes resultados do TeamBeam ao ser comparado com outros projetos. Este fato é dado em virtude das características que são levadas em consideração no processamento da ferramenta, utilizando de dicionários, informações de layout e modelos de linguagem.

A fim de analisar os resultados obtidos com a aplicação das técnicas descritas no TeamBeam, os autores comparam as técnicas utilizadas com outras técnicas de outras ferramentas, que utilizam processos diferentes de análises.

Para fins de comparação de resultados, os autores citam as ferramentas ParsCit (subseção 2.3.6) e Mendeley Desktop (subseção 2.3.3). Como conclusão ele compara estas três ferramentas separadamente, por não abrangerem todos os detalhes que o TeamBeam possui, o que tornaria a comparação desleal.

Assim, ele chega à conclusão que em virtude das diferentes formas de processamento dos dados feitas por cada umas das ferramentas, chega-se a resultados mais precisos para cada campo extraído. As ferramentas baseadas em dicionário apresentam melhores resultados para extração de autores, visto que baseiam-se em *data-sets*[3] já consolidados.

[2] Biblioteca de manipulação de arquivos PDF mantida pela Fundação Apache. Disponível em <https://pdfbox.apache.org/>

[3] Bases de dados com nomes dos autores mais citados e outras informações já catalogadas que são

2.3.3 Mendeley

Mendeley é um dos projetos mais complexos e completos que existem na atualidade. Além de uma poderosa ferramenta na extração de metadados o projeto se transformou em uma plataforma completa para pesquisadores. Sua pesquisa se iniciou em novembro de 2007 por três estudantes alemães de doutorado, tendo sua primeira versão lançada em agosto de 2008. Em 2013 o projeto foi vendido para uma empresa, que passou a liderar o processo de criação e desenvolvimento. Atualmente o Mendeley é um projeto amplo, porém estritamente comercial, que possui uma plataforma de pesquisa e gestão de documentos como seu principal produto ofertado.

Os serviços oferecidos possuem modalidades grátis e paga, de acordo com a necessidade de cada usuário. Assim como o CiteULike (subseção 2.3.4) o projeto se tornou referência no meio acadêmico, sendo utilizado e comentado por diversos pesquisadores. Em virtude de seu objetivo comercial a ferramenta não possui código aberto atualmente, o que não permite seu uso sem ser através do serviço oferecido pela empresa.

A ferramenta pode ser utilizada via Web, como aplicativo *desktop*, para *tablets* ou *smartphones*. Visto seu aspecto social e suas inúmeras funcionalidades o projeto se tornou uma rede social acadêmica, tanto do ponto de vista de organização de documentos até mesmo para interação entre pesquisadores. A ferramenta permite que usuários façam *upload* de artigos em formato PDF para que sejam armazenados em sua biblioteca pessoal. Estes arquivos são analisados e seus metadados são extraídos e enviados para os servidores centrais da ferramenta.

Além de organizar sua biblioteca pessoal é possível também fazer anotações em documentos, destacar partes de texto, exportar citações, gerenciar documentos e compartilhar informações com outros pesquisadores, o que torna a plataforma extremamente completa e funcional.

A pesquisa iniciada pelos três estudantes de doutorado levou em consideração técnicas de *machine learning* e *metadata extraction*, como SVM, por exemplo (GRANITZER et al., 2012a). A ferramenta utiliza um modelo de SVM de dois estágios (*two-stages SVM*) (HAN et al., 2003) no que tange a extração dos metadados, obtendo uma grande exatidão dos resultados.

Pela sua complexidade e por abranger uma gama de funcionalidades muito grande, s serviço foi comparado à rede Last.FM (HENNING; REICHELT, 2008). Assim como a rede social de música o Mendeley permite que o usuário organize sua biblioteca digital de pesquisa de maneira bem eficaz, podendo compartilhar, distribuir e realizar diversas outras funções, assim como é possível na plataforma musical Last.FM.

Por ser um serviço a bastante tempo no mercado o Mendeley funciona também

armazenadas para consulta pública.

como um grande *dataset*, permitindo que parte de sua coleção de artigos seja utilizada para fins de pesquisa, assim como ocorre com o CiteULike (subseção 2.3.4), por exemplo. Isso facilita muito a descoberta de novas ferramentas e técnicas pois permite que uma quantidade muito grande de documentos seja analisada de maneira semi-estruturada, gerando então novos conhecimentos na área.

Após a venda da plataforma o projeto teve seu foco muito mais voltado ao comercial, ao contrário do que era anteriormente. A utilização de seus dados ficou restrita e a geração de receita se transformou no principal objetivo da companhia, o que trouxe algumas desvantagens para a pesquisa livre como um todo.

Algumas das funcionalidades do Mendeley:

- Multiplataforma, podendo funcionar tanto na Web como em Windows, Mac, Linux, iPhone e iPad;
- Extração de metadados de artigos em formato PDF (ponto de pesquisa deste trabalho);
- Centralização de sua biblioteca digital, sendo a mesma disponibilizada em qualquer plataforma, pois fica disponível na nuvem (*cloud computing*);
- Visualizador de PDF embutido, o que permite marcações em texto e inclusão de notas personalizadas;
- Pesquisa completa da base de dados dos artigos existentes;
- Inclusão de *tags* nos documentos, permitindo uma categorização dos mesmos dentro da biblioteca pessoal de cada usuário;
- Permite que citações e bibliografias sejam exportas em formato Microsoft Word, LibreOffice e LaTeX;
- Criação de grupos públicos e privados entre pesquisadores, permitindo compartilhamento de conteúdo entre os usuários;
- É uma Rede Social completa, permitindo *posts*, comentários, páginas de perfil, etc;
- Provém uma série de estatísticas com base nos documentos pertencentes à sua biblioteca digital.

Embora o sucesso do projeto seja muito grande e suas funcionalidades permitam uma série de funções a serem utilizadas pelos pesquisadores, em virtude do foco deste trabalho, o Mendeley se torna uma ferramenta pouco útil, visto que não possui o seu código aberto e não permite que seu código seja instalado em outras máquinas para que

sua funcionalidade de extração de metadados seja testada em separado. Mesmo assim, o projeto é bastante promissor e garante uma gama de funções que todo pesquisador necessita, além de organizar de maneira bem eficaz os artigos e demais documentos que fazem parte de um bom trabalho de pesquisa.

2.3.4 CiteULike

Uma outra ferramenta de destaque é o CiteULike (EMAMY; CAMERON, 2007) e pode ser acessada em <http://citeulike.org>. Ela é uma fusão de dois serviços: um serviço de *bookmarking* via Internet, bem como também um sistema de gestão de referências bibliográficas. A união destas duas funcionalidades permitiu um compartilhamento muito grande de conhecimento entre pesquisadores, de modo a promover uma disseminação de informação com o envio de links contendo *papers* de pesquisa.

O projeto foi criado em 2004 por Richard Cameron, mas em 2006 a empresa Oversity Ltd. assumiu seu controle e manutenção (GALLAGHER, 2008). O CiteULike pode ser acessado através de um navegador e fica disponível na Internet, de fácil acesso. O projeto ainda encontra-se ativo e em constante evolução, tanto por parte da equipe que o mantém, como por parte de seus usuários.

O projeto é livre desde seu lançamento e não tem nenhum custo para seus utilizadores. Seu objetivo inicial surgiu de um problema para busca de material bibliográfico. Deste modo o CiteULike foi criado para remover este espaço vazio entre pesquisadores, além de permitir que eles compartilhem bibliografias de maneira simplificada, podendo inclusive contribuir para grupos de pesquisa, por exemplo.

Um dos detalhes desta ferramenta é que ela funciona com base em fontes externas de dados. Quando um usuário encontra na Internet um artigo que lhe é interessante ele pode adicioná-lo à sua lista pessoal (*bookmarking*). Assim, o CiteULike automaticamente visita este endereço informado e realiza uma extração de dados dos detalhes das citações do artigo, salvando o link para este *paper* em sua base de dados (EMAMY; CAMERON, 2007). Além disso, ele permite que os metadados deste artigo também sejam extraídos, como título, autores, *journal name*, número de páginas, etc, com base no link fornecido, seguinte para tal o padrão visual daquele *publisher*.

Como se pode ver, o CiteULike preserva a informação livre. Ele somente coleta *papers* que já estão disponíveis na Internet, com base em um link fornecido por um usuário. Outro ponto de interesse é que a ferramenta também analisa informações de citação para que possa criar um link reverso para os *papers* já analisados e existentes em sua base. Deste modo, de posse de um link para um determinado artigo o CiteULike consegue analisar quais outros artigos este artigo cita, permitindo criar uma rede de citações entre documentos, o que facilita a disseminação de conhecimento e permite que pesquisadores

distintos possam usufruir deste conjunto de informações previamente analisadas.

Outro detalhe importante é o fato do projeto ser ao mesmo tempo um serviço social. Além do armazenamento destes artigos em forma de links, ao adicionar um artigo em sua lista pessoal, o usuário também pode associar *tags* àquele artigo. Isso permite uma categorização muito particular e ainda permite que artigos sejam procurados bem facilmente com bases nestas *tags* fornecidas. De certa forma, outros pesquisadores podem se beneficiar destas informações, pois elas funcionam como uma folksonomia específica para aquele determinado domínio, que pode ser de interesse de outros pesquisadores também. Outro detalhe importante de destaque do ponto social da ferramenta é que ela permite um sistema de votação nos artigos. Cada artigo pode ser votado por usuários, criando-se um ranking dos artigos mais interessantes, tendo um local de destaque na página inicial da ferramenta.

Um grande detalhe do CiteULike é que ele permite que suas informações sejam utilizadas para análise de dados futuros, permitindo assim uma evolução da área de *machine learning* e *metadada extraction*. Como o CiteULike cria uma rede de artigos, ao mesmo tempo ele possui um banco de dados muito rico, que pode se utilizado como um *dataset* posteriormente, visto suas informações semi-estruturadas já existentes. Inclusive os autores citam o fato de já existirem diversos projetos independentes atualmente que utilizam o *dataset* do CiteULike para realizar suas análises (EMAMY; CAMERON, 2007). Atualmente, no momento da realização deste trabalho, na base de dados do CiteULike podemos encontrar mais de 7.900.000 (sete milhões e novecentos mil) artigos.

Como o banco de dados do CiteULike é criado de forma estruturada a disseminação de conhecimento entre os pesquisadores é facilmente realizada. De posse do link para um determinado artigo a ferramenta consegue todos os detalhes deste documento, como título, autores, etc. Assim permite-se uma pesquisa muito rica em um conjunto de metadados previamente extraídos, criando então uma rede de distribuição de artigos, contribuindo para a comunicação entre pesquisadores de modo geral.

O projeto cumpriu basicamente duas regras:

1. Ele criou um modelo para coleta de informações que é de fácil entendimento para o usuário final, extraindo metadados relevantes que permitem criar automaticamente uma rede de conteúdo, sem deixar de apontar para o artigo em seu site original;

2. Ele mudou a maneira tradicional de descobrir e compartilhar informação, visto que a ferramenta permite que usuários acessem suas listas pessoais entre si, disseminando conhecimento com base em um conhecimento já pré-definido por um pesquisador.

Outro detalhe que permite um uso diferenciado do projeto é a formação de grupos de pesquisa. Com base em interesses similares no histórico de pesquisa, o CiteULike

permite que pesquisadores sejam ligados, compartilhando temas semelhantes e informações de maneira precisa e direta.

Tecnicamente o CiteULike foi criado utilizando como banco de dados o PostgreSQL[4], Tcl[5] como linguagem de programação e ainda utilizando Memcached[6] para criação de cache dos dados armazenados, o que aumentou (e muito) a performance da ferramenta. Seus servidores são baseados em Linux e backups são feitos de 15 em 15 minutos (EMAMY; CAMERON, 2007).

Com base nesta quantidade de detalhes e vantagens o CiteULike se tornou um ponto de referência para catalogação e organização de artigos científicos, permitindo a contribuição entre pesquisadores e formando então uma rede de contribuição que favorece a disseminação da pesquisa ao redor do mundo.

2.3.5 CiteSeer

Um dos projetos mais específicos encontrados é o CiteSeer (GILES; BOLLACKER; LAWRENCE, 1998) e pode ser acessado em <http://citeseerx.ist.psu.edu/>. Seu objetivo não é apenas extrair dados em artigos, mas sim também analisar citações de outros documentos no conteúdo textual encontrado. Assim, ele é capaz de identificar quais documentos são citados, quantas vezes são citados e onde são citados, se assemelhando muito ao processo de citação natural. O CiteSeer foi criado pelos pesquisadores Lee Giles, Kurt Bollacker e Steve Lawrence em 1997 e utiliza modelos SVM para extração de metadados com bastante eficácia (GRANITZER et al., 2012a).

Desta forma, com estas informações identificadas, ele consegue criar um *ranking* dos documentos citados, incluindo seus autores e *journals*, criando a partir de um documento fonte todo um conjunto de relações com outros documentos de uma maneira estatística bem eficaz.

Os índices de citações (*citation indexes*), que são utilizados no projeto, foram originalmente criados para a recuperação de informação, porém sua utilidade é tamanha que permite que citações sejam indexadas de maneira eficaz, permitindo inclusive que referências de documentos em idiomas diferentes sejam identificadas. Desta maneira, essa técnica pode ser utilizadas de diversas formas, não somente na identificação de citações, mas envolvendo também um conjunto de informações, como inclusive identificar a reputação de um determinado artigo científico no meio em que se encontra, simplesmente analisando onde é referência em relação ao número de vezes em que é citado.

[4] Sistema de Banco de Dados relacional com tradição e excelente performance. Sua página inicial é <http://www.postgresql.org>.

[5] Linguagem de programação de scripts muito utilizada em ambientes Linux. Pode ser acessada em <http://www.tcl.tk>.

[6] Sistema de cacheamento baseado em memória, que permite um ganho excelente de performance em ambientes Web, sendo uma das mais utilizadas nos dias atuais. Detalhes em <http://memcached.org>.

Uma proposta interessante em que se baseia o CiteSeer é a de Cameron (CAMERON, 1997). Seu objetivo é de formar uma Base de Citações Universal, onde todos os artigos estariam ligados entre si, independente de qualquer fator externo, como idioma, por exemplo. A diferença entre esta proposta e o trabalho realizado com o CiteSeer é que este último permite que os documentos sejam analisados sem nenhum esforço extra, ou seja, sem a intervenção dos autores dos documentos, como é proposto por Cameron. Neste caso, os documentos seriam analisados de maneira automática, diminuindo o tempo necessário para o relacionamento de todos eles e aumentando a eficácia e eficiência do processo.

O funcionamento do CiteSeer é relativamente simples, porém o trabalho realizado por detrás do processo envolve muito estudo e dedicação. Basicamente ele é capaz de fazer o *download* de *papers* utilizando a Internet (como um coletor de artigos), convertê-los em texto e realizar a análise de todas as suas citações e metadados. Este resultado é armazenado em um banco de dados para consultas e relacionamentos futuros. Um dos pontos interessantes desta análise é que, como ela é feita com base em referência textual, identificado origem e destino, ela pode ser facilmente aplicada tanto no sentido natural de leitura (um artigo cita outros) quanto no sentido inverso (um artigo é citado por outros).

O projeto também possui suas desvantagens, como por exemplo, a não cobertura de alguns *journals* importantes, indexando todos seus documentos online e de maneira automática. Além disso o projeto original não é capaz de identificar mais de um autor nos documentos, sendo a identificação feita apenas pelo campo autor, tendo ele um ou mais pesquisadores envolvidos.

Basicamente o projeto analisa o documento em partes:

- **URL:** a URL onde o documento foi obtido;
- **Cabeçalho:** o bloco de título e autor do documento;
- **Resumo:** o bloco de resumo do documento;
- **Introdução:** a introdução do texto do documento;
- **Citações:** a lista de referências a outros artigos citados no decorrer do texto;
- **Contexto de citação:** o contexto no qual um documento cita outro;
- **Texto completo:** o texto completo do artigo e suas respectivas citações como um todo.

Um dos detalhes importantes do projeto é a identificação das *tags* de citações, que são basicamente as representações visuais quando um outro documento é referenciado, como por exemplo: [4], [Giles997] e “Marr 1982”. Estes pequenos pedaços de textos permitem ao

CiteSeer identificar a relação (e em que momento do texto) entre documentos, permitindo assim que suas análises sejam realizadas de maneira objetiva.

Uma das dificuldades relatadas durante o desenvolvimento do projeto é a identificação de artigos iguais com formas de escrita e informações divergentes. Alguns artigos podem vir com autores com sobrenomes utilizando abreviação, ou até mesmo em ordem diferente. A fim de aumentar esta identificação alguns passos a mais são realizados, como a conversão para caixa baixa de todas as letras, remoção de hifens e das próprias *tags* de citação, expansão de abreviaturas e remoção de algumas palavras externas como "*volume*", "*pages*" e "*number*", por exemplo.

O CiteSeer é um projeto bem maduro e abrangente. Após cerca de 18 (dezoito) anos desde sua criação ele ainda se encontra ativo na Internet, abrangendo cada vez mais documentos e aprimorando cada vez mais suas técnicas de identificação e extração de documentos.

2.3.6 ParsCit

Assim como o CiteSeer o ParsCit é um projeto baseado na identificação de citação de documentos. Porém eles utiliza um modelo de CRF *(Conditional Random Fields)* para identificar sequências nas referências bibliográficas, utilizando uma implementação desta técnica chamada CRF++ (COUNCILL; GILES; KAN, 2008).

O projeto encontra-se ainda ativo e possui atuais contribuições de desenvolvedores ao longo do mundo. Desenvolvido na linguagem de programação *Perl*, o projeto pode ser executado tanto na forma de um *web service*[7] quanto de maneira *standalone*, com execução do código direto quando necessário, dentro de um servidor.

Um detalhe bastante interessante do projeto é a descrição de seu processamento antes e depois da análise do artigo, que os autores chamam de *Pre-Processing Steps* e *Post-Processing Steps*.

Inicialmente o processamento inicia buscando certos *tokens* que podem estar ligados a alguma referência, seja em formato numérico, seja na citação dos nomes dos autores nos mais diversos formatos. Com essa referência coletada o próximo passo é buscar dentro do artigo onde ela está localizada, com base em um conjunto de heurísticas previamente definidas. Para isso é necessária a conversão total do conteúdo do artigo para o formato texto, que deve estar codificado em UTF-8[8] (COUNCILL; GILES; KAN, 2008).

Desta forma a fase de pré-processamento se inicia com esta análise puramente textual, em busca de padrões comuns que podem ter sido utilizados para representações

[7] É a disponibilização de algum serviço na Internet permitindo que outros projetos consultem suas informações e as obtém de maneira simplicada.

[8] Formato de exibição de caracteres que englobam os formatos mais utilizados no mundo, com acentuação e caracteres especiais.

Tabela 6 – Resultados comparativos entre ParsCit e Peng (PENG; MCCALLUM, 2004)

Field	ParsCit			Peng	
	Precision	Recall	F1	Acc.	F1
Author	98.7	99.3	.99	99.9	.99
Booktitle	92.7	94.2	.93	97.7	.94
Date	100	98.4	.99	99.8	.99
Editor	92.0	81.0	.86	99.5	.88
Institution	90.9	87.9	.89	99.7	.94
Journal	90.8	91.2	.91	99.1	.91
Location	95.6	90.0	.93	99.3	.87
Note	74.2	59.0	.65	99.7	.81
Pages	97.7	98.4	.98	99.9	.99
Publisher	95.2	88.7	.92	99.4	.76
Tech	94.0	79.6	.86	99.4	.87
Title	96.0	98.4	.97	98.9	.98
Volume	97.3	95.5	.96	99.9	.98
Average	95.7	95.7	.95	–	.91

de referências à artigos científicos. Com os resultados coletados um modelo CRF é então aplicado aos dados encontrados para processamento futuro.

Com este modelo CRF definido algumas etapas são aplicadas a fim de normalizar os dados encontrados. Nomes de autores podem estar escritos de maneira diferente, com abreviações em nomes de maneira distinta, dependendo do modo como as referências foram escritas no artigo analisado.

Esta análise posterior dos nomes dos autores é feita com base na inspeção de cada palavra individualmente, respeitando então um padrão de normalização definido, sempre com as iniciais dos nomes seguidas do sobrenome escrito normalmente.

A utilização deste projeto é feita de maneira muito simples, podendo ser executado por linha de comando, o que facilitam os testes e a extração de dados. Os resultados obtidos desta análise é então exibido (ou gravado em arquivo) em formato XML, que permite utilização posterior em qualquer tecnologia. Como informado anteriormente os resultados também podem ser coletados em formato de *Web Service*, mas para os fins desta pesquisa apenas sua execução em linha de comando é suficiente.

Visando comparar os resultados obtidos com alguns projetos em que o ParsCit foi baseado, os autores realizam comparações com os resultados obtidos pelo processo descrito por Peng (PENG; MCCALLUM, 2004), onde de maneira geral obteve uma melhora na extração e comparação das referências em torno de 5% (precisão de 0.91 passou para 0.95), demonstrando a eficácia da ferramenta (detalhes na Tabela 6).

Visando também analisar outras ferramentas a mesma comparação de resultados

foi feita com o projeto CiteSeer (subseção 2.3.5), onde o aumento da qualidade dos dados extraídos foi de aproximadamente 19% levando em consideração algumas limitações do CiteSeer que não são compreendidas como são no ParsCit.

2.3.7 CrossRef

CrossRef é uma associação independente mantida por diversas editoras científicas (*publishers*). Além de ser uma organização sem fins lucrativos ela atua como um identificador de um recurso de pesquisa na Internet. Ela permite que referências cruzadas de citações sejam associadas com seus artigos originais, fazendo o link para o recurso no site de sua editora de origem (BERGMARK, 2000).

Suas atividades foram iniciadas em 1999 como uma iniciativa a desenvolver um serviço de links de referências bibliográficas utilizando um identificador único, o *Digital Object Identifier (DOI)*. O lançamento do CrossRef trouxe benefícios para pesquisadores, editores e bibliotecários, visto a permitir uma série de inferências sobre artigos e suas referências cruzadas, permitindo que análises quantificativas fossem realizadas.

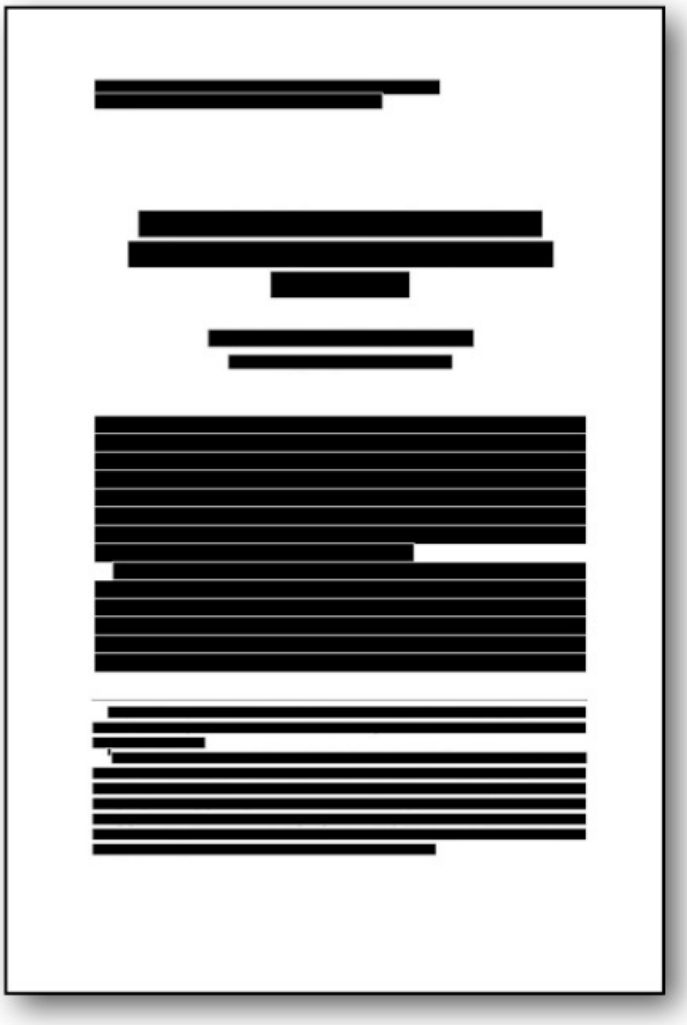
Inicialmente a extração realizada pelo CrossRef levava em consideração apenas as referências citadas em um artigo, mas posteriormente verificou-se que demais partes de um documento eram necessárias, inclusive para que a referência cruzada fosse feita de maneira completa. Assim, um artigo foi dividido em três partes (BERGMARK, 2000):

- *Header Material*: Dados gerais do cabeçalho do artigo, como título, autores, ano de publicação, etc;
- *The Body*: O corpo do documento, onde citações podem ser analisadas de maneira quantitativa, com base na coleta de referências existente do final do documento;
- *The Reference Section*: Parte em que as referências completas são citadas, permitindo que ligações possam ser feitas entre as ligações encontradas no corpo do documento.

O CrossRef não possui uma base de dados de artigos definida com seus próprios documentos, como acontece nos outros projetos e ferramentas. Ele apenas une informações sobre os artigos e faz a referência cruzada destes documentos na Internet, permitindo que citações sejam transformadas em links para os artigos originais, em sua editora de origem.

Por outro lado existe um incentivo por parte do CrossRef para realização do link direto entre documentos PDF e suas referências cruzadas. O projeto CrossRef Labs (<http://labs.crossref.org/>) fornece uma série de ferramentas de código aberto que permitem diversas ações no que tange a área científica, como por exemplo, extração de metadados em artigos científicos.

Figura 7 – Extração de Metadados com base na suposta localização de cada metadado dos artigos.



Fonte: (CROSSREF, 2009)

Esta ferramenta de extração de metadados fornecida pelo CrossRef Labs, chamada `pdfextract`, foi criada a fim de permitir que editoras de menor porte possam integrar sua bases de dados com as referências do CrossRef, aumentando ainda mais o número de documentos que fazem parte do projeto. A ferramenta é desenvolvida na linguagem de programação Ruby, estando em constante atualização, conforme pode ser visto em seu repositório oficial em <https://github.com/CrossRef/pdfextract>.

A `pdfextract` utiliza de informações visuais para identificar cada parte de um artigo científico, pré-selecionando áreas onde cada metadado pode ser encontrado, realizando então um conjunto de identificações por padrões (expressões regulares) que espera-se encontrar (Figura 7).

2.3.8 Outras Ferramentas e Projetos

Além das ferramentas apresentadas diversas outras apresentam características semelhantes, porém com pouca participação de mercado ou com foco em funcionalidades que não são objetos de estudo deste trabalho, embora utilizem de alguma forma extração de metadados, seja com sua própria implementação ou utilizando-se de ferramentas de terceiros. Alguns destes projetos que podem ser citados são:

- **Zotero:** Seu objetivo é servir como um repositório centralizado onde os artigos ficam armazenados, permitindo que as referências sejam exportadas para diversos formatos. Disponível em <https://www.zotero.org>.
- **Citavi:** Permite que você organize artigos científicos de maneira bem estruturada, permitindo inclusive contribuição através de formação de times de pesquisadores. Também permite exportação de suas referências para diversos outros formatos. Disponível em <http://www.citavi.com/>.
- **BibDesk:** Exclusivo para usuários Mac/Apple. É um *software* final que permite que usuários organizem seus artigos, simplificando todo o processo de exportação dos formatos de referências. É focado para usuários LaTeX. Disponível em <http://bibdesk.sourceforge.net>.
- **Docear:** Permite que usuários organizem suas bibliotecas digitais de artigos científicos, permitindo também compartilhar informações com outros usuários. Possui versões para Windows, Linux e Mac. Disponível em <https://www.docear.org/>.
- **Qiqqa:** Permite que além de organizar artigos seus usuários façam gestão das referências destes documentos, que ficam armazenados também em *cloud computing*. Possui versões Windows e Android. Disponível em <http://www.qiqqa.com/>.
- **Papers:** Aplicativo disponível para Windows, Mac, iPhone e iPad que permite que usuários gerenciem e pesquisem artigos científicos. Possui um banco de dados próprio e é oferecido apenas em versão paga. Disponível em <http://www.papersapp.com/>.
- **JabRef:** Muito parecido com a ferramenta *BibDesk*, visto que seu objetivo é organizar artigos e referências focados em usuários LaTeX. Possui versões para Windows, Linux e Mac. Uma de suas vantagens é possuir uma interface muito simplificada, facilitando a utilização pelos usuários finais. Disponível em <http://jabref.sourceforge.net/>.
- **EndNote:** Além de permitir a organização muito bem estruturada de artigos científicos ele é também uma ferramenta de pesquisa, permitindo também exportar seu catálogo de referências para os mais diversos formatos. Possui versões para Windows e Mac, podendo ser acessado também através da Web. Disponível em <http://endnote.com/>.

Tabela 7 – Características de cada ferramenta analisada

Ferramenta	**Linguagens de Programação**	**Técnicas Utilizadas**	**Utilização via *Command Line***
Cermine	Java	SVM, CRF, Word Clustering	Sim
TeamBeam	Java	Maximun Entropy, HMM	Não
Mendeley	Qt	SVM, Word Clustering	Não
CiteULike	Perl, Python, Ruby, Tcl, Java	Expressões Regulares	Não
CiteSeer	Python, Perl, Java	SVM, CRF (ParsCit), Word Clustering	Sim
ParsCit	Perl, Ruby	CRF	Sim
CrossRef	Ruby, Python	Expressões Regulares, Posicionamento Espacial	Sim

Comentário: Falar sobre o ResearchGate.

Após a análise das ferramentas foi criado um quadro consolidando as principais informações sobre cada uma, permitindo que as ferramentas que farão parte dos testes deste trabalho possam ser selecionadas. A Tabela 7 contém dados como linguagem de programação utilizada, técnicas de extração de metadados empregadas e a possibilidade de execução de cada ferramenta utilizando a linha de comando, o que permite com facilidade a execução de testes com cada uma delas.

Comentário: Escrever sobre Posicionamento Espacial

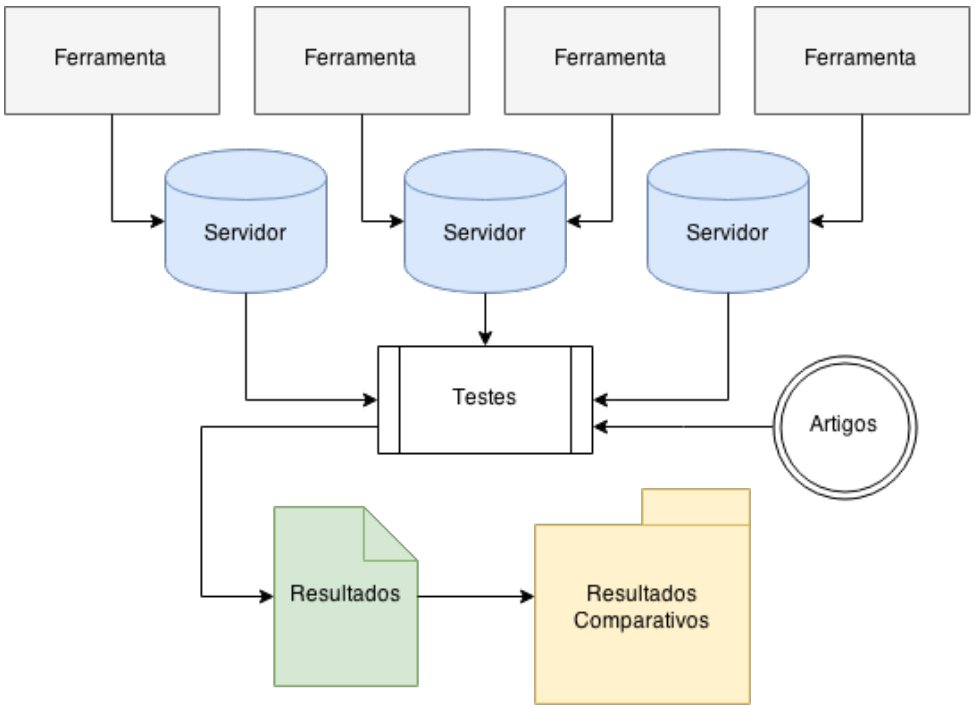
3 Metodologia

Este trabalho tem como metodologia uma pesquisa de caráter não-experimental e quantitativa, por se tratar de extração automática de metadados por ferramentas previamente selecionadas, tendo os resultados comparados com a extração manual do mesmo conjunto de artigos científicos.

Primeiramente, são filtradas as ferramentas encontradas a fim de analisar realmente as que possuem viabilidade técnica de testes dentro do objetivo da pesquisa. Desta forma, diversos elementos serão utilizados a fim de se obter os resultados desejados.

De modo geral o procedimento de testes deste trabalho será realizado através da instalação de cada ferramenta selecionada em máquinas (servidores), permitindo que cada uma tenha seu conjunto necessário de tecnologias para seu correto funcionamento. Assim, os artigos selecionados para testes serão utilizados em cada uma destas ferramentas e seus resultados analisados, centralizados e consolidados a fim de se obter uma nota final para cada análise encontrada. O fluxo de passos necessários para a realização destes testes pode ser melhor visualizado na Figura 8.

Figura 8 – Processo de Metodologia utilizado



Fonte: O próprio autor

Tabela 8 – Áreas do Conhecimento (CNPq)

Áreas do Conhecimento	**Subáreas**
Ciências Agrárias	157
Ciências Biológicas	104
Ciências da Saúde	76
Ciências Exatas e da Terra	243
Ciências Humanas	163
Ciências Sociais Aplicadas	185
Engenharias	305
Lingüística, Letras e Artes	53
Outros	4
Total	**1288**

Fonte: <http://www.memoria.cnpq.br/areasconhecimento/index.htm>

3.1 Escolha do Corpus

Visando provar a eficiência das ferramentas - juntamente com a implementação das técnicas por elas utilizadas -, desejamos ter resultados exatos da extração de metadados, de maneira que possam ser comparados e verificados com os metadados manualmente extraídos. Deste modo, foi selecionada uma série de artigos científicos das mais diversas áreas de pesquisa, de diversos eventos distintos, com padrões visuais totalmente diferentes e que são passível de ser analisados e seus metadados coletados.

Em virtude da necessidade de realizar testes com as ferramentas selecionadas em um ambiente real e representativo, foi realizada uma pesquisa no site do CNPq <http://www.cnpq.br/> a fim de se obter a relação das áreas e subáreas do conhecimento reconhecidas oficialmente. Deste modo foi constatada a existência de 9 (nove) áreas do conhecimento, totalizando 1.288 (mil duzentas e oitenta e oito) subáreas, conforme pode ser verificado na Tabela 8.

Com base no grande número de subdivisões de cada área do conhecimento (vide Tabela 8), a seleção das subáreas será limitada a 2 (duas) e a escolha feita com base na existência de curso de graduação e/ou departamento na Universidade Federal de Minas Gerais (UFMG), o que facilitaria o contato com professores e coordenadores dos respectivos cursos. Desta forma, excluindo-se a área “Outros”, seriam analisadas 16 (dezesseis) subáreas, sendo 2 (duas) para cada uma das 8 (oito) áreas do conhecimento.

Com as subáreas selecionadas, foi realizada uma entrevista com professores e/ou coordenadores de cada curso ou departamento correspondente na UFMG, obtendo então as bases de dados e/ou revistas mais utilizadas e relevantes para cada subárea do conhecimento, de maneira a construir um *corpus* realmente significativo. A relação dos professores entrevistados e as subáreas do conhecimento selecionadas pode ser vista na Tabela 9.

Tabela 9 – Professores entrevistados para cada subárea do conhecimento.

Subárea do Conhecimento	**Professor(a)**
Arquitetura e Urbanismo	Eleonora Sad de Assis
Ciência da Computação	Adriano Alonso Veloso Alberto Henrique Frade Laender Arnaldo de Albuquerque Araújo
Ciência da Informação	Beatriz Valadares Cendón
Ciências Biológicas (Genética)	Mônica Bucciarelli Rodriguez
Ciências Biológicas (Zoologia)	Alfredo Hannemann Wieloch
Enfermagem	Eline Lima Borges
Engenharia Civil	
Engenharia Mecânica	Antônio Eustáquio de Melo Pertence
Fonoaudiologia	Sirley Alves da Silva Carvalho
Geologia	
História	
Letras	Adriana Silvina Pagano
Medicina Veterinária	João Paulo Amaral Haddad Jenner Karlisson Pimenta dos Reis
Música	João Pedro Paiva de Oliveira
Psicologia	Carmen Elvira Flores-Mendoza Prado Livia de Oliveira Borges
Zootecnia	Ângela Maria Quintão Lana

Para cada uma das subáreas do conhecimento selecionadas foram coletados 7 (sete) artigos científicos, levando em consideração a quantidade de bases de dados informadas, divididas da maneira mais igualitária possível. Por exemplo, para uma subárea que possui apenas 1 (uma) base de dados informada foram coletados 7 (sete) artigos desta base. Para uma subárea com 2 (duas) bases de dados foram coletados 4 (quatro) artigos para a primeira base e 3 (três) para a segunda, e assim por diante. No total foram selecionados 112 (cento e doze) artigos, contemplando 16 subáreas do conhecimento, formando então o *corpus* da pesquisa.

A seleção dos artigos em cada base de dados foi feita de maneira arbitrária, levando em consideração diferenças de layouts e posicionamento dos elementos, permitindo que uma maior variedade de documentos seja analisada.

Todos os artigos selecionados foram escritos na língua inglesa. Esta decisão foi tomada em virtude de, além de ser a língua inglesa a universal para disseminação de conhecimento, ela é a mais utilizada no meio acadêmico, possuindo um universo muito maior e mais rico de artigos escrito no idioma. Além disso, algumas das ferramentas e respectivas técnicas utilizadas nos testes utilizam de "processamento de linguagem natural" para extração dos metadados, tendo por padrão a utilização do inglês na análise dos textos dos documentos.

Em virtude destas colocações a abrangência de outros idiomas entraria em um aspecto que não é objetivo deste trabalho abordar, visto a diversificação de culturas e símbolos, fazendo com que línguas orientais, como o mandarim ou japonês por exemplo, tenham análises diferenciadas em função de suas diferenças nas formas de representação e leitura, necessitando de outras técnicas e/ou ferramentas mais direcionadas a fim de se obter os resultados esperados.

No que tange a escolha das ferramentas para testes foi utilizado apenas um ponto na seleção: a sua utilização por linha de comando. Embora algumas ferramentas possuem código aberto a extração de metadados faz parte de um contexto específico da aplicação, dificultando a utilização de apenas este recurso. Assim, foram selecionadas para teste apenas as ferramentas que permitem o uso de sua funcionalidade de extração de metadados de maneira individual, independente da linguagem de programação ou tecnologia apresentada.

Deste modo, dentre as ferramentas apresentadas neste trabalho, presentes na Tabela 7, as ferramentas selecionadas para teste foram: Cermine, CiteSeer, CrossRef e ParsCit.

3.2 Desenho do Experimento

Tendo selecionadas as ferramentas e os artigos que serão utilizados para os testes parte-se para a instalação adequada de cada ferramenta nos servidores, juntamente com as tecnologias necessárias e as linguagens de programação utilizadas pelos seus desenvolvedores.

Cada ferramenta será testada em separado, observando suas características particulares. Assim, cada artigo selecionado será testado para aquela ferramenta, anotando os resultados obtidos na extração. Estes resultados serão separados por metadados, o que permitirá calcular qual a porcentagem de acerto que a ferramenta teve na extração de cada metadado analisado.

Assim, o processo será repetido para cada ferramenta e o resultado registrado, permitindo calcular sua porcentagem total de acertos de maneira simplificada. Para isso será criado um "Quadro Comparativo", no qual serão inseridos os resultados dos testes de cada ferramenta.

3.2.1 Metadados, Pesos e Resultados

Em se tratando de pesquisa por artigos científicos, pequenos detalhes podem fazer a diferença. Desta forma, uma extração de metadados não muito eficaz pode prejudicar direta ou indiretamente os resultados apresentados por esta busca. Por outro lado, alguns metadados tendem a ser mais utilizados na pesquisa que outros, o que implica em uma

Tabela 10 – Os metadados e seus pesos atribuídos

Metadado	**Relevância**	**Peso**
Título	Um dos termos mais buscados quando se pesquisa um artigo	5
Autor(es)	Outro termo muito utilizado na busca por artigos	4
E-mail(s)	Pouco relevante no quesito pesquisa de artigos	1
Resumo	Importante por conter palavras chaves e o resumo propriamente dito	3
Referências	Muito importante e necessário, pois será utilizada na referência inversa de autores	4

responsabilidade maior na eficiência de sua extração.

Geralmente quando vamos buscar artigos, procuramos primeiro pelo título (quando procuramos por um documento específico) ou então pelo nome do autor (quanto procuramos por artigos de um determinado pesquisador). Deste modo serão atribuídos pesos para cada um dos metadados, de maneira a valorizar a extração destas informações que podem influenciar diretamente os resultados de busca.

Deste modo apresentamos a Tabela 10, que demonstra como cada metadado deve ser interpretado e qual o peso que lhe será atribuído, sendo utilizado o inteiro 1 (um) para o peso mais baixo e o 5 (cinco) para o peso mais alto, sendo consequentemente o(s) metadado(s) mais importante(s). Os pesos utilizados, assim como a ordem de importância escolhida foram fundamentados de maneira arbitrária, baseados na experiência do autor.

Como a extração de um metadado nem sempre ocorre de maneira 100% eficaz, visando uma avaliação mais detalhada de cada ferramenta, será calculada a precisão do resultado da extração de cada metadado, feita com base na porcentagem de sucesso obtida para aquele conjunto de caracteres. Este cálculo será feito por algoritmo desenvolvido pelo próprio autor, analisando qual a taxa de acerto para cada metadado, referenciado posteriormente por $P_{título}$ (precisão de acerto para o metadado título).

Comentário: Falar a fórmula ou escrever o algorítimo citado no parágrafo acima.

Como cada ferramenta será testada em separado, os resultados da extração de cada artigo serão registrados, tendo o total da precisão calculado de acordo com a média aritmética dos resultados obtidos para aquele metadado. Por exemplo, para a Ferramenta "A" serão analisados 100 (cem) artigos. A precisão na extração do título de cada artigo ($P_{título1}$, $P_{título2}$, ..., $P_{tituloN}$), por exemplo, será somada e o resultado dividido pelo número de artigos - no caso 100. Assim tem-se a precisão geral para o metadado "Título" da

Ferramenta "A" ($P_{título}$):

$$P_{título} = (P_{título1} + P_{título2} + P_{título3}... + P_{título100})/100$$

De posse dos resultados para cada metadado extraído podemos comparar as ferramentas a fim de obter qual é mais adequada para cada tipo de metadado. Podemos inferir, portanto, que a ferramenta "X" apresenta melhores resultados do que "Y" na extração do nome dos autores, por exemplo.

3.2.2 Índice de Confiabilidade

Considerando que cada metadado possui um peso diferente necessitamos calcular o índice de acertos a ser utilizado em cada resultado coletado para cada ferramenta testada. Assim chegamos a uma fórmula matemática à qual chamaremos "Índice de Confiabilidade", que calcula o resultado obtido através dos pesos que foram atribuídos a cada metadado, para cada ferramenta. Este índice é a nota final de cada ferramenta, levando em consideração todos os resultados obtidos por ela.

Este índice utiliza os pesos anteriormente definidos e a precisão dos resultados obtida, de maneira a permitir chegar em uma única nota final para cada ferramenta testada.

Basicamente a fórmula é a média ponderada dos resultados alcançados na extração de cada metadado dos artigos, seguindo os pesos apresentados na Tabela 10. Cada peso é atribuído ao resultado encontrado em cada ferramenta.

A título de exemplo, após o teste de uma ferramenta, supondo que ela conseguiu extrair 87% dos títulos de todos artigos com sucesso, sua precisão com relação ao título será 87 ($P_{título} = 87$), que será multiplicado pelo peso correspondente, neste caso, o inteiro 5. Isso ocorre para todos os metadados extraídos, seguindo seus respectivos pesos. A descrição de cada variável no Índice de Confiabilidade poderá ser obtida de acordo com a Tabela 11.

$$IC_{FerramentaX} = (5 * P_{título} + 4 * P_{autor} + 1 * P_{email} + 3 * P_{resumo} + 4 * P_{referências})/17$$

Tabela 11 – Descrição de cada variável no Índice de Confiabilidade

Variável	**Descrição**
$P_{título}$	Precisão na obtenção do título
P_{autor}	Precisão na obtenção do(s) autor(es)
P_{email}	Precisão na obtenção dos e-mails dos autores
P_{resumo}	Precisão na obtenção do resumo
$P_{referências}$	Precisão na obtenção das referências

Assim, de posse do Índice de Confiabilidade de cada ferramenta podemos classificar cada uma com base nos seus resultados apresentados. Esta classificação será feita de maneira arbitrária, sem objetivo algum de denegrir alguma ferramenta em favorecimento de outra, mas sim classificar cada uma delas com base nos resultados obtidos e critérios adotados neste trabalho. Desta forma, podemos classificar cada ferramenta seguindo os tópicos abaixo:

1. **Precisa (P):** Quando o Índice de Confiabilidade é maior ou igual a 90 ($IC \geq 90$).
2. **Satisfatória (S):** Quando o Índice de Confiabilidade é maior ou igual a 80 e menor que 90 ($80 \leq IC < 90$).
3. **Insatisfatória (I):** Quando o Índice de Confiabilidade é menor que 80 ($IC < 80$).

3.3 Ambiente Tecnológico

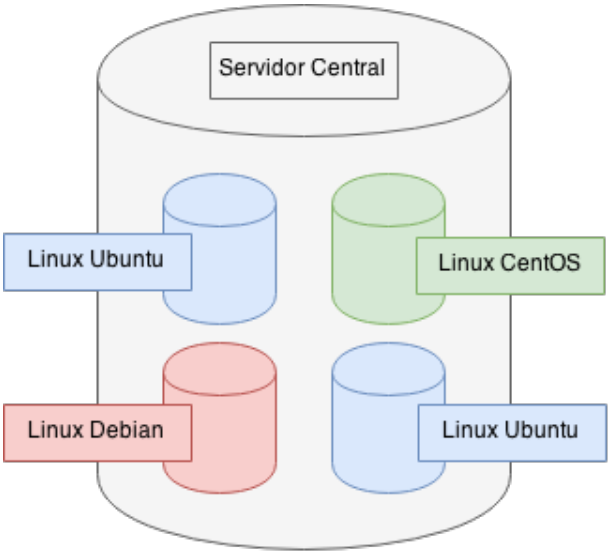
Algumas ferramentas exigem um conjunto de tecnologias muito diferente das demais, como é o caso do CiteSeer (subseção 2.3.5). Sendo assim, neste caso em específico, será utilizado um servidor apenas para os testes desta ferramenta, juntamente com as tecnologias que ela exige para um correto funcionamento.

Outras ferramentas, por serem escritas em linguagens de programação iguais ou por terem seu conjunto tecnológico muito semelhante, serão instaladas em um único servidor, compartilhando recursos e simplificando o trabalho de configuração, visto que possuem necessidades parecidas.

Estes servidores serão criados através de máquinas virtuais, o que traz benefícios não somente de performance mas de flexibilidade quanto das tecnologias necessárias para o funcionamento de cada técnica, permitindo que os testes possam ser feitos em sistemas operacionais distintos mas utilizando dos mesmos recursos computacionais.

Deste modo, dentro de um mesmo servidor podemos criar diversos outros servidores “virtuais”, com características diferentes e linguagens de programação diferentes,

Figura 9 – Utilização de Máquinas Virtuais como Ambiente de Testes



Fonte: O próprio autor

compartilhando dos mesmos recursos do servidor original, como memória RAM, espaço em disco, dentre outros recursos, como pode ser visto na Figura 9.

Referências

BERGER, A. L.; PIETRA, S. A. D.; PIETRA, V. J. D. A maximum entropy approach to natural language processing. 1996. Citado 2 vezes nas páginas 29 e 41.

BERGMARK, D. *Automatic Extraction of Reference Linking Information from Online Documents*. Ithaca, NY, USA, 2000. Citado na página 50.

CAMERON, R. D. A universal citation database as a catalyst for reform in scholarly communication. v. 2, n. 4, 1997. Citado na página 47.

CATHRO, W. Metadata: an overview. 1997. Disponível em: <http://www.nla.gov.au/openpublish/index.php/nlasp/article/view/1019/1289>. Citado na página 15.

CHIEU, H. L.; NG, H. T. *A Maximum Entropy Approach to Information Extraction from Semi-Structured and Free Text*. 2002. Citado na página 22.

COUNCILL, I. G.; GILES, C. L.; KAN, M. yen. Parscit: An open-source crf reference string parsing package. In: *International Language Resources and Evaluation*. [S.l.]: European Language Resources Association, 2008. Citado na página 48.

CROSSREF. *A short history of CrossRef*. [S.l.], 2009. Disponível em: <http://www.crossref.org/08downloads/CrossRef10Years.pdf>. Citado na página 51.

EMAMY, K.; CAMERON, R. Citeulike: A researcher's social bookmarking service. 2007. Disponível em: <http://www.ariadne.ac.uk/issue51/emamy-cameron/>. Citado 3 vezes nas páginas 44, 45 e 46.

GALLAGHER, F. *CiteULike: Everyone's library*. 2008. Disponível em: <http://www.citeulike.org/faq/faq.adp>. Citado na página 44.

GILES, C. L.; BOLLACKER, K. D.; LAWRENCE, S. Citeseer: An automatic citation indexing system. p. 89–98, 1998. Citado na página 46.

GRANITZER, M. et al. A comparison of layout based bibliographic metadata extraction techniques. In: . [S.l.]: ACM, 2012. Citado 4 vezes nas páginas 37, 38, 42 e 46.

GRANITZER, M. et al. A comparison of metadata extraction techniques for crowdsourced bibliographic metadata management. *SAC 2012*, Mar 2012. Citado na página 37.

GROUP, T. O. *The Open Group Base Specifications Issue 7*. 2013. Citado na página 18.

HAN, H. et al. Automatic document metadata extraction using support vector machines. 2003. Citado 5 vezes nas páginas 22, 24, 31, 39 e 42.

HAN, H. et al. Rule-based word clustering for document metadata extraction. p. 1049–1053, 2005. Citado 3 vezes nas páginas 29, 30 e 31.

HENNING, V.; REICHELT, J. Mendeley – a last.fm for research? *eScience, 2008. eScience '08. IEEE Fourth International Conference on*, IEEE, p. 327–328, Dec 2008. Citado na página 42.

KERN, R. et al. Teambeam — meta-data extraction from scientific literature. v. 18, n. 7/8, 2012. Disponível em: <http://www.dlib.org/dlib/july12/kern/07kern.html>. Citado na página 40.

KLEENE, S. C. Representation of events in nerve nets and finite automata. *Princeton University Press*, p. 3–42, 1956. Citado na página 17.

KUNZE, J.; BAKER, T. *The Dublin Core Metadata Element Set*. [S.l.], 2007. Disponível em: <http://www.ietf.org/rfc/rfc5013.txt>. Citado na página 16.

LAFFERTY, J.; MCCALLUM, A.; PEREIRA, F. C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ACM (Ed.). *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*. [S.l.]: ACM, 2001. p. 282–289. Citado 3 vezes nas páginas 29, 32 e 33.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of Machine Learning*. [S.l.: s.n.], 2012. ISBN 978-0-262-01825-8. Citado na página 12.

PENG, F.; MCCALLUM, A. Accurate information extraction from research papers using conditional random fields. In: *HLT-NAACL04*. [S.l.: s.n.], 2004. p. 329–336. Citado 6 vezes nas páginas 7, 33, 35, 36, 41 e 49.

RABINER, L. R.; JUANG, B. H. An introduction to hidden markov models. 1986. Citado na página 25.

SEYMORE, K.; ROSENFIELD, R. Learning hidden markov model structure for information extraction. p. 37–42, 1999. Citado 5 vezes nas páginas 22, 26, 27, 28 e 34.

THOMPSON, K. Programming techniques: Regular expression search algorithm. *Communications of the ACM 11*, p. 419–422, 1968. Citado na página 17.

TKACZYK, D. et al. Cermine — automatic extraction of metadata and references from scientific literature. 2014. Citado 2 vezes nas páginas 39 e 40.

VAPNIK, V.; CORTES, C. Support-vector networks. In: PUBLISHERS, B. K. A. (Ed.). *Machine Learning*. [S.l.]: Kluwer Academic Publishers, Boston, 1995. v. 20, p. 273–297. Citado 3 vezes nas páginas 21, 22 e 23.

WEIBEL, S. The dublin core: A simple content descrip- tion format for electronic resources. In: SCIENCE, B. of the American Society for I. (Ed.). [S.l.: s.n.], 1997. p. 9–11. Citado na página 15.

ZHANG, N. R. Hidden markov models for information extraction. 2001. Citado 3 vezes nas páginas 6, 28 e 29.

Apêndices

APÊNDICE A – Listagem de artigos analisados para realização do experimento (em atualização)

#	**Título *(Autores)***	**Área/Disciplina**
1	Benefits of extra-pair mating may depend on environmental conditions—an experimental study in the blue tit (Cyanistes caeruleus) *(Aneta Arct, Szymon M. Drobniak, Edyta Podmokta, Lars Gustafson, Mariusz Cichoń)*	Ciências Biológicas
2	Cost-Effective Suppression and Eradication of Invasive Predators *(Peter W. J. Baxter, John L. Sabo, Chris Wilcox, Michael A. McCarthy, Hugh P. Possingham)*	Ciências Biológicas
3	Model selection and model averaging in behavioural ecology: the utility of the IT-AIC framework *(Shane A. Richards, Mark J. Whittingham, Philip A. Stephens)*	Ciências Biológicas
4	Territory size of the flavescent warbler, Basileuterus flaveolus (Passeriformes, Emberizidae), in a forest fragment in Southeastern Brazil *(Charles Duca, Miguel Â. Marini)*	Ciências Biológicas
5	Effectiveness of Corridors Relative to Enlargement of Habitat Patches *(Mattew R. Falcy, Cristián F. Estades)*	Ciências Biológicas
6	High fidelity on islands: a comparative study of extrapair paternity in passerine birds *(Simon C. Griffith)*	Ciências Biológicas
7	Parentage and Relatedness in Polyandrous Comb-Crested Jacanas Using ISSRs *(S. M. Haig, T. R. Mace, T. D. Mullins)*	Ciências Biológicas
8	Factors influencing double brooding in Eurasian Hoopoes Upupa epops *(Jael Hoffmann, Erik Postma, Michael Schaub)*	Ciências Biológicas
9	The importance of fission–fusion social group dynamics in birds *(Matthew J. Silk, Darren P. Croft, Tom Tregenza, Stuart Bearhop)*	Ciências Biológicas
10	Genetic monogamy in two long-lived New Zealand passerines *(Sabrina S. Taylor, Sanne Boessenkool, Ian G. Jamieson)*	Ciências Biológicas

11	A Probabilistic Framework for Semi-Supervised Clustering *(Sugato Basu, Mikhail Bilenko, Raymond J. Mooney)*	Ciência da Computação
12	Document Clustering: The Next Frontier *(David C. Anastasiu, Andrea Tagarelli, George Karypis)*	Ciência da Computação
13	Enhancing Semi-Supervised Document Clustering with Feature Supervision *(Yeming Hu, Evangelos E. Milios, James Blustein)*	Ciência da Computação
14	Instance-Level Constraint-Based Semisupervised Learning With Imposed Space-Partitioning *(Jayaram Raghuram, David J. Miller, George Kesidis)*	Ciência da Computação
15	Alternatives to the k-means algorithm that find better clusterings *(Greg Hamerly, Charles Elkan)*	Ciência da Computação
16	Towards Parameter-Free Data Mining *(Eamonn Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana)*	Ciência da Computação
17	Learning Nonparametric Kernel Matrices from Pairwise Constraints *(Steven C. H. Hoi, Rong Jin, Michael R. Lyu)*	Ciência da Computação
18	Enhancing Semi-Supervised Clustering: A Feature Projection Perspective *(Wei Tang, Hui Xiong, Shi Zhong, Jie Wu)*	Ciência da Computação
19	Integrating Constraints and Metric Learning in Semi-Supervised Clustering *(Mikhail Bilenko, Sugato Basu, Raymond J. Mooney)*	Ciência da Computação
20	On Weighting Clustering *(Richard Nock, Frank Nielsen)*	Ciência da Computação
21	Masculinity, Sexuality and the Body of Male Soldiers *(Nyameka Mankayi)*	Psicologia
22	Consumption of alcohol and depression in students of a public school of CoatzaCoalCos, VeraCruz, Mexico *(Brenda Alicia Hernández-Cortaza, Leticia Cortaza-Ramirez, Moacyr Lobo da Costa Junior)*	Psicologia
23	Psychometric Properties of the Spanish Language Version of the Stress in Children Questionnaire (SiC) *(Alejandra Caqueo-Urízar, Alfonso Urzúa, Walter Osika)*	Psicologia
24	Facial information processing in schizophrenia *(João Paulo Machado de Sousa, Jaime Eduardo Cecílio Hallak)*	Psicologia
25	Intelligence and socioeconomic success: A meta-analytic review of longitudinal research *(Tarmo Strenze)*	Psicologia

26	The Science of Sex Differences in Science and Mathematics *(Diane F. Halpern, Camilla P. Benbow, David C. Geary, Ruben C. Gur, Janet Shibley Hyde, Morton Ann Gernsbacher)*	Psicologia
27	Hardiness and Burnout Syndrome: A Cross-Cultural Study among Portuguese and Brazilian Nurses *(Mary Sandra Carlotto, Cristina Queirós, Sofia Dias, Mariana Kaiseler)*	Psicologia
28	Social Psychology: Fundamentals and Fundamentalisms *(Marcus Eugênio Oliveira Lima)*	Psicologia
29	Computerized Assessment of Food Preferences in Adolescents in the Stimulus Equivalence Paradigm *(Gisele Straatmann, Sebastião Sousa Almeida, Julio C. de Rose)*	Psicologia

Anexos

ANEXO A – Elementos do padrão Dublin Core, versão 1.1.

Name	**Label**	**Definition**	**Comment**
title	Title	A name given to the resource.	
creator	Creator	An entity primarily responsible for making the resource.	Examples of a Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.
subject	Subject	The topic of the resource.	Typically, the subject will be represented using keywords, key phrases, or classification codes. Recommended best practice is to use a controlled vocabulary. To describe the spatial or temporal topic of the resource, use the Coverage element.
description	Description	An account of the resource.	Description may include but is not limited to: an abstract, a table of contents, a graphical representation, or a free-text account of the resource.
publisher	Publisher	An entity responsible for making the resource available.	Examples of a Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity.
contributor	Contributor	An entity responsible for making contributions to the resource.	Examples of a Contributor include a person, an organization, or a service. Typically, the name of a Contributor should be used to indicate the entity.

date	Date	A point or period of time associated with an event in the lifecycle of the resource.	Date may be used to express temporal information at any level of granularity. Recommended best practice is to use an encoding scheme, such as the W3CDTF profile of ISO 8601 [W3CDTF].
type	Type	The nature or genre of the resource.	Recommended best practice is to use a controlled vocabulary such as the DCMI Type Vocabulary [DCTYPE]. To describe the file format, physical medium, or dimensions of the resource, use the Format element.
format	Format	The file format, physical medium, or dimensions of the resource.	Examples of dimensions include size and duration. Recommended best practice is to use a controlled vocabulary such as the list of Internet Media Types [MIME].
identifier	Identifier	An unambiguous reference to the resource within a given context.	Recommended best practice is to identify the resource by means of a string conforming to a formal identification system.
source	Source	A related resource from which the described resource is derived.	The described resource may be derived from the related resource in whole or in part. Recommended best practice is to identify the related resource by means of a string conforming to a formal identification system.
language	Language	A language of the resource.	Recommended best practice is to use a controlled vocabulary such as RFC 4646 [RFC4646].
relation	Relation	A related resource.	Recommended best practice is to identify the related resource by means of a string conforming to a formal identification system.

coverage	Coverage	The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant.	Spatial topic and spatial applicability may be a named place or a location specified by its geographic coordinates. Temporal topic may be a named period, date, or date range. A jurisdiction may be a named administrative entity or a geographic place to which the resource applies. Recommended best practice is to use a controlled vocabulary such as the Thesaurus of Geographic Names [TGN]. Where appropriate, named places or time periods can be used in preference to numeric identifiers such as sets of coordinates or date ranges.
rights	Rights	Information about rights held in and over the resource.	Typically, rights information includes a statement about various property rights associated with the resource, including intellectual property rights.