

José Alberto Grossi Júnior

Análise Comparativa de Ferramentas de Extração Automatizada de Metadados em Artigos Científicos

Belo Horizonte/MG, Brasil

Fev 2016, v-0.9.0, [f029ce2]

José Alberto Grossi Júnior

Análise Comparativa de Ferramentas de Extração Automatizada 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

Fev 2016, v-0.9.0, [f029ce2]

José Alberto Grossi Júnior

Análise Comparativa de Ferramentas de Extração Automatizada de Metadados em Artigos Científicos/ José Alberto Grossi Júnior. – Belo Horizonte/MG, Brasil, Fev 2016, v-0.9.0, [f029ce2]-

84 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, Fev 2016, v-0.9.0, [f029ce2].

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

CDU 02:141:005.7

José Alberto Grossi Júnior

Análise Comparativa de Ferramentas de Extração Automatizada 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, 27 de novembro de 2015:

Marcello Peixoto Bax
Orientador

Renato Rocha Souza
Coorientador

Beatriz Valadares Cendón
Professora Convidada - ECI/UFMG

Max Cirino de Mattos
Professor Convidado - UNA

Renata M. Abrantes Baracho Porto
Professora Suplente - ECI/UFMG

Belo Horizonte/MG, Brasil
Fev 2016, v-0.9.0, [f029ce2]

*Este trabalho é dedicado a todas as pessoas que
desejam superar seus próprios limites.*

Agradecimentos

Agradeço aos meus orientadores Prof. Marcello Peixoto Bax e Prof. Renato Rocha Souza pelo apoio, paciência e dedicação em cada etapa do trabalho.

Agradeço também aos membros da banca pelo aceite do convite e pela paciência na escolha da data da defesa.

Agradecimentos especiais são direcionados aos meus pais por acreditarem sempre em minhas escolhas e principalmente pelo incentivo para que eu me torne sempre uma pessoa melhor a cada dia.

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 número cada vez maior de artigos permanece sem uma extração de metadados eficaz, o que dificulta a disseminação de conhecimento e principalmente a pesquisa eletrônica desses documentos.

Este trabalho compara a capacidade de extração de metadados de algumas ferramentas pré-selecionadas - Cermine, CiteSeer, CrossRef e ParsCit - utilizando para isso um experimento empírico com um conjunto de artigos. Esse conjunto abrange diversas áreas do conhecimento, diversos eventos e formatos visuais diferentes. O experimento foi realizado em ambientes pré-configurados de acordo com a característica técnica de cada ferramenta, permitindo que todos os artigos tivessem seus metadados extraídos por cada uma delas e os resultados comparados individualmente.

Desta forma, com base nos resultados apresentados, pôde-se identificar o comportamento de cada uma das ferramentas no tocante à sua capacidade de extração correta de metadados. Exceto pela ferramenta CrossRef, todas as demais obtiveram resultados acima de 60%, chegando a 86.83% da ferramenta Cermine. Além disso, foram evidenciadas as principais fragilidades dessas ferramentas; pontos onde seriam necessários ajustes; metadados para os quais se obtém um maior sucesso na extração. Ademais, é apresentado também um índice de confiabilidade, que permite estabelecer para cada ferramenta uma nota. Essa nota é calculada 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 its own particularity, technology and used techniques. However, with the increasing scientific production and the numerous publishers, events and conferences, a large part of papers still remain without an effective automated metadata extraction, hindering the knowledge dissemination and mainly the electronic search for these documents.

The present work compares the correct metadata extraction from some preselected tools - Cermine, CiteSeer, CrossRef and ParsCit - using an empirical experiment with a set of scientific papers. This set covers different knowledge areas, conferences and different layouts. The experiment was made inside custom environments according the technologies each tool needs, allowing all papers to have their metadata extracted by each one and comparing results one by one.

Thereby, according the presented results, it's possible to identify the behavior of each tool related to the right metadata extraction. Except for the CrossRef tool, all others obtained results over 60%, including Cermine with to 86.83%. Moreover, the major weaknesses were identified for each tool; points to be fixed; metadata fields with better extraction results. Furthermore it's also presented a reliability index, that allows to establish a grade for each tool. This grade is calculated based on the metadata extraction results using the selected set of papers.

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

Lista de ilustrações

Figura 1 – Exemplos de artigos da área de Ciência da Computação	13
Figura 2 – Exemplos de artigos com padrões visuais diferentes, de diversas áreas do conhecimento	14
Figura 3 – Processo de Extração de Metadados	15
Figura 4 – Distância representando a separação entre classes na técnica de SVM. .	24
Figura 5 – 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.	27
Figura 6 – Estados utilizados por (ZHANG, 2001) em seu modelo HMM.	29
Figura 7 – <i>Workflow</i> da extração de metadados usando <i>cluster</i> de palavras.	31
Figura 8 – Cermine Extraction Workflow	41
Figura 9 – Extração de Metadados com base na suposta localização de cada meta- dado dos artigos.	52
Figura 10 – Esquema visual da arquitetura do experimento	56

Lista de tabelas

Tabela 1 – Formas de representação de repetições em Expressões Regulares. . . .	20
Tabela 2 – Classes do padrão POSIX.	22
Tabela 3 – Relação de classes utilizadas e comparação com o padrão Dublin Core. . . .	24
Tabela 4 – Resultados de extração para CRFs após análise do <i>dataset</i> com cabeçalhos (PENG; MCCALLUM, 2004).	36
Tabela 5 – Resultados de extração para CRFs após análise do <i>dataset</i> com referências (PENG; MCCALLUM, 2004).	37
Tabela 6 – Resultados comparativos entre ParsCit e Peng (PENG; MCCALLUM, 2004)	50
Tabela 7 – Características de cada ferramenta analisada	54
Tabela 8 – Ferramentas selecionadas para o experimento.	55
Tabela 9 – Áreas do Conhecimento (CNPq)	56
Tabela 10 – Bases de Dados informadas pelos professores entrevistados, por subárea do conhecimento.	58
Tabela 11 – Os metadados e seus pesos atribuídos	60
Tabela 12 – Descrição de cada variável no Índice de Confiabilidade	61
Tabela 13 – Resultados da ferramenta Cermine por subárea do conhecimento. . . .	66
Tabela 14 – Resultados da ferramenta CiteSeer por subárea do conhecimento. . . .	67
Tabela 15 – Resultados da ferramenta CrossRef por subárea do conhecimento. . . .	67
Tabela 16 – Resultados da ferramenta ParsCit por subárea do conhecimento. . . .	68
Tabela 17 – Índice de Confiabilidade de cada ferramenta	68
Tabela 18 – Classificação de cada ferramenta.	68
Tabela 19 – Melhores ferramentas para o metadado “Título”	73
Tabela 20 – Melhores ferramentas para o metadado “Autores”	74
Tabela 21 – Melhores ferramentas para o metadado “E-mails”	74
Tabela 22 – Melhores ferramentas para o metadado “Resumo”	75
Tabela 23 – Melhores ferramentas para o metadado “Referências”	75

Lista de abreviaturas e siglas

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

Sumário

1	INTRODUÇÃO	13
2	REFERENCIAL TEÓRICO	16
2.1	Metadados	16
2.1.1	Conceito de Metadado	16
2.1.2	Padrões de Metadados	17
2.1.3	Técnicas de Extração de Metadados	18
2.1.3.1	Regular Expressions (RegEx)	18
2.1.3.2	Support Vector Machines (SVM)	23
2.1.3.3	Hidden Markov Models (HMM)	26
2.1.3.4	Word Clustering	30
2.1.3.5	Conditional Random Fields (CRFs)	33
2.2	Trabalhos Correlatos	36
2.3	Ferramentas de Extração de Metadados	39
2.3.1	Cermine	39
2.3.2	TeamBeam	41
2.3.3	Mendeley	42
2.3.4	CiteULike	44
2.3.5	CiteSeer	47
2.3.6	ParsCit	49
2.3.7	CrossRef	50
2.3.8	Outras Ferramentas e Projetos	51
3	METODOLOGIA	55
3.1	Escolha do Corpus	55
3.2	Desenho do Experimento	58
3.2.1	Metadados, Pesos e Resultados	59
3.2.2	Índice de Confiabilidade	61
3.3	Ambiente Tecnológico	62
4	ANÁLISE E APRESENTAÇÃO DE RESULTADOS	63
4.1	Resultados	66
4.2	Ambiente de Testes	68
5	DISCUSSÃO / TRABALHOS FUTUROS	70
5.1	Contribuições	75

5.2	Trabalhos Futuros	76
5.3	Considerações Finais	77
	Referências	78
	ANEXOS	81
	ANEXO A – ELEMENTOS DO PADRÃO DUBLIN CORE, VER-	
	SÃO 1.1.	82

1 Introdução

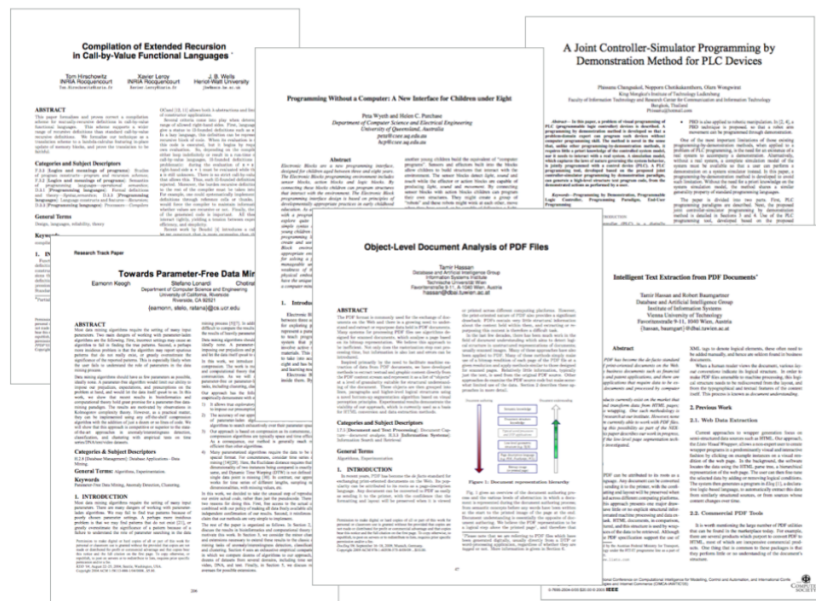
Em virtude da grande produção científica existente nos dias atuais, ferramentas automatizadas de extração de metadados em artigos científicos são cada vez mais úteis. Elas contribuem para uma melhor organização dos documentos e facilitam os processos de busca, tornando-os mais rápidos e eficientes.

A pesquisa aqui realizada situa-se no campo da extração de metadados segundo a abordagem *machine learning*. O trabalho considera as ferramentas de código aberto mais populares atualmente. Diversas ferramentas e técnicas para extração de metadados em artigos podem ser encontradas na literatura científica da área de Ciência da Informação.

Algumas ferramentas são propriedades de universidades ou instituições privadas, o que dificulta a análise. Outras não permitem que testes automatizados sejam feitos, visto que não há acesso ao código fonte ou não podem ser utilizadas via linha de comando.

De modo geral, as ferramentas de extração são focadas em leiautes pré-definidos, geralmente seguindo modelos (ou *templates*) de revistas e encontros científicos, que possuem um padrão visual já estabelecido (Figura 1). Esse 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.

Figura 1 – Exemplos de artigos da área de Ciência da Computação



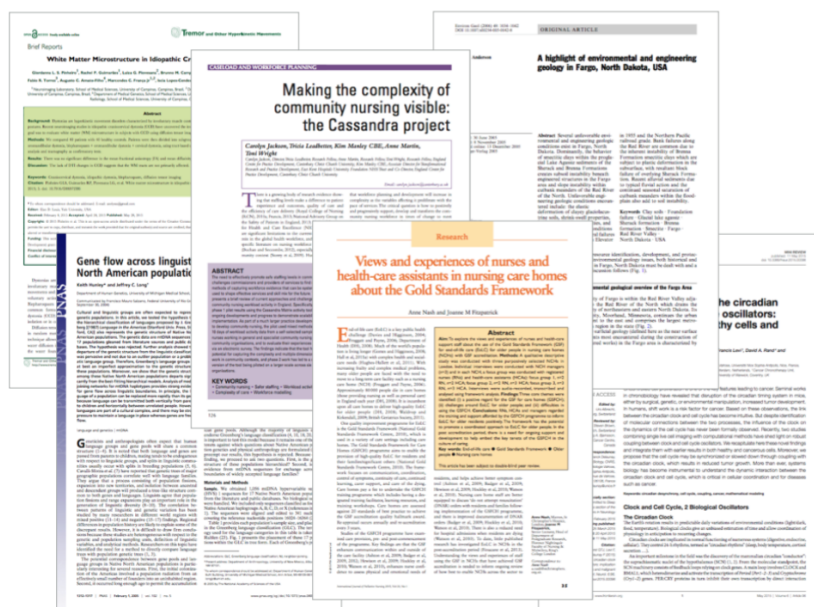
Fonte: O próprio autor

Porém, existem diversos outros eventos e revistas que empregam *templates* específicos. A extração nesses artigos exige adaptações das ferramentas para que o processo seja

satisfatório.

Algumas ferramentas são aparentemente muito eficazes para um certo grupo de artigos, já seguindo um padrão visual pré-determinado. Porém, para alguns *templates* pouco comuns, de áreas de conhecimento diversas, elas não são tão eficazes (Figura 2). A eficácia varia de acordo com a tecnologia utilizada e, principalmente, de acordo com o princípio teórico utilizado.

Figura 2 – Exemplos de artigos com padrões visuais diferentes, de diversas áreas do conhecimento



Fonte: O próprio autor

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.

A área é muito ampla e sua aplicabilidade é diversificada, abrangendo necessidades específicas da Ciência da Informação, podendo ser usada na classificação, 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. Essa última é a aplicação foco deste trabalho.

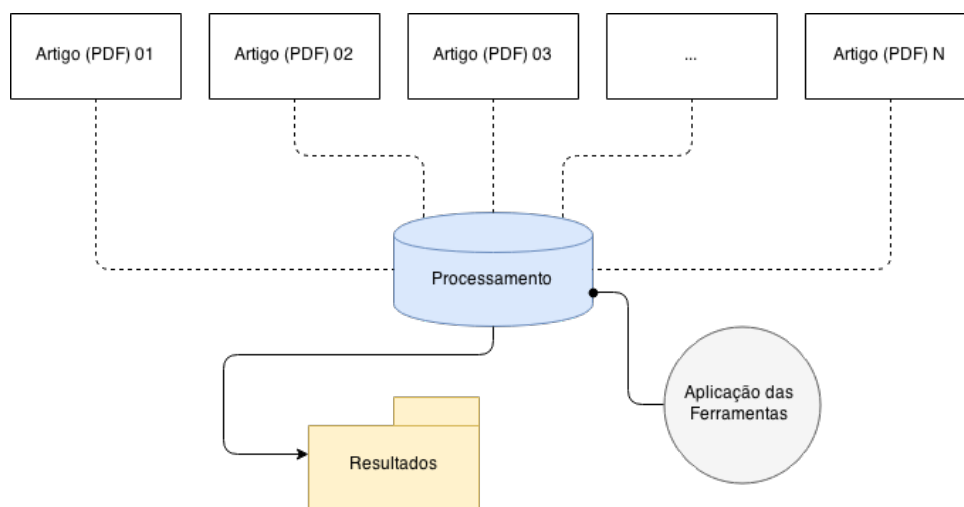
Claro que as técnicas de extração existentes hoje são, de maneira geral, insuficientes para tratar todos os leiautes de artigos existentes, limitando-se a apenas uma parcela destes, que usam padrões visuais comuns. Espera-se que certos artigos científicos não tenham seus metadados extraídos com total exatidão. Estes metadados são importantes para a produção científica, que exige análises precisas para promover o acesso à informação.

Com base na diferenciação dos leiautes de artigos científicos, o objetivo da pesquisa é comparar o desempenho de ferramentas na tarefa de extração de metadados. Isso será feito com um conjunto de documentos pré-selecionados para testes, dos mais diversos padrões e de diversas áreas do conhecimento.

Espera-se com isso poder identificar o desempenho de tais ferramentas, suas limitações e melhores aplicações: quais ferramentas apresentam melhores resultados para cada padrão visual? Que ferramenta é melhor aplicada para determinado tipo distinto de metadado?

A pesquisa é focada em técnicas e ferramentas para extração de metadados em artigos científicos. Cada ferramenta é testada juntamente com um grupo de artigos previamente selecionados. Estes artigos já possuem seus metadados extraídos manualmente, o que permite comparar os resultados com os resultados obtidos por cada uma das ferramentas. Os critérios utilizados serão de natureza explicitamente prática, numericamente representados.

Figura 3 – Processo de Extração de Metadados



Fonte: O próprio autor

O documento é estruturado iniciando com essa breve introdução e motivação sobre o tema. O segundo capítulo traz o referencial teórico, onde 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 o método usado nesta pesquisa para a realização dos 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 final e a exposição de algumas conclusões mais relevantes, além dos trabalhos futuros e considerações finais sobre o trabalho apresentado.

2 Referencial Teórico

O campo de extração de informação vem sendo bastante pesquisado e difundido. Por sua vasta aplicação, vários estudos foram e são desenvolvidos, abrangendo diversos métodos e modelos de extração, que utilizam também de PNL (Processamento Natural de Linguagem) ([SARAWAGI, 2008](#)).

Visando abranger grande parte da literatura sobre o assunto, alguns eventos e bases de dados importantes foram mapeados para a pesquisa, a fim de encontrar trabalhos relevantes para a área. Foram mapeados:

- 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 através da observação de todas as citações e referências utilizadas em cada artigo publicado, bem como outros detalhes importantes para a pesquisa.

2.1 Metadados

2.1.1 Conceito de Metadado

Com base nas ideias apresentadas por ([CATHRO, 1997](#)), “[...] um elemento de metadado descreve um recurso de informação, ou ajuda a fornecer acesso a um recurso de informação.”. Todo dado que agregue nova informação a um recurso pode ser considerado um metadado. Quanto mais metadados um recurso tiver, mais detalhado ele é, ou seja, mais dados sobre ele se tem. Podemos simplificar ainda mais a definição de metadado como sendo “um conjunto de dados sobre um determinado recurso”.

Podemos citar como exemplo a utilização de pequenos pedaços de dados sobre um conjunto de livros, dentro de um ambiente de biblioteca, o que é considerado uma coleção de elementos de metadados ([CATHRO, 1997](#)). O mesmo autor também cita como exemplos de metadados os dados coletados por mecanismos de busca no momento em que páginas da Internet são indexadas e então armazenadas.

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 metadados. Assim, foram definidos 15 (quinze) elementos para descreverem um recurso informacional, 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. 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. Suas especificações técnicas são mantidas pela *Dublin Core Metadata Initiative* (DCMI), agência responsável pela definição destes elementos.

Este padrão é utilizado para se representar um recurso na Internet (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, apresentados em forma de 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="José Alberto Grossi Júnior" >  
<meta name="DC.Title" content="Análise Comparativa de Ferramentas de  
  Extração Automatizada de Metadados em Artigos Científicos" >  
<meta name="DC.Language" content="pt_BR" >
```

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. 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.

2.1.3 Técnicas de Extração de Metadados

Algumas técnicas e algoritmos de extração de metadados são utilizadas em diversos projetos, variando de acordo com sua aplicação. Estas técnicas são baseadas na classificação de dados com base nas suas representações escritas, desde padrões preestabelecidos até com base em dicionários de palavras capazes de reconhecer ocorrências em diversas partes de um documento, o que garante assertividade ao processo de extração.

Alguns trabalhos, inclusive, utilizam segmentação de texto (*text segmentation*) para a extração de informação em campos específicos, agrupando e armazenando os resultados de forma estruturada em banco de dados ou arquivos XML, permitindo que possam ser pesquisados e analisados (CORTEZ; SILVA, 2010).

Técnicas de *machine learning*, extração e classificação de dados utilizam, em sua grande maioria, dados de entrada previamente selecionados em forma semi-estruturada. Algumas bases de dados já consolidadas no mercado fornecem estes conjuntos de dados reais, compostos por artigos científicos catalogados internamente, que são utilizados como documentos de entrada para a análise e desenvolvimento de novas pesquisas.

Com a utilização destes *datasets*¹ o processo de teste fica muito mais fácil. Existem algumas técnicas que possuem melhor desempenho ao se utilizar dos chamados *training sets*, que são por definição estes *datasets* utilizados para treinar um determinado modelo, promovendo um padrão de extração com base em dados previamente informados.

2.1.3.1 Regular Expressions (RegEx)

No caso específico de extração de metadados a utilização de *Regular Expressions* (ou Expressões Regulares) é muito eficaz no reconhecimento de padrões, como é o caso do metadado e-mail, por exemplo, que possui um formato muito específico. 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

¹ Conjuntos de dados estruturados semanticamente fornecidos por diversas entidades/órgãos para serem utilizados para fins de pesquisa.

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 então 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 também 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 (*Portable Operating System Interface*) - 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 ocorrências”.

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 pode 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 texto.

- 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.
- 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](#).

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

Além dos dígitos podemos representar também caracteres puros, utilizando-se também do operador $|$ (chamado *pipe* em inglês), que representa alternância, usado quando temos mais de uma opção. Tomemos a frase “O amor da vida de Ana se chama

Paulo”. Para identificar o reconhecimento dos nomes próprios nesta frase podemos utilizar o padrão `/Ana|Paulo/`, que quer dizer “Ana 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` - somos capazes 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 os chamados “metacaracteres”, que possuem significados definidos em uma expressão regular, assim como `+` e `?`, mas com outras formas de representação e objetivos.

- 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 vistos anteriormente - representam um conjunto de caracteres únicos dentro de várias possibilidades. Isso quer dizer que em `/[abc]/` desejamos identificar qualquer um dos caracteres `a`, `b` ou `c`. Assim, dentro da palavra `casa` conseguimos identificar a letra `c` e a letra `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`, `abcdefghijkl` 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 uma Expressão Regular o `$` indica o fim. A lógica é a mesma, se aplicando para todo o texto, e não apenas em ocorrências

isoladas. Assim, para a expressão `/^a.+z$/` exige-se que o texto comece com a letra `a` e termine com a letra `z`.

- Já os parênteses - (e) - representam grupos. Estes grupos, além de serem utilizados como no exemplo `/abaca(telexi)/`, podem ser utilizados para substituição de ocorrências, aumentando ainda mais a aplicação de expressões regulares. Como exemplo, dentro do texto “os estudantes adoram comer abacaxi depois do almoço”, podemos substituir toda ocorrência de `abacaxi` por qualquer outra palavra, 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” - os caracteres especiais que possuem significados específicos nas Expressões Regulares -, caso seja necessária a representação de algum em sua forma pura, utiliza-se do caractere de escape `\` (barra 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](#).

Tabela 2 – Classes do padrão POSIX.

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

As aplicações de Expressões Regulares são muito amplas, sendo capazes de identificar qualquer sequência que possa ser representada em forma de texto, como telefones,

endereços, dentre outras, o que facilita muito o reconhecimento de padrões, inclusive para a área de extração de dados.

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 através do reconhecimento de padrões, formando uma memória (CORTES; VAPNIK, 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 é baseada na redução de erros com base em um resultado de treinos consecutivos que permitem a criação de um padrão e estabelecem um aprendizado com base na distância entre ocorrências (CORTES; VAPNIK, 1995). Todas as análises realizadas são mapeadas, permitindo que um registro histórico em forma matemática seja realizado, levando o algoritmo à possibilidade de diferenciação numérica entre um resultado e outro.

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, por exemplo.

Esta técnica analisa diversos campos chamando-os de classes, e atribui a cada uma características que permitem que ela seja identificada. Deste modo, cada linha do cabeçalho do documento é classificada em uma ou mais classes. Algumas dessas classes fazem parte do padrão Dublin Core, conforme detalhado na [subseção 2.1.2](#).

Seymore et al. (SEYMORE; MCCALLUM; ROSENFELD, 1999) definiram 15 (quinze) *tags* para esta definição do cabeçalho de um documento. Porém, destas 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).

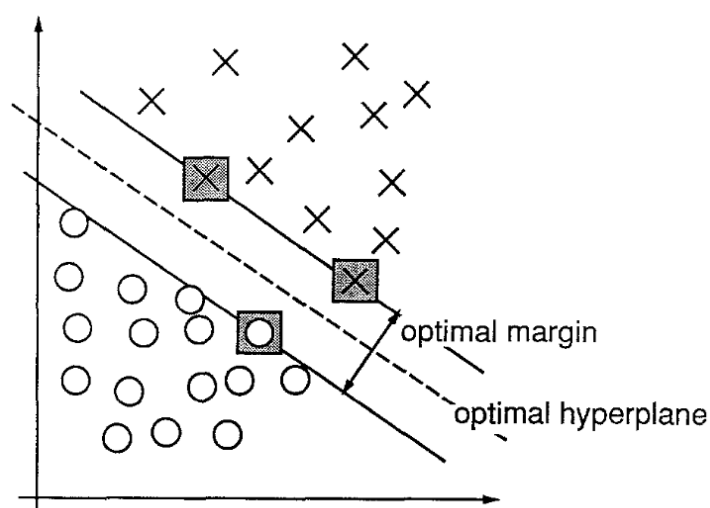
A ocorrência dos campos é mapeada em uma representação bidimensional, o que permite identificar visualmente os padrões encontrados na análise de cada classe. Desta forma, com a visualização do posicionamento de cada ocorrência é possível obter uma distância clara entre os pontos de reconhecimento, chamada pelo autor de *hyperplanes*

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, <i>copyright</i>
Email		Endereço de e-mail do autor
Date		Data da publicação
Abstract Introdution	Description	A introdução ou resumo do artigo
Phone		Telefone do autor
Keyword	Subject	As palavras-chave do documento
Web		Endereço na Web do autor
Degree		Associação com o grau acadêmico
Pubnum		Número da publicação do documento
Page		O final da página

(CORTES; VAPNIK, 1995). Por sua vez, estes pontos são marcados como sendo os “support vectors”, e permitem que o *hyperplane* entre eles determine a divisão entre as classes de forma clara e eficaz, como pode ser visto na Figura 4. Esta divisão permite então a distinção entre os metadados, diferenciando os elementos analisados pelo algoritmo.

Figura 4 – Distância representando a separação entre classes na técnica de SVM.



Fonte: (CORTES; VAPNIK, 1995)

Além desta análise também são utilizadas comparações de palavras dentro de um contexto, através do uso de *clusters* de palavras, que facilita a identificação de classes

nos cabeçalhos analisados. Han (HAN et al., 2003) utiliza em suas análises *clusters* com palavras comuns, compostos por:

- Dicionário online padrão em sistemas Linux;
- 8.441 nomes e 19.613 sobrenomes;
- Sobrenomes chineses;
- Nomes dos estados dos Estados Unidos e das províncias canadenses;
- Nomes das cidades dos Estados Unidos;
- Nome dos países do mundo, de acordo com World Fact Book²;
- Nome dos meses e suas respectivas abreviações.

Para cada uma das classes analisadas são feitas correlações com o tipo de dado esperado, que permite extrair, por exemplo, endereços de e-mail com base em expressões regulares utilizadas em linguagens de programação.

Support Vector Machines 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 comparação entre um conjunto de opções, identificando as semelhanças e permitindo classificações. Por isso, Han (HAN et al., 2003) decidiu utilizar este mesmo conceito na extração de dados, confrontando e comparando classes de metadados, para posterior identificação e diferenciação.

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 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 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.

² 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. Se um sinal é observado ele pode ser utilizado para futuras referências, de acordo com o padrão estipulado.

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, isolando totalmente qualquer outro cômodo. Este observador não consegue ver nada que acontece do outro lado, 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 ocorrências 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 para 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 - os *training datasets* -, 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, compondo toda a teoria e a 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

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

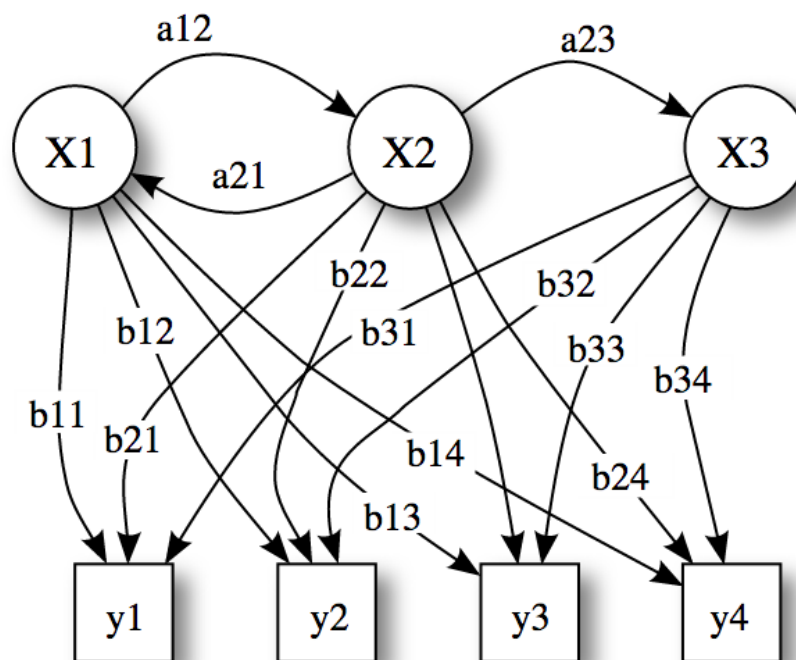
3. Após cada transição o observador registra um símbolo de acordo com a distribuição de probabilidade, que por sua vez depende do estado atual do modelo.

A utilização dos resultados passados - *training datasets* - 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 e treinado.

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; MCCALLUM; ROSENFELD, 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.

Figura 5 – 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.



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 dos modelos.

Deste modo, para extração dos metadados, o HMM pode ser utilizado aplicando-se um marcador (*label*) em cada palavra do cabeçalho de um documento (artigo científico), relacionando cada palavra a uma classe, como título, autor, etc. Assim, pode-se criar um modelo com N estados, onde cada estado corresponde 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; MCCALLUM; ROSENFELD, 1999).

Para isso seria necessário entender melhor a estrutura do modelo de acordo com os dados de treino (*training datasets*), 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; MCCALLUM; ROSENFELD, 1999).

Este modelo, segundo os autores, poderia ser usado como 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, estados correspondentes ao título seriam reunidos 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 poderia 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 melhoria deste modelo.

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

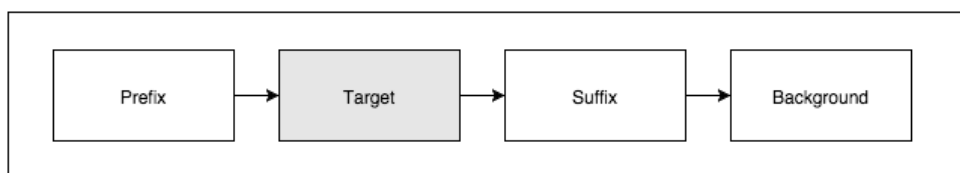
O resumo (*abstract*) é extraído facilmente com a utilização de expressão regular (subseção 2.1.3.1). Algumas classes de palavras especiais são identificadas também através de expressão regular e então são transformadas em *tags* ou *tokens*, como <EMAIL> ou <YEAR_NUMBER>, por exemplo. Em todos estes 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 modelos que utilizam apenas um estado para cada classe analisada (SEYMORE; MCCALLUM; ROSENFELD, 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 *Prefix* e *Suffix* são estados que emitem símbolos que aparecem respectivamente antes e depois desse campo-alvo. Todos os demais símbolos são emitidos no estado *Background*. A relação entre os estados pode ser visualizada na Figura 6.

Figura 6 – Estados utilizados por (ZHANG, 2001) em seu modelo HMM.



Fonte: O próprio autor.

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 seria complicado,

embora se tenha uma variedade muito grande na representação 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 explicado em função da característica dos HMMs de utilizar como modelo resultados de aprendizados anteriores, os *training datasets*. Como nestes dois campos há uma grande possibilidade de representação, 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 do HMM na extração de informação como positiva, visto as possibilidades de variação do modelo, que permite um resultado mais preciso e próximo dos objetivos reais (ZHANG, 2001). Além disso, a utilização de resultados passados garante um aprendizado 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, levando a citações independentes por alguns autores, porém, com herança conceitual dos modelos 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).

O conceito de *clusters* compreende um conjunto de palavras que formam um banco de dados de domínio (*domain database*), que é aliado a um conjunto de propriedades ortográficas de palavras dentro do 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 de performance

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

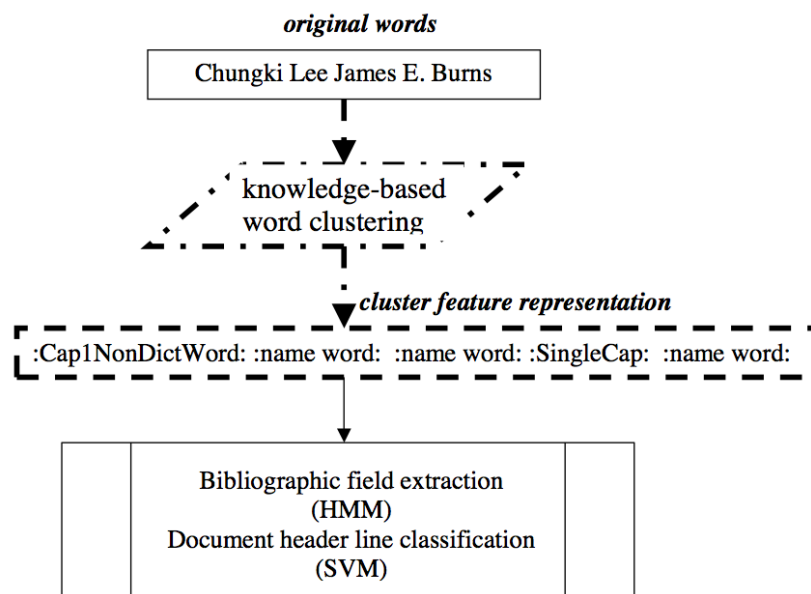
A utilização destes grupos de palavras demonstra uma relação entre textos semelhantes dentro de um determinado contexto, permitindo que a extração dos metadados ocorra de maneira natural, com resultados mais eficazes.

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

Han et al. agruparam 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 *cluster* 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 também para outras classes, como “e-mail” por exemplo, que pode ser identificado com a presença do caractere “@”, levando à utilização de expressões regulares para encontrar padrões de ocorrências (subseção 2.1.3.1).

Figura 7 – *Workflow* da extração de metadados usando *cluster* de palavras.



Fonte: (HAN et al., 2005)

A utilização de *word clustering* possui um custo computacional muito baixo, o que

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

Han et al. ainda utilizam 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, unindo as duas técnicas em prol do mesmo objetivo.

A técnica de *Word Clustering* se resume em 3 (três) etapas. A primeira compreende a construção das bases de dados, como referenciado por Han (HAN et al., 2003), onde foram utilizadas também bases externas - nomes de estados americanos, países, cidades, nomes de dicionários, nomes de pessoas, códigos postais, etc -, unindo também bases construídas dentro de um domínio específico, como palavras pertencentes a uma classe específica.

A segunda etapa é chamada de *Cluster Design*. Nesta etapa os *clusters* são arquitetados, contemplando também propriedades ortográficas das palavras, formando então o dicionário com base nas características apresentadas.

Já a terceira etapa é chamada de *Rule Design*, que consiste na combinação de palavras em diferentes domínios, nas suas verificações ortográficas e na classificação delas em 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 classes distintas, o que tornou o processo muito menos trabalhoso, porém mais 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 são apresentados dentro de 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 de classes, na separação e criação dos *clusters*, por exemplo, um dígito ou conjunto deles é substituído pela identificação `:number:`. Isso faz com que ele se torne apenas um número qualquer, sem uma semântica específica, ou seja, pode ser tanto uma referência a alguma página do documento ou até mesmo um mês ou 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 máquina (*machine learning*).

Esta técnica oferece algumas vantagens se comparada 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 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*) (LAFFERTY; MCCALLUM; PEREIRA, 2001).

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.

Lafferty et al. realizam 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 importante diferença entre CRFs e MEMMs é que MEMM utiliza modelos exponenciais como probabilidade de ocorrência de um próximo estado, enquanto a técnica de CRF possui um modelo exponencial único para toda a sequência de *labels*, com base na sequência de observação. Segundo Lafferty et al. (LAFFERTY; MCCALLUM; PEREIRA, 2001) pode-se 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 boa parte de suas qualidades, mas com a particularidade de resolver o *bias problem*. Outra grande vantagem sobre 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 também 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 àquele 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 melhor técnica para 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 funções de classificação sequencial, o que permite que 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 permitem o uso de características arbitrárias nos dados de entrada. As CRFs são utilizadas também em marcação e classificação de dados sequenciais, 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. apresentam resultados desta extração utilizando Conditional Random Fields (CRF) e aponta também algumas questões acerca da utilização desta 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, segundo os autores, a utilização de CRFs para a tarefa de extração de metadados - ao se comparar com as demais técnicas de SVM e HMM - possui melhoras significativas.

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. (PENG; MCCALLUM, 2004) também consideram como sendo o cabeçalho de um artigo como sendo a parte inicial do documento até a 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; MCCALLUM; ROSENFELD, 1999).

Para extração dos metadados dos cabeçalhos foi utilizado um *dataset* (PENG; MCCALLUM, 2004) com 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 o *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, de 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.

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* HMM possui performance bem melhor que quando utilizado SVM (98% contra 93,8%).

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

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.

2.2 Trabalhos Correlatos

Técnicas de *machine learning* não são novas, porém suas aplicações são inúmeras. Estas técnicas inicialmente eram utilizadas apenas para classificação de palavras e pro-

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%

cessamento de linguagem natural, mas 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 extração de informação acarretou em um resultado muito positivo, melhorando 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, é possível notar, em grande parte, que algumas possuem 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 entre técnicas foram feitas. Algumas - inclusive já mencionadas ao longo deste capítulo - foram feitas quando do surgimento de uma nova técnica, onde esta era comparada com técnicas anteriores. 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

tendem 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, conseqüentemente, mais passível de erros. Estas comparações são relevantes para o Estado da Arte deste trabalho.

Granitzer et al. (GRANITZER et al., 2012b) 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. Para isso são utilizados *datasets* multidisciplinares, como *Mendeley* e *e-Prints*, que fazem parte de um grupo social de *datasets*, permitindo uma contribuição global entre pesquisadores de diversas áreas do conhecimento.

Em virtude 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 (GRANITZER et al., 2012b). 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., 2012b) é uma continuação do trabalho anteriormente realizado (GRANITZER et al., 2012a), 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 (GRANITZER et al., 2012b), 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 uma das etapas principais para um bom processamento e uma boa precisão 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., 2012b) estas tarefas são 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., 2012b). 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 (GRANITZER et al., 2012b) foi o fato de a extração dos títulos ter ocorrido mais facilmente do que a extração dos autores, variando a performance dos resultados em função da técnica 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., 2012b).

De acordo com Granitzer 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., 2012b).

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

2.3 Ferramentas de Extração de Metadados

Algumas ferramentas fundamentam suas funcionalidades de extração em padrões pré-definidos, identificando dados relevantes dentro de uma região específica dos artigos, o que facilita a procura e consequentemente aumenta a velocidade nos resultados finais.

Estas ferramentas geralmente permitem uma variedade muito grande de leiautes, embora nem todos já estejam previamente definidos. Geralmente suporte a novos leiautes 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 em artigos científicos, com informações sobre sua história, funcionamento e algumas técnicas que utilizam.

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 próprio que permite que, na medida que dados forem sendo alterados, ele consiga absorver os detalhes passados e realizar mudanças em sua forma de realizar a extração. Deste modo ele permite adaptações para novos padrões de leiautes, 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 também analisa todo o seu conteúdo, incluindo citações a outros documentos, que podem ser facilmente cruzados por meio de informações como título e autor(es).

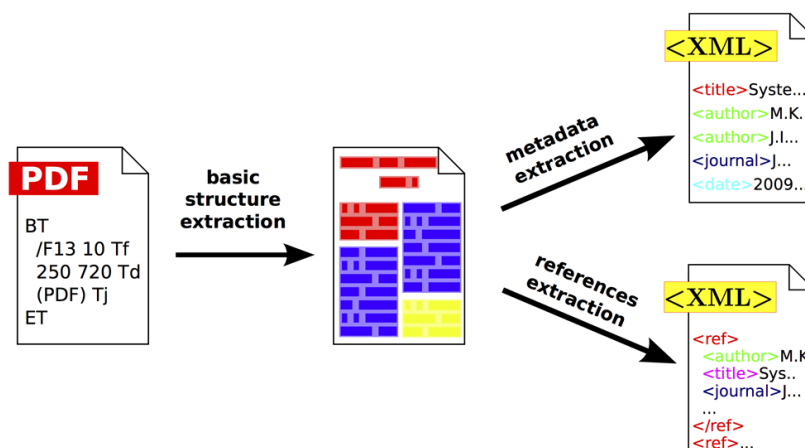
Seu mecanismo considera arquivos PDF em forma textual, sem a utilização de imagens, ou seja, não abrange documentos gerados a partir de artigos escaneados. A ferramenta considera regiões, linhas e páginas como pontos estratégicos para a extração de informações. As bases destas regiões possuem padrões que são utilizados juntamente com técnicas de SVM (HAN et al., 2003) (detalhado na [subseção 2.1.3.2](#)). Dessa forma seu mecanismo condensa um leiaute onde as informações geralmente estão dispostas, permitindo aferir que em um determinado local do arquivo estejam o título e o nome dos autores, por exemplo.

Com estas regiões definidas o Cermine extrai as informações com base em padrões preestabelecidos, gerando resultados para os metadados e referências encontradas. O formato de saída dos resultados é no formato XML, permitindo que possam ser compartilhados com outros sistemas por possuir uma leitura semântica e ao mesmo tempo fácil de ser interpretada pelas linguagens de programação. A [Figura 8](#) demonstra como o processo de extração do Cermine funciona.

Após o mapeamento definido a ferramenta identifica regiões de acordo com seu conteúdo de entrada, as quais ele chama de *zones*. Estas regiões são determinadas a fim de extrair suas informações mais relevantes, separando, por exemplo, a área destinada aos 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 identificar 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.

Figura 8 – Cerminé Extraction Workflow



Fonte: (TKACZYK et al., 2014)

- **Other:** Outros detalhes menos significantes semanticamente, como número das páginas, dentre outros dados.

A extração das referências abrange também seus próprios metadados. Tanto no texto corrido (*Body*) quanto na lista de referências do documento o *parser* do Cerminé 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 ano.

2.3.2 TeamBeam

Outra ferramenta de destaque é o TeamBeam (KERN et al., 2012), que possui uma característica bem social, contribuindo para o compartilhamento de conhecimento. O objetivo do projeto é extrair metadados de artigos científicos, como 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 PDFBox³, que fornece meios eficazes de extrair textos em arquivos PDF com base em regiões específicas dos documentos.

O TeamBeam utiliza o algoritmo de *Maximum Entropy* (BERGER; PIETRA; PIETRA, 1996), que utiliza 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).

³ Biblioteca de manipulação de arquivos PDF mantida pela Fundação Apache. Disponível em <<https://pdfbox.apache.org/>>

O processo de extração é feito em duas etapas. A primeira é a etapa de classificação de blocos de texto (*text block classification*), onde 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 leiaute do artigo alguns metadados podem vir divididos em blocos de texto diferentes, necessitando de um processamento posterior, como é o caso dos blocos com informações sobre os autores. Neste caso também é realizada a etapa de classificação de token (*token classification*), que consiste 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 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 leiautes 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 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). As três ferramentas são comparadas separadamente, por não abrangerem todos os detalhes que o TeamBeam possui, o que tornaria a comparação desleal.

Assim, eles chegam à conclusão que, em virtude das diferentes formas de processamento dos dados feitas por cada uma das ferramentas, os resultados são mais precisos para cada campo extraído. As ferramentas baseadas em dicionários apresentam melhores resultados para extração de autores, visto que baseiam-se em *data-sets*⁴ já consolidados.

2.3.3 Mendeley

Além de uma ferramenta de extração de metadados o Mendeley se transformou em uma plataforma 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 privada, 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.

⁴ Bases de dados com nomes dos autores mais citados e outras informações já catalogadas que são armazenadas para consulta pública.

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 do 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., 2012b](#)). A ferramenta utiliza um modelo de SVM de dois estágios (*two-stages SVM*) ([HAN et al., 2003](#)) para extração dos metadados, obtendo resultados bem precisos.

Em virtude de sua complexidade e por abranger uma grande quantidade de funcionalidades, o 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 tradicional no mercado, o Mendeley funciona também 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 o desenvolvimento 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 voltado à venda 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 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 na 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 exportadas em formato Microsoft Word, LibreOffice e LaTeX;
- Criação de grupos públicos e privados entre pesquisadores, permitindo compartilhamento de conteúdo entre 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, com base no foco deste trabalho o Mendeley se torna uma ferramenta pouco útil, visto que não possui o seu código aberto e não permite que sua instalação em outras máquinas para que sua funcionalidade de extração de metadados seja testada em separado. Mesmo assim, o projeto é bastante promissor e garante funcionalidades 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, promovendo uma disseminação de informação com o envio de links contendo artigos 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. Por isso o CiteULike foi criado para remover este espaço vazio entre pesquisadores, permitindo o compartilhamento de bibliografias de maneira simplificada, podendo inclusive contribuir para grupos de pesquisa, por exemplo.

Um dos detalhes desta ferramenta é o fato de funcionar com fontes externas de dados. Quando um usuário encontra na Internet um artigo científico que lhe é interessante ele pode adicioná-lo à sua lista pessoal (*bookmarking*). Assim, o CiteULike automaticamente visita o endereço informado e realiza uma extração de dados dos detalhes das citações do artigo, salvando o link para este documento 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.

Como se pode ver, o CiteULike preserva a informação livre. Ele somente coleta artigos que já estão disponíveis na Internet, com base em um link fornecido pelo próprio usuário. Outro ponto de interesse é que a ferramenta também analisa informações de citação para que possa criar um link reverso para os artigos já analisados e existentes em sua base. Deste modo, de posse de um endereço para um determinado artigo o CiteULike consegue analisar quais outros documentos este artigo cita, permitindo criar uma rede de citações, 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 possibilita que artigos sejam procurados 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. 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 documentos mais interessantes, tendo um local de destaque na página inicial da ferramenta.

Um outro 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 *metadata 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 sua grande quantidade de informação semi-estruturada. Inclusive os autores citam o fato de já existirem diversos projetos independentes 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 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 permitiu que usuários compartilhassem suas listas pessoais entre si, disseminando conhecimentos já pré-definidos 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 utilização da ferramenta, o CiteULike possibilita 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 PostgreSQL⁵, Tcl⁶ como linguagem de programação e ainda utilizando Memcached⁷ para criação de cache dos dados armazenados, o que aumentou (e muito) a performance da ferramenta. Seus servidores são baseados em Linux e os backups são feitos de 15 em 15 minutos (EMAMY; CAMERON, 2007).

Com base nesta quantidade de detalhes e vantagens o CiteULike se tornou uma ferramenta de referência para catalogação e organização de artigos científicos, permitindo

⁵ Sistema de Banco de Dados relacional com tradição e excelente performance. Sua página inicial é <<http://www.postgresql.org>>.

⁶ Linguagem de programação de scripts muito utilizada em ambientes Linux. Pode ser acessada em <<http://www.tcl.tk>>.

⁷ 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>>.

a contribuição entre pesquisadores e formando uma rede rica 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 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., 2012b).

Desta forma 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 artigos de uma maneira estatística bem eficaz.

Os índices de citação (*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 forma simples, permitindo inclusive que referências de documentos em idiomas diferentes sejam identificadas. Desta maneira, essa técnica pode ser utilizada de diversas formas, não somente na identificação de citações, mas envolvendo também um conjunto de dados, como 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.

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 entre eles, aumentando a eficiência do processo.

O funcionamento do CiteSeer é relativamente simples, porém o trabalho realizado por detrás do processo envolve muito estudo e dedicação. Ele é capaz de fazer o *download* de artigos utilizando a Internet (como um coletor), 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, identificados 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 o fato de não cobrir alguns *journals* importantes 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.

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 as representações visuais quando um outro documento é referenciado, como por exemplo: [4], [Giles997] ou “Marr 1982”. Estes pequenos pedaços de texto permitem ao CiteSeer identificar a relação 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 diferentes. Essa problemática é muito discutida na área de Ciência da Informação, que busca soluções eficazes para a desambiguação e identificação de homônimos. 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 e em desenvolvimento, abrangendo cada vez mais documentos e aprimorando cada vez mais suas técnicas de identificação e extração de metadados.

2.3.6 ParsCit

Assim como o CiteSeer o ParsCit é um projeto baseado na identificação de citações em documentos. Porém ele utiliza um modelo de CRF ([subseção 2.1.3.5](#)) 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 redor do mundo. Desenvolvido na linguagem de programação *Perl*, o projeto pode ser executado tanto na forma de um *web service*⁸ quanto de maneira *standalone*, com execução do código direto quando necessário, dentro de um servidor.

Um detalhe bastante interessante do projeto é a forma como ele processa os dados antes e depois da análise do artigo, o que os autores chamam de *Pre-Processing Steps* e *Post-Processing Steps*.

Inicialmente o processamento busca *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 o local 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-8⁹ ([COUNCILL; GILES; KAN, 2008](#)).

Desta forma a fase de pré-processamento realiza uma análise puramente textual, buscando por padrões comuns que podem ter sido utilizados para representações de referências a 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 visando normalizar os dados encontrados. Nomes de autores podem estar escritos de maneiras diferentes, com abreviações diversas, dependendo do modo como as referências foram escritas no artigo analisado.

Esta análise posterior é feita com base na inspeção de cada palavra respeitando um padrão de normalização definido, sempre com as iniciais dos nomes seguidas do sobrenome em sua forma literal.

A utilização deste projeto é feita de maneira muito simples, podendo ser executado por linha de comando, o que facilita os testes e a extração de dados objetivada neste trabalho. Os resultados obtidos desta análise possuem saídas em formato XML, o que permite utilização posterior com qualquer tecnologia.

⁸ É a disponibilização de algum serviço na Internet permitindo que outros projetos consultem seus dados e os obtenham de maneira simplificada.

⁹ Formato de exibição de caracteres que engloba os formatos mais utilizados no mundo, com acentuação e caracteres especiais.

Visando contrapor os resultados obtidos com alguns projetos em que o ParsCit foi baseado, os autores realizaram comparações com os resultados obtidos pelo processo descrito por Peng (PENG; MCCALLUM, 2004), alcançando 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).

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

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 recursos na Internet. Ela permite que referências cruzadas de citações sejam associadas com seus artigos originais, fazendo uma referência ao recurso no site de sua editora de origem (BERGMARK, 2000).

Suas atividades foram iniciadas em 1999 como uma iniciativa para 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, permitindo uma série de inferências sobre artigos, suas referências cruzadas e análises quantitativas.

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, ano de publicação, etc;
- *The Body*: O corpo do documento, onde citações podem ser analisadas de maneira quantitativa, com base na coleta de referências existentes ao final do documento;
- *The Reference Section*: Parte em que as referências completas são citadas, permitindo que ligações possam ser feitas com as citações 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 na área científica, como por exemplo, extração de metadados em artigos científicos.

Esta ferramenta de extração de metadados fornecida pelo CrossRef Labs, chamada `pdfextract`, foi criada para permitir que editoras de menor porte pudessem integrar sua bases de dados com as referências do CrossRef, aumentando ainda mais o volume de documentos indexados e extraídos. A ferramenta é desenvolvida na linguagem de programação Ruby <https://www.ruby-lang.org> e está em constante atualização, conforme pode ser visto em seu repositório oficial em <https://github.com/CrossRef/pdfextract>.

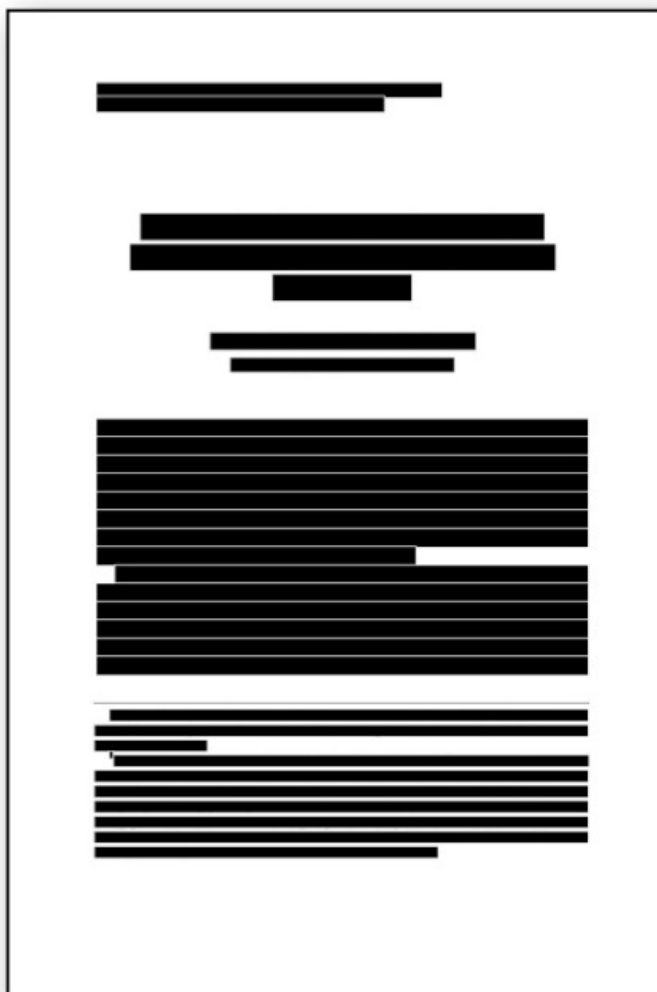
A `pdfextract` utiliza de informações visuais para identificar cada parte de um artigo científico, pré-selecionando áreas onde cada metadado pode ser encontrado, realizando então um conjunto de identificações por padrões (expressões regulares) que se espera encontrar (Figura 9).

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 foco principal 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>.

Figura 9 – Extração de Metadados com base na suposta localização de cada metadado dos artigos.



Fonte: (CROSSREF, 2009)

- **Citavi:** Permite que você organize artigos científicos de maneira bem estruturada, permitindo inclusive contribuição através de formação de times de pesquisadores. Disponível em <<http://www.citavi.com/>>.
- **BibDesk:** Exclusivo para usuários Mac/Apple. É um *software* que permite que usuários organizem seus artigos, simplificando todo o processo de exportação das referências. É focado para usuários L^AT_EX. 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 na nuvem (*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 para usuários L^AT_EX. Possui versões para Windows, Linux e Mac. Uma de suas vantagens é possuir uma interface muito simplificada, facilitando a utilização por 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 também ser acessado pela Web. Disponível em <<http://endnote.com/>>.
- **Research Gate:** Além de possuir toda uma complexidade técnica por trás o projeto é apresentado em forma de serviço, permitindo contatos e interligação entre pesquisadores. Atualmente é um dos projetos mais usados para compartilhamento de pesquisas e publicações. Disponível em <<http://researchgate.com/>>
- **FLUX-CiM:** Focado em extração de citações em artigos científicos. Uma de suas grandes vantagens é não necessitar de uma fase de treino dos dados de entrada, o que garante um alto grau de automação e flexibilidade (CORTEZ et al., 2007).

Após a análise e pesquisa das ferramentas foi criado um quadro consolidando as principais informações sobre cada uma, permitindo que as ferramentas utilizadas neste trabalho pudessem ser escolhidas. A [Tabela 7](#) contém dados como linguagem de programação utilizada, técnicas empregadas e a possibilidade de execução de cada ferramenta utilizando linha de comando.

Tabela 7 – Características de cada ferramenta analisada

Ferramenta	Linguagens de Programação	Técnicas Utilizadas	Utilização via <i>Command Line</i>
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

3 Metodologia

Este trabalho tem como metodologia uma pesquisa de caráter 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, de maneira empírica.

Primeiramente, foram selecionadas as ferramentas encontradas a fim de analisar realmente as que possuem viabilidade técnica de testes dentro do objetivo da pesquisa. Assim, de acordo com os critérios adotados, as ferramentas selecionadas foram: Cermine, CiteSeer, CrossRef e ParsCit, como pode ser visto na [Tabela 8](#).

Tabela 8 – Ferramentas selecionadas para o experimento.

Ferramenta	Linguagens de Programação	Técnicas Utilizadas
Cermine	Java	SVM, CRF, Word Clustering
CiteSeer	Python, Perl, Java	SVM, CRF (ParsCit), Word Clustering
CrossRef	Ruby, Python	Expressões Regulares, Posicionamento Espacial
ParsCit	Perl, Ruby	CRF

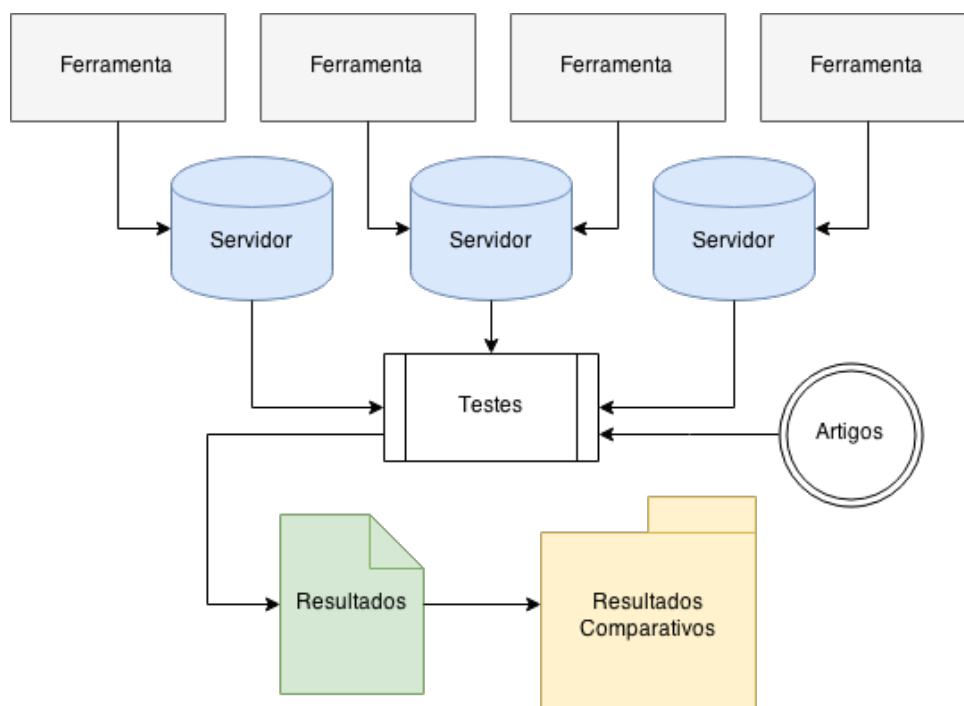
O procedimento de testes deste trabalho será realizado através da instalação e execução de cada ferramenta selecionada, permitindo que cada uma tenha seu conjunto necessário de tecnologias para seu correto funcionamento. Assim, os artigos selecionados para testes serão utilizados como dados de entrada em cada uma destas ferramentas, e seus resultados de extração coletados, comparados e analisados. Os passos necessários para a realização destes testes podem ser melhor visualizados na [Figura 10](#).

3.1 Escolha do Corpus

Visando verificar a eficiência das ferramentas - juntamente com a implementação das técnicas por elas utilizadas -, desejamos ter resultados precisos da extração de metadados, para 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, com padrões visuais distintos.

Em virtude da necessidade de realizar testes em um ambiente real e representativo, foi realizada uma pesquisa no site do CNPq <<http://www.cnpq.br/>> para obter a relação das áreas e subáreas do conhecimento reconhecidas nacionalmente. Deste modo foi

Figura 10 – Esquema visual da arquitetura do experimento



Fonte: O próprio autor

constatada a existência de 9 (nove) áreas do conhecimento, totalizando 1.290 (mil duzentas e noventa) subáreas, conforme pode ser verificado na [Tabela 9](#).

Tabela 9 – 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
Linguística, Letras e Artes	53
Outros	4
Total	1290

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

Por causa do grande número de subdivisões de cada área do conhecimento (vide [Tabela 9](#)), a seleção das subáreas foi limitada a 2 (duas), sendo 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 (dezesesseis) 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, construindo um *Corpus* realmente significativo.

Para cada uma destas subáreas do conhecimento foram coletados 7 (sete) artigos científicos. Em virtude da diversidade de bases de dados informadas pelos professores (Tabela 10), a pesquisa foi limitada a 2 (duas) bases, na ordem apresentada pelos próprios pesquisadores, contemplando 4 (quatro) artigos para a primeira base e 3 (três) para a segunda.

Para a subárea “Ciências Biológicas (Genética)” somente uma base de dados foi utilizada, portanto dela foram retirados todos os 7 artigos necessários. No total foram selecionados 112 (cento e doze) artigos, contemplando 16 (dezesesseis) subáreas do conhecimento e 32 (trinta e duas) bases de dados, formando então o *Corpus* utilizado nesta pesquisa.

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

Todos os artigos selecionados foram escritos na língua inglesa. Embora existam trabalhos locais relevantes para a área de extração de informação, esta decisão foi tomada por ser a língua mais utilizada no meio acadêmico, possuindo um universo muito maior de artigos escritos no idioma. Além disso, algumas das ferramentas e respectivas técnicas utilizam formas de “processamento de linguagem natural” para extração dos metadados, tendo por padrão a utilização do inglês na análise dos textos dos documentos de entrada.

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

Sobre a escolha das ferramentas para a realização dos testes foi utilizado apenas um critério na seleção: a sua utilização por linha de comando (*command line*). Embora algumas ferramentas possuem código aberto a extração de metadados faz parte de um contexto específico da aplicação, dificultando a utilização somente deste recurso. Assim, foram selecionadas para testes apenas as ferramentas que permitem o uso de sua funcionalidade de extração de metadados de maneira individualizada, independente da linguagem de programação ou tecnologia apresentada.

Deste modo, dentre as ferramentas apresentadas neste trabalho, presentes na

Tabela 10 – Bases de Dados informadas pelos professores entrevistados, por subárea do conhecimento.

Subárea do Conhecimento	Bases de Dados
Arquitetura e Urbanismo	Scielo, Web of Science, Scopus
Ciência da Computação	DBLP, ACM Digital Library, IEEE Xplore
Ciência da Informação	LISA, ISTA, LISTA
Ciências Biológicas (Genética)	PubMed
Ciências Biológicas (Zoologia)	Zoological Records, Biological Abstracts
Enfermagem	MedLine, Lilacs, CINAHL, EBSCO, IBECs, BDNF
Engenharia Civil	Construction and Building Materials (ELSEVIER), Cement and Concrete Composites (ELSEVIER), Composites Science and Technology (ScienceDirect), Cement and Concrete Research (ELSEVIER), Materials Research (Scielo)
Engenharia Mecânica	Scopus, ScienceDirect, Web of Science, SpringerLink, Elsevier, Research Gate
Fonoaudiologia	Pubmed, Bireme
Geologia	Springer, Scielo, Portal CAPES
História	Scielo, Jstor, Redalyc
Letras	Delta (Scielo), Periódicos Letras UFMG, Periódicos UFSC
Medicina Veterinária	PubMed, Scielo
Música	RISM, RILM, JSTOR, Grove Dictionary of Music
Psicologia	Scopus, PsycInfo, Scopus, Psycodoc
Zootecnia	Dairy Science, Animal, Poultry Science

Tabela 7, as ferramentas selecionadas para os testes foram: Cermine, CiteSeer, CrossRef e ParsCit.

3.2 Desenho do Experimento

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

Cada ferramenta foi testada em separado, observando suas características particulares. Assim, cada artigo selecionado foi testado para cada uma das ferramentas, com os respectivos resultados de cada extração. Estes resultados foram separados por metadados, o que permitiu calcular qual a porcentagem de acerto que cada ferramenta obteve na extração de cada metadado analisado.

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

No total foram analisados 112 (cento e doze) artigos científicos, para um total de 4 (quatro) ferramentas, totalizando 448 (quatrocentas e quarenta e oito) extrações de metadados através da linha de comando.

Todas as extrações foram feitas de forma automatizada, levando em consideração as necessidades de chamada de cada ferramenta, bem como os resultados de cada processamento para comparação. Todo o código criado pelo autor para este processo encontra-se disponível em <<http://github.com/jgrossi/met>>.

3.2.1 Metadados, Pesos e Resultados

Em se tratando de pesquisa por artigos científicos, pequenos detalhes podem fazer diferença. Dessa forma, uma extração de metadados não muito eficaz pode prejudicar direta ou indiretamente os resultados da busca. Por outro lado, alguns metadados tendem a ser mais utilizados em pesquisas que outros, o que implica em uma 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 - quando procuramos por artigos de um determinado pesquisador. Assim foram atribuídos pesos para cada um dos metadados, de maneira a valorizar essas informações que influenciam diretamente os resultados de busca.

A [Tabela 11](#) mostra como cada metadado foi interpretado e qual o peso que lhe foi 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) para uma pesquisa eficaz. Os pesos utilizados, assim como a ordem de importância escolhida se fundamentam apenas 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, foi 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 foi feito com o uso da função `similar_text` da linguagem de programação PHP <http://php.net/similar_text>, que calcula a porcentagem de similaridade entre dois textos de acordo com o algoritmo proposto por Oliver (OLIVER, 1993). Assim, foram comparados:

1. O dado correto, retirado manualmente dos artigos, pelo próprio autor;

Tabela 11 – 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, além do resumo propriamente dito	3
Referências	Muito importante e necessário, pois será utilizada na referência inversa de autores	4

2. O dado extraído, obtido por cada ferramenta.

Esta taxa de acerto é referenciada posteriormente como, por exemplo, $P_{\text{título}}$ (porcentagem de acertos para o metadado título). Segundo a documentação da função `similar_text` temos:

“This calculates the similarity between two strings as described in Programming Classics: Implementing the World’s Best Algorithms by Oliver (ISBN 0-131-00413-1). Note that this implementation does not use a stack as in Oliver’s pseudo code, but recursive calls which may or may not speed up the whole process. Note also that the complexity of this algorithm is $O(N^3)$ where N is the length of the longest string.”

Esta função recebe três parâmetros: o primeiro texto, o segundo texto e uma variável onde será armazenada a porcentagem de acerto. Como retorno tem-se um inteiro representando o número de caracteres comuns entre os dois textos comparados. Sua estrutura de utilização é a seguinte:

```
int similar_text ( string $first , string $second [, float &$percent ] )
```

Como cada ferramenta é testada em separado, os resultados da extração de cada artigo são gravados, tendo o total da precisão calculado de acordo com a média aritmética dos resultados obtidos para aquele metadado. Por exemplo, para a Ferramenta “A” foram analisados 100 (cem) artigos. A precisão na extração do título de cada artigo ($P_{\text{título}1}$, $P_{\text{título}2}$, ..., $P_{\text{título}N}$), por exemplo, é somada e o resultado dividido pelo número de artigos - no caso 100. Assim tem-se a precisão geral para o metadado “Título” para a Ferramenta “A” ($P_{\text{título}}$):

$$P_{\text{título}} = (P_{\text{título}1} + P_{\text{título}2} + P_{\text{título}3} \dots + P_{\text{título}100})/100$$

De posse dos acertos de cada metadado extraído podemos comparar os resultados de cada ferramenta, permitindo conclusões sobre o comportamento de cada uma perante cada metadado. Espera-se poder 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 (vide [Tabela 11](#)) é necessário calcular o índice de acertos com base nos resultados obtidos por cada ferramenta, para cada metadado. Assim chegou-se a uma fórmula matemática nomeada “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 para os artigos utilizados neste trabalho. Nele são empregados os pesos anteriormente definidos e a precisão dos resultados, permitindo chegar a uma única nota para cada ferramenta testada.

Esta fórmula é a média ponderada dos resultados alcançados na extração de cada metadado dos artigos, seguindo os pesos apresentados na [Tabela 11](#). 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 os artigos com sucesso, sua precisão com relação ao título será 87 ($P_{título} = 87$), que será multiplicada pelo peso correspondente, neste caso, o inteiro 5. Isso ocorre para todos os metadados extraídos, seguindo seus respectivos pesos. A descrição de cada variável do Índice de Confiabilidade é apresentada na [Tabela 12](#).

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

Tabela 12 – 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 classificá-las com base em seus resultados. Esta classificação não tem por objetivo qualquer favorecimento de ferramentas, mas sim classificar cada uma delas com base nos resultados obtidos

e critérios adotados neste trabalho. Desta forma, cada ferramenta foi classificada seguindo os valores abaixo:

1. **Precisa (P):** Quando o Índice de Confiabilidade é maior ou igual a 80 ($IC \geq 80$).
2. **Satisfatória (S):** Quando o Índice de Confiabilidade é maior ou igual a 60 e menor que 80 ($60 \leq IC < 80$).
3. **Insatisfatória (I):** Quando o Índice de Confiabilidade é menor que 60 ($IC < 60$).

3.3 Ambiente Tecnológico

As ferramentas testadas - por utilizarem das mesmas linguagens de programação ou por terem seus conjuntos tecnológicos semelhantes - foram instaladas em um único servidor, permitindo também que recursos computacionais fossem compartilhados, simplificando o trabalho de configuração em função de necessidades parecidas.

Este servidor foi criado através de máquina virtual, o que traz benefícios não somente de performance mas de flexibilidade quanto às tecnologias necessárias para o correto funcionamento de cada ferramenta. Este fato permitiu que os testes fossem feitos em sistemas operacionais distintos porém utilizando os mesmos recursos computacionais da máquina de origem.

4 Análise e Apresentação de Resultados

Com o Corpus totalmente definido e as ferramentas devidamente instaladas no ambiente de testes foram realizados diversos experimentos para que os resultados pudessem ser analisados e comparados numericamente.

Durante a extração dos metadados algumas observações puderam ser feitas tanto pela análise manual de cada resultado individual como também dos resultados em conjunto, tendo em vista os números apresentados pelas ferramentas utilizadas.

A ferramenta Cermine demonstrou-se de bem simples execução. Por se tratar de um arquivo em formato `.jar` (Java) em forma executável, a extração ocorreu de maneira fluida, com os dados de saída da ferramenta gravados em arquivos isolados para posterior comparação. Além disso, os resultados apresentados foram os mais completos, com utilização de diversas *tags* XML que permitiram uma fácil manipulação dos dados, com uma grande riqueza de detalhes. O processo de extração dos metadados para a ferramenta Cermine foi o mais lento das 4 (quatro) ferramentas testadas, demorando entre 15 (quinze) e 20 (vinte) segundos para uma completa análise de cada artigo.

Já a ferramenta CiteSeer foi a que mais exigiu conhecimentos técnicos para que pudesse ser testada. Sua execução dependeu da instalação de diversos outros componentes e serviços de terceiros, o que contribuiu para um aumento da complexidade de seu uso. Um fato interessante é que a ferramenta utiliza de outras ferramentas para alguns processos específicos de extração, como é o caso da sessão de referências, onde utiliza a ferramenta ParsCit. Embora a ferramenta utilizada seja a mesma a forma de entrada de dados é diferenciada, implicando em resultados numericamente diferentes.

No caso da ferramenta CrossRef algumas particularidades devem ser mencionadas. Seus resultados de extração são apresentados de maneira muito básica, com campos muito genéricos e resultados pouco precisos, dificultando um pós-processamento dos dados. Os metadados “autores”, “e-mails” e “resumo” não puderam ser extraídos. A versão atual de desenvolvimento da ferramenta não permite uma separação de dados muito específica, agrupando diversas informações em tags chamadas “sections”. Estas tags possuem informações textuais gerais, não sendo possível serem filtradas com a utilização da própria ferramenta. Portanto, para a ferramenta CrossRef somente os metadados “título” e “referências” foram extraídos e considerados. Os resultados para a extração das referências também merecem considerações, por serem apresentados de maneira muito genérica, em uma única tag XML, sendo impossível separar título e autor dentro do conteúdo retornado.

A ferramenta ParsCit também foi utilizada sem maiores dificuldades. Em virtude de sua particularidade de processar apenas dados de entrada em formato texto ou XML,

conforme sugerido pelos desenvolvedores, foi utilizada a ferramenta de linha de comando `pdftotext` (disponível em ambiente Linux) para conversão dos artigos em `.pdf` em arquivos `.txt`, permitindo que a ferramenta fosse utilizada conforme recomendações. Esta conversão foi feita em tempo de execução e os resultados coletados e gravados com sucesso.

De modo geral, exceto pela ferramenta CrossRef as demais ferramentas tiveram um processo de extração bem eficaz visualmente e dentro do esperado, em virtude da grande diferenciação visual testada com o Corpus selecionado.

No que diz respeito à comparação dos resultados foi necessária uma padronização dos dados para que as quatro ferramentas pudessem ser testadas de maneira uniforme. Em virtude de apresentar resultados bastante detalhados, a ferramenta Cermine permitiu que os autores das referências fossem retornados seguindo a forma “primeiro nome” e em seguida “sobrenome”. Já as demais ferramentas não apresentaram os resultados com tanta flexibilidade, variando em alguns momentos a ordem e disposição do nome dos autores. Assim, foi necessário um pré-processamento computacional a fim de manter, quando possível, o primeiro nome antes do sobrenome, tornando as comparações mais padronizadas e justas. Este pré-processamento foi realizado para todas as ferramentas testadas.

Já para a extração do metadado “e-mails”, algumas ferramentas extraíram mais informações em conjunto, como foi o caso de algumas poucas extrações realizadas pela ferramenta Cermine. Em um destes casos a ferramenta retornou como e-mail o seguinte conteúdo: Email: `mvpein@yahoo.com`. Assim, sempre visando uma justa comparação entre as ferramentas foi realizada uma análise em todos os resultados deste metadado para que somente pudessem ser comparados endereços de e-mails, o que tornou o processo bem simplificado e correto. Os endereços de e-mail foram filtrados com a utilização de expressão regular ([subseção 2.1.3.1](#)) alcançando um conjunto homogêneo de dados comparados.

Os demais metadados foram comparados sem problemas. O metadado “título” foi utilizado sem sua pontuação final, retirando antes da comparação qualquer caractere passível de erros como: asteriscos, pontos finais e espaços em branco. O resultado das extrações dos títulos foi feito seguindo a lógica anteriormente apresentada, comparando a similaridade entre os dois textos através do uso da função `similar_text` da linguagem de programação PHP, que apresenta como resultado um valor numérico representando o percentual de similaridade. Esta mesma lógica descrita foi aplicada para o metadado “resumo”.

Os nomes dos autores foram comparados seguindo a mesma lógica do metadado “título”, porém levando em consideração a ordem de apresentação e extração dos mesmos. Sendo assim, além de verificar a similaridade entre os nomes os testes levaram em consideração a ordem de apresentação dos resultados de cada ferramenta.

Dentro do Corpus escolhido diversos nomes de autores continham acentos e caracteres característicos de seu idioma de origem, como é o caso da autora polonesa “Anna Białk-Bielińska”. Em virtude desta questão, as ferramentas se comportaram de maneiras distintas. Algumas conseguiram extrair os nomes como no artigo original, porém, outras substituíram caracteres como “ń” por apenas “n”, ou ainda “n ´”. Algumas ferramentas simplesmente desprezaram estes caracteres.

No caso específico do metadado “e-mail” a comparação foi realizada com base na identificação correta ou não do endereço eletrônico. Neste caso não foi considerada a porcentagem de similaridade entre resultados, ou seja, o endereço foi corretamente identificado ou não. Para estes resultados foram utilizados os inteiros 0 (zero) para a extração ineficiente e 100 (cem) para a extração eficiente.

Uma grande parte dos artigos utilizados no Corpus deste trabalho não continha informações de e-mail dos autores. Desta forma, as extrações destes documentos foram desconsideradas, permitindo que as ferramentas tivessem seus resultados avaliados apenas para as extrações realmente computadas, valorizando ainda mais o trabalho de cada uma.

Já para a comparação das referências foram utilizadas duas informações: o título e o nome dos autores. Para o caso do título das referências, a lógica utilizada foi a mesma para o metadado “título”, utilizando-se de um valor percentual representando a similaridade entre os dados comparados. Já para o nome dos autores, a lógica seguiu a mesma do metadado “autores”, onde foram consideradas tanto a similaridade textual como também a ordem de apresentação. Deste modo, a extração de cada referência considerou um peso de 60% do resultado para o título e 40% para os nomes dos autores, chegando em um número final que representasse o resultado da extração de cada referência comparada.

Com os dados de cada extração armazenados a comparação foi feita de maneira automática levando em consideração todos os pontos apresentados acima. Para cada subárea do conhecimento foi realizada uma comparação, registrando o resultado consolidado para cada artigo extraído, bem como a média aritmética dos resultados daquela subárea em específico. Portanto, para cada ferramenta e subárea, foi registrado um valor médio dos resultados.

Posteriormente foi feita a coleta destes dados separados por subáreas, porém consolidando-os para cada ferramenta. Assim, foi calculada a média aritmética dos resultados de cada ferramenta para todas as subáreas analisadas, chegando então a uma nota final em cada metadado extraído, possibilitando então o cálculo do “Índice de Confiabilidade” (subseção 3.2.2) para cada ferramenta.

4.1 Resultados

Conforme esperado os resultados foram coletados de maneira individualizada - para cada artigo - e consolidados de maneira geral para cada ferramenta e metadado. Os resultados apresentados por área do conhecimento estão presentes em 4 (quatro) tabelas, separadas por cada uma das ferramentas.

Os resultados da ferramenta Cermine estão presentes na [Tabela 13](#). Os resultados da CiteSeer estão na [Tabela 14](#). Os resultados da ferramenta CrossRef na [Tabela 15](#) e da ParsCit na [Tabela 16](#). Todas as tabelas mostram o percentual de acerto separados por subárea do conhecimento e por metadados, representados pelas colunas Tit. (Título), Aut. (Autores), Ema. (E-mails), Res. (Resumo) e Ref. (Referências).

Tabela 13 – Resultados da ferramenta Cermine por subárea do conhecimento.

Subárea do Conhecimento	Tit.	Aut.	Ema.	Res.	Ref.
Arquitetura e Urbanismo	100	58.75	16.67	99.01	82.67
Ciência da Computação	88.27	71.87	21.43	98.83	77.25
Ciência da Informação	76.55	61.90	28.57	78.02	53.81
Ciências Biológicas (Genética)	91.58	81.00	50.00	84.72	96.11
Ciências Biológicas (Zoologia)	99.78	73.16	42.86	84.74	72.28
Enfermagem	99.77	39.38	16.67	98.09	81.69
Engenharia Civil	71.43	76.34	37.50	94.18	56.23
Engenharia Mecânica	99.45	75.97	58.33	77.97	82.87
Fonoaudiologia	100	77.75	71.43	98.13	80.05
Geologia	99.54	100	66.67	53.66	64.03
História	99.20	89.29	50.00	65.59	53.40
Letras	88.01	99.50	42.86	82.10	86.74
Medicina Veterinária	85.71	91.11	85.71	98.77	80.05
Música	99.03	90.61	66.67	95.47	68.50
Psicologia	88.46	63.96	47.62	92.53	63.25
Zootecnia	49.95	70.82	42.86	87.40	81.99
Média Geral	89.80	76.34	46.62	86.83	73.81

Para que os resultados pudessem ser melhor interpretados foi calculado o “Índice de Confiabilidade” de cada ferramenta, detalhado no capítulo de Metodologia ([subseção 3.2.2](#)). Para calcular este índice foram utilizadas as médias dos resultados de extração de todas as subáreas, com os devidos pesos para cada metadado, obtendo-se então uma nota geral para cada ferramenta. Os resultados calculados para este índice estão presentes na [Tabela 17](#).

De posse do “Índice de Confiabilidade” de cada ferramenta, conforme previsto na [subseção 3.2.2](#), cada uma foi classificada de acordo com seus resultados de extração. Estes resultados e suas respectivas classificações estão presentes na [Tabela 18](#).

Tabela 14 – Resultados da ferramenta CiteSeer por subárea do conhecimento.

Subárea do Conhecimento	Tit.	Aut.	Ema.	Res.	Ref.
Arquitetura e Urbanismo	100	96.89	0	97.43	70.95
Ciência da Computação	100	83.75	23.81	99.81	71.79
Ciência da Informação	84.44	99.50	0	74.12	55.56
Ciências Biológicas (Genética)	80.92	83.15	28.57	60.63	25.15
Ciências Biológicas (Zoologia)	57.14	64.12	0	71.14	70.10
Enfermagem	71.43	52.82	0	70.31	34.67
Engenharia Civil	97.81	62.42	0	71.18	35.61
Engenharia Mecânica	71.11	46.00	0	71.36	63.18
Fonoaudiologia	100	61.14	0	94.68	61.85
Geologia	73.77	34.69	0	42.79	57.13
História	99.53	71.09	0	65.26	63.81
Letras	99.57	85.73	0	75.78	58.82
Medicina Veterinária	85.71	86.38	0	98.88	63.53
Música	49.02	56.87	0	54.28	54.55
Psicologia	94.93	83.85	14.29	88.87	66.75
Zootecnia	71.43	82.41	0	76.39	22.59
Média Geral	83.55	71.93	4.17	75.81	54.75

Tabela 15 – Resultados da ferramenta CrossRef por subárea do conhecimento.

Subárea do Conhecimento	Tit.	Aut.	Ema.	Res.	Ref.
Arquitetura e Urbanismo	72.68	0	0	0	22.79
Ciência da Computação	64.19	0	0	0	14.64
Ciência da Informação	32.32	0	0	0	8.14
Ciências Biológicas (Genética)	47.05	0	0	0	14.62
Ciências Biológicas (Zoologia)	70.70	0	0	0	32.72
Enfermagem	55.96	0	0	0	10.29
Engenharia Civil	74.70	0	0	0	12.21
Engenharia Mecânica	89.27	0	0	0	27.50
Fonoaudiologia	71.43	0	0	0	13.92
Geologia	97.62	0	0	0	15.72
História	64.08	0	0	0	16.11
Letras	75.66	0	0	0	32.58
Medicina Veterinária	49.66	0	0	0	23.09
Música	84.63	0	0	0	28.16
Psicologia	82.92	0	0	0	23.19
Zootecnia	32.05	0	0	0	25.21
Média Geral	66.56	0	0	0	20.06

Tabela 16 – Resultados da ferramenta ParsCit por subárea do conhecimento.

Subárea do Conhecimento	Tit.	Aut.	Ema.	Res.	Ref.
Arquitetura e Urbanismo	0	17.14	0	73.23	51.62
Ciência da Computação	37.54	58.36	47.62	74.82	69.72
Ciência da Informação	32.30	31.51	28.57	59.60	50.09
Ciências Biológicas (Genética)	8.97	1.17	0	40.91	39.76
Ciências Biológicas (Zoologia)	0	0	0	69.98	54.61
Enfermagem	11.06	14.29	0	65.65	37.24
Engenharia Civil	11.48	15.64	37.50	56.61	37.97
Engenharia Mecânica	14.29	23.23	22.22	55.93	55.34
Fonoaudiologia	5.70	0.89	0	27.74	62.52
Geologia	14.29	14.29	16.67	68.42	55.15
História	5.62	14.29	0	59.33	60.17
Letras	24.42	42.21	21.43	68.53	54.00
Medicina Veterinária	14.29	13.82	14.29	36.57	53.31
Música	34.24	42.86	0	81.39	55.73
Psicologia	14.29	14.29	14.29	71.14	68.84
Zootecnia	14.29	5.88	0	78.60	51.21
Média Geral	15.17	19.37	12.66	61.78	53.58

Tabela 17 – Índice de Confiabilidade de cada ferramenta

Ferramenta	Resultado
Cermine	79.81
CiteSeer	68.00
CrossRef	24.30
ParsCit	33.27

Tabela 18 – Classificação de cada ferramenta.

Ferramenta	Índice de Confiabilidade	Classificação
Cermine	79.81	Satisfatória
CiteSeer	68.00	Satisfatória
CrossRef	24.30	Insatisfatória
ParsCit	33.27	Insatisfatória

4.2 Ambiente de Testes

Para a realização das extrações e das comparações foi criado um ambiente de testes contendo todas as tecnologias necessárias para que as ferramentas pudessem ser executadas dentro do esperado. Desta maneira, foi utilizado um servidor virtual com a seguinte configuração:

- Sistema Operacional Linux Ubuntu 14.04 64 Bits
- 2GB de Memória RAM
- 20GB de Espaço em Disco

As tecnologias utilizadas foram instaladas de acordo com as recomendações de cada ferramenta, com suas dependências e necessidades de cada linguagem de programação. Foram instaladas as seguintes linguagens/bibliotecas, separadas de acordo com cada ferramenta:

- **Cermine:** Java OpenJDK Runtime Environment 1.7.0_79
- **CiteSeer:** Python 2.7.6, GROBID <<https://github.com/kermitt2/grobid>>, PDF-Box <<http://pdfbox.apache.org/>>, PDF Classifier Jar, Java SE Environment (Maven).
- **CrossRef:** Ruby 2.1.2p95, RubyGem pdf-extract 0.0.1 e pdf-reader 1.3.2.
- **ParsCit:** Perl 5.18.2, G++ Compiler e CRF++ 0.51. Diversas outras dependências da linguagem Perl foram também instaladas: Class::Struct, Getopt::Long, Getopt::Std, File::Basename, File::Spec, FindBin, HTML::Entities, IO::File, POSIX, XML::Parser, XML::Twig, XML::Writer e XML::Writer::String.

5 Discussão / Trabalhos Futuros

Após todo o processo de pesquisa, de extração dos metadados pelas ferramentas analisadas e coleta de seus respectivos resultados, algumas considerações podem ser feitas, relativas aos objetivos propostos no início do trabalho.

Os resultados apresentados, de modo geral, foram inferiores às expectativas iniciais da pesquisa. As extrações não foram tão precisas quanto se imaginava. A grande diferença no leiaute dos elementos, presente no Corpus escolhido, realmente teve alto impacto nos resultados, principalmente no que diz respeito à extração dos autores e das referências.

Por outro lado, as ferramentas Cermine e CiteSeer obtiveram resultados para a extração do metadado “título” bem positivos, atingindo entre 83 e 89% de precisão. Já a ferramenta CrossRef ficou bem abaixo do esperado, com 66.56% de precisão apenas, porém acima da última colocada, a ferramenta ParsCit, que conseguiu extrair com sucesso apenas 15.17% dos resultados dos “títulos”, muito abaixo do esperado.

Para o metadado “autores” a ferramenta com maior precisão foi a Cermine, que atingiu 76.34%, resultado próximo da segunda colocada, a CiteSeer, com 71.93%. Já as demais ferramentas não obtiveram êxito na extração dos nomes dos autores, ficando abaixo dos 20% de acerto.

Para a extração dos e-mails dos autores o resultado obtido, de modo geral, foi pior. A ferramenta que obteve maior êxito na extração deste metadado foi a Cermine, que conseguiu obter apenas 46.62% de sucesso. Os resultados para este metadado obtidos pela ferramenta CiteSeer foram bem inferiores às expectativas, pois somente 4.17% dos endereços foram extraídos com sucesso, resultado inferior ainda à ferramenta ParsCit, que extraiu 12.16%. Como informado no capítulo “Resultados” ([Capítulo 4](#)) a ferramenta CrossRef não conseguiu realizar a extração de nomes de autores, endereços de e-mails e do resumo, sendo estes resultados desconsiderados nesta sessão.

Em virtude da variação de leiaute do Corpus e da ausência de padronização da formatação do metadado “resumo” (*abstract*), os resultados obtidos para este metadado superaram as expectativas. Exceto pela ferramenta CrossRef, todas as demais obtiveram resultados acima de 60%, chegando a 86.83% da ferramenta Cermine, a maior precisão encontrada.

Esses resultados para “resumo” podem ser considerados positivos, principalmente em virtude de alguns artigos apresentarem o metadado de maneira bem diferente, com posicionamento bem divergente do habitual, inclusive, sem indícios de que ali se apresentava o resumo do artigo.

Outro ponto onde as expectativas não foram atingidas foi na extração das “referências”. A ferramenta Cermine, mais uma vez, demonstrou-se mais precisa, alcançando 73.81% de sucesso. A ferramenta CiteSeer, que utiliza a ParsCit para extração das referências, ao ser comparada com a própria ParsCit, produziu resultados pouco superiores, 54.75% e 53.58%, respectivamente.

A diferença nos resultados se deve ao fato da ParsCit necessitar de arquivos `.txt` como forma de entrada de dados. No caso das extrações realizadas pela própria ferramenta, os arquivos `.txt` foram gerados pelo programa `pdftotext`, conforme detalhado no capítulo de “Resultados” (Capítulo 4), diferentemente da ferramenta CiteSeer, que transforma o arquivo `.pdf` em `.txt` de sua própria maneira, causando então uma pequena divergência nos resultados gerais (1.17%).

Já a ferramenta CrossRef obteve apenas 20.06% de precisão na extração das referências, o que era esperado em função de seus resultados com poucos detalhes, com apenas um único campo com todas as informações de cada referência.

Embora estes resultados da extração da ferramenta CrossRef não tenham sido positivos, um detalhe interessante que merece atenção é a forma como a ferramenta trata as referências. A ferramenta CrossRef permite que elas sejam comparadas com o banco de dados existente na URL <http://api.crossref.org>, possibilitando identificar exatamente quais artigos já foram catalogados pelo site, gerenciando seu conteúdo e relacionando-o a outros documentos.

Para os artigos encontrados na base de dados do CrossRef é possível obter, inclusive, a descrição de cada um em formato BibTeX. Para este trabalho, em virtude dos poucos resultados obtidos, e por se tratar de um Corpus específico, esta funcionalidade não foi utilizada na extração ou na comparação dos resultados para esta ferramenta.

Em se tratando da separação dos resultados por área do conhecimento as ferramentas Cermine e CiteSeer obtiveram destaque, conseguindo 100% de acertos em 4 (quatro) subáreas do conhecimento - Arquitetura e Urbanismo, Geologia, Fonoaudiologia, e Ciência da Computação -, porém para metadados diferentes.

A ferramenta Cermine acertou todos os títulos das áreas de Arquitetura e Urbanismo e Fonoaudiologia, além de 100% dos nomes dos autores da área de Geologia. A ferramenta CiteSeer conseguiu precisão total na extração dos títulos de Arquitetura e Urbanismo, Ciência da Computação e Fonoaudiologia.

Já a ferramenta CrossRef obteve melhor resultado na extração dos títulos dos artigos da área de Geologia, obtendo 97.62% de precisão, superando as ferramentas CiteSeer e ParsCit, que obtiveram 73.77% e 14.29%, respectivamente.

Para a extração dos títulos dos artigos, os piores resultados foram encontrados nas áreas de Música (CiteSeer, com 49.02%), Zootecnia (Cermine, com 49.95% e CrossRef,

com 32.05%) e as áreas Ciências Biológicas (Zoologia) e Arquitetura e Urbanismo (ParsCit, com nenhum acerto).

A ferramenta Cermine se destacou na extração dos títulos em 8 (oito) subáreas do conhecimento - Arquitetura e Urbanismo, Ciências Biológicas (Zoologia), Enfermagem, Engenharia Mecânica, Fonoaudiologia, Geologia, História e Música -, obtendo resultados superiores a 99%, o que foi considerado excelente.

Para a extração dos nomes dos autores, os maiores destaques foram nas áreas de Geologia, Letras, Medicina Veterinária e Música, com resultados acima de 90%.

Na extração dos e-mails dos autores a ferramenta Cermine obteve resultados superiores a 85% somente na área de Medicina Veterinária, seu melhor resultado para este metadado. Além disso, a ferramenta destacou-se na extração dos resumos em 5 (cinco) áreas, com resultados acima dos 98%, e na extração das referências da área de Ciências Biológicas (Genética), onde obteve resultados acima de 96% de precisão.

Já a ferramenta CiteSeer foi bem eficiente na extração dos títulos de 5 (cinco) subáreas: Arquitetura e Urbanismo, Ciência da Computação, Fonoaudiologia, História e Letras, com resultados superiores a 99%.

Para a extração dos nomes dos autores o resultado foi relevante em apenas 2 (duas) subáreas: Arquitetura e Urbanismo e Ciência da Informação, com precisão acima de 90%.

Já para os e-mails dos autores os resultados deixaram a desejar em 13 (treze) das 16 (dezesesseis) subáreas, com 0% de acerto, tendo resultados positivos apenas para as subáreas de Ciência da Computação, Ciências Biológicas (Genética) e Psicologia, porém com resultados abaixo de 29% de acerto.

Para a extração dos resumos a ferramenta CiteSeer também se mostrou bem eficiente, com resultados acima de 90% para 4 (quatro) subáreas: Arquitetura e Urbanismo, Ciência da Computação, Fonoaudiologia e Medicina Veterinária.

Para as referências (utilizando o ParsCit) os resultados deixaram a desejar, com acertos abaixo de 72%.

A ferramenta CrossRef mostrou resultados positivos apenas para a extração de títulos de artigos da subárea de Geologia, como já dito anteriormente, com 97.62% de acerto, não tendo resultados considerados satisfatórios para as demais áreas.

Para a extração das referências os resultados deixaram a desejar, com apenas 2 (duas) subáreas com precisão próxima de 30%: Ciências Biológicas (Zoologia), com 32.72% e Letras, com 32.58%.

Por fim, a ferramenta ParsCit obteve resultados abaixo dos 38% para os títulos, em todas as subáreas analisadas.

O acerto dos nomes dos autores também foi baixo, onde os melhores resultados ficaram entre 43% e 59%, para as subáreas de Ciência da Computação, Letras e Música.

Para os e-mails dos autores o melhor resultado foi para os artigos da subárea de Ciência da Computação, com 47.62% de precisão.

Já para o resumo dos artigos, os resultados foram um pouco melhores, acima de 70% para 5 (cinco) subáreas do conhecimento.

Os resultados para as referências foram semelhantes aos obtidos pela ferramenta CiteSeer, que utiliza a mesma ferramenta. Os 2 (dois) melhores resultados foram para as subáreas de Ciência da Computação e Psicologia, com precisão de 69.72% e 68.84%, respectivamente.

Para melhor visualização dos resultados, as Tabelas 19, 20, 21, 22 e 23 apresentam as ferramentas que obtiveram os melhores resultados para cada subárea do conhecimento, separados por metadado.

Tabela 19 – Melhores ferramentas para o metadado “Título”

Subáreas do Conhecimento	Ferramentas	Precisão
Arquitetura e Urbanismo	Cermine/CiteSeer	100%
Ciência da Computação	CiteSeer	100%
Ciência da Informação	CiteSeer	84.44%
Ciências Biológicas (Genética)	Cermine	91.58%
Ciências Biológicas (Zoologia)	Cermine	99.78%
Enfermagem	Cermine	99.77%
Engenharia Civil	CiteSeer	97.81%
Engenharia Mecânica	Cermine	99.45%
Fonoaudiologia	Cermine/CiteSeer	100%
Geologia	Cermine	99.54%
História	CiteSeer	99.53%
Letras	CiteSeer	99.57%
Medicina Veterinária	Cermine/CiteSeer	85.71%
Música	Cermine	99.03%
Psicologia	CiteSeer	94.93%
Zootecnia	CiteSeer	71.43%

Os resultados mostram que, para o Corpus escolhido, para o metadado “Título”, a ferramenta Cermine foi superior, com 89.8% de precisão, seguida da CiteSeer, com 83.55%. O mesmo acontece para o metadado “Autores”, onde a Cermine obteve os melhores resultados (76.34%), seguida da CiteSeer, com 71.93%.

Já para o metadado “E-mails” a Cermine foi sem dúvida a melhor, com 46.62% de acertos, deixando uma grande diferença para a segunda colocada ParsCit, com apenas 12.66%. Para o metadado “Resumo” a Cermine também se saiu melhor, com 86.83% de

Tabela 20 – Melhores ferramentas para o metadado “Autores”

Subáreas do Conhecimento	Ferramentas	Precisão
Arquitetura e Urbanismo	CiteSeer	96.89%
Ciência da Computação	CiteSeer	83.75%
Ciência da Informação	CiteSeer	99.50%
Ciências Biológicas (Genética)	CiteSeer	83.15%
Ciências Biológicas (Zoologia)	Cermin	73.16%
Enfermagem	CiteSeer	52.82%
Engenharia Civil	Cermin	76.34%
Engenharia Mecânica	Cermin	75.97%
Fonoaudiologia	Cermin	77.75%
Geologia	Cermin	100%
História	Cermin	89.29%
Letras	Cermin	99.50%
Medicina Veterinária	Cermin	91.11%
Música	Cermin	90.61%
Psicologia	CiteSeer	83.85%
Zootecnia	CiteSeer	82.41%

Tabela 21 – Melhores ferramentas para o metadado “E-mails”

Subáreas do Conhecimento	Ferramentas	Precisão
Arquitetura e Urbanismo	Cermin	16.67%
Ciência da Computação	ParsCit	47.62%
Ciência da Informação	Cermin/ParsCit	28.57%
Ciências Biológicas (Genética)	Cermin	50.00%
Ciências Biológicas (Zoologia)	Cermin	42.86%
Enfermagem	Cermin	16.67%
Engenharia Civil	Cermin/ParsCit	37.50%
Engenharia Mecânica	Cermin	58.33%
Fonoaudiologia	Cermin	71.43%
Geologia	Cermin	66.67%
História	Cermin	50.00%
Letras	Cermin	42.86%
Medicina Veterinária	Cermin	85.71%
Música	Cermin	66.67%
Psicologia	Cermin	47.62%
Zootecnia	Cermin	42.86%

precisão, e em segunda posição a CiteSeer com 75.81%.

Por fim, para a extração do metadado “Referências” novamente a Cermin obteve o melhor resultado, acertando 73.81% das extrações, seguida pela CiteSeer com 54.75%.

Tabela 22 – Melhores ferramentas para o metadado “Resumo”

Subáreas do Conhecimento	Ferramentas	Precisão
Arquitetura e Urbanismo	Cerminate	99.01%
Ciência da Computação	CiteSeer	99.81%
Ciência da Informação	Cerminate	78.02%
Ciências Biológicas (Genética)	Cerminate	84.72%
Ciências Biológicas (Zoologia)	Cerminate	84.74%
Enfermagem	Cerminate	98.09%
Engenharia Civil	Cerminate	94.18%
Engenharia Mecânica	Cerminate	77.97%
Fonoaudiologia	Cerminate	98.13%
Geologia	Cerminate	53.66%
História	Cerminate	65.59%
Letras	Cerminate	82.10%
Medicina Veterinária	CiteSeer	98.88%
Música	Cerminate	95.47%
Psicologia	Cerminate	92.53%
Zootecnia	Cerminate	87.40%

Tabela 23 – Melhores ferramentas para o metadado “Referências”

Subáreas do Conhecimento	Ferramentas	Precisão
Arquitetura e Urbanismo	Cerminate	82.67%
Ciência da Computação	Cerminate	77.25%
Ciência da Informação	CiteSeer	55.56%
Ciências Biológicas (Genética)	Cerminate	96.11%
Ciências Biológicas (Zoologia)	Cerminate	72.28%
Enfermagem	Cerminate	81.69%
Engenharia Civil	Cerminate	56.23%
Engenharia Mecânica	Cerminate	82.87%
Fonoaudiologia	Cerminate	80.05%
Geologia	Cerminate	64.03%
História	CiteSeer	63.81%
Letras	Cerminate	86.74%
Medicina Veterinária	Cerminate	80.05%
Música	Cerminate	68.50%
Psicologia	ParsCit	68.84%
Zootecnia	Cerminate	81.99%

5.1 Contribuições

Como dito, os resultados coletados após as comparações ficaram abaixo das expectativas, exceto pelo metadado “Título”, onde os números foram bem expressivos.

Em virtude da grande diferença no posicionamento visual dos elementos dos

artigos do Corpus, os resultados foram muito variáveis, não sendo possível definir, com precisão, qual ferramenta se comporta melhor para uma determinada área ou subárea do conhecimento, mesmo que os resultados demonstrem, numericamente, o comportamento diferenciado de cada uma.

Estes resultados permitem aferir que as ferramentas de extração de metadados ainda tem espaço para evoluir, sendo necessários ajustes e adaptações para que uma maior quantidade de metadados seja extraída com sucesso.

Além disso, algumas ferramentas apresentam melhores resultados em algumas subáreas do conhecimento, mas sem generalização possível, o que demandaria uma análise mais aprofundada dos resultados e uma seleção dos artigos mais específica.

Todo o código criado para a extração dos metadados e comparação das ferramentas está disponível em [<http://github.com/jgrossi/met>](http://github.com/jgrossi/met), podendo ser utilizado para futuras pesquisas. É possível incluir novas ferramentas e metadados de maneira simplificada, reaproveitando todo o processo de comparação utilizado neste trabalho.

Ademais, todo o processo de comparação elaborado neste trabalho pode ser também reutilizado, permitindo inclusive o cálculo do Índice de Confiabilidade segundo os critérios adotados pelo autor. O índice permite a classificação de uma ferramenta segundo pesos definidos para cada metadado (subseção 3.2.2).

Por fim, pôde-se observar que o comportamento das técnicas de extração utilizadas pelas ferramentas é muito variável. Uma parcela dos resultados parece ser influenciada pelo modo de uso de cada técnica em cada ferramenta. Porém, deve-se levar em consideração a maneira como os algoritmos de cada técnica são implementados, bem como a maneira como os dados são tratados, tanto antes quanto depois da realização da extração. Assim, com base nos resultados numéricos apresentados, não é possível determinar qual técnica é mais aplicada para a extração de cada metadado.

5.2 Trabalhos Futuros

Embora este trabalho tenha abrangido 16 (dezesesseis) subáreas do conhecimento, com um total de 112 (cento e doze) artigos científicos analisados, a variedade real de formatos e leiautes vai muito além.

Poderia-se pensar em trabalhos mais detalhados para cada subárea do conhecimento, permitindo testar uma maior quantidade de artigos e padrões, de maneira a obter resultados mais próximos da realidade de cada área. Assim, seria possível identificar o comportamento destas ferramentas para cada área em específico, com um Corpus bem maior e variado, porém mais direcionado.

Um outro estudo possível seria a comparação por periódicos (revistas) ou bases de

dados. Embora a diferenciação de leiaute para uma área do conhecimento seja muito ampla, geralmente existe uma padronização visual para uma determinada base, como é o caso da Elsevier <<http://www.elsevier.com>>, onde, independente da área do conhecimento, os artigos passam por uma padronização visual, permitindo adaptações mais precisas para as extrações destes documentos.

Apesar de selecionadas as quatro ferramentas aqui comparadas, existem muitas outras ferramentas que merecem atenção, possibilitando um estudo de caso focado para uma determinada ferramenta, aprofundando muito mais suas características, funcionalidades e implementações, permitindo conclusões mais direcionadas e críticas mais precisas.

Seria ainda interessante estender a pesquisa considerando possíveis variações na extração manual dos metadados. Embora os dados tenham sido extraídos de maneira bem cautelosa é possível que as mesmas extrações, realizadas por pessoas diferentes, produzam resultados variados, o que poderia alterar em pequena proporção os resultados apresentados.

5.3 Considerações Finais

Em virtude dos resultados apresentados e com base nas comparações realizadas, sugere-se que as ferramentas ainda tem espaço para evoluir, a fim de abranger um maior número de artigos e áreas do conhecimento.

Embora algumas ferramentas tenham se comportado melhor para alguns padrões visuais, não é possível estabelecer uma regra ou afirmação com base nos resultados encontrados.

Além disso, dadas as fragilidades das ferramentas testadas, sugere-se que o desenvolvimento de uma nova solução poderia ser de interesse, possibilitando uma análise mais profunda das necessidades de cada metadado, bem como das técnicas mais adequadas em cada aplicação.

Referências

BERGER, A. L.; PIETRA, V. J. D.; PIETRA, S. A. D. A maximum entropy approach to natural language processing. *Comput. Linguist.*, MIT Press, Cambridge, MA, USA, v. 22, n. 1, p. 39–71, mar 1996. ISSN 0891-2017. Disponível em: <http://dl.acm.org/citation.cfm?id=234285.234289>. Citado 2 vezes nas páginas 30 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. *First Monday*, v. 2, n. 4, 1997. Disponível em: <http://dblp.uni-trier.de/db/journals/firstmonday/firstmonday2.html#Cameron97>. Citado na página 47.

CATHRO, W. Metadata: an overview. 1997. Disponível em: <http://www.nla.gov.au/openpublish/index.php/nlaspp/article/view/1019/1289>. Citado na página 16.

CHIEU, H. L.; NG, H. T. A maximum entropy approach to information extraction from semi-structured and free text. In: *Eighteenth National Conference on Artificial Intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 2002. p. 786–791. ISBN 0-262-51129-0. Disponível em: <http://dl.acm.org/citation.cfm?id=777092.777213>. Citado na página 23.

CORTES, C.; VAPNIK, V. Support-vector networks. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 20, n. 3, p. 273–297, sep 1995. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A:1022627411411>. Citado 2 vezes nas páginas 23 e 24.

CORTEZ, E.; SILVA, A. S. da. Unsupervised strategies for information extraction by text segmentation. In: *Proceedings of the Fourth SIGMOD PhD Workshop on Innovative Database Research*. New York, NY, USA: ACM, 2010. (IDAR '10), p. 49–54. ISBN 978-1-4503-0191-6. Disponível em: <http://doi.acm.org/10.1145/1811136.1811145>. Citado na página 18.

CORTEZ, E. et al. Flux-cim: Flexible unsupervised extraction of citation metadata. In: *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*. New York, NY, USA: ACM, 2007. (JCDL '07), p. 215–224. ISBN 978-1-59593-644-8. Disponível em: <http://doi.acm.org/10.1145/1255175.1255219>. Citado na página 53.

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 49.

CROSSREF. *A short history of CrossRef*. [S.l.], 2009. Disponível em: <http://www.crossref.org/08downloads/CrossRef10Years.pdf>. Citado na página 52.

EMAMY, K.; CAMERON, R. 'Citeulike: A Researcher's Social Bookmarking Service'. *Ariadne Issue 51*, n. Issue 51, apr 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 45.

GILES, C. L.; BOLLACKER, K. D.; LAWRENCE, S. Citeseer: An automatic citation indexing system. In: *Proceedings of the Third ACM Conference on Digital Libraries*. New York, NY, USA: ACM, 1998. (DL '98), p. 89–98. ISBN 0-89791-965-3. Disponível em: <<http://doi.acm.org/10.1145/276675.276685>>. Citado na página 47.

GRANITZER, M. et al. A comparison of metadata extraction techniques for crowdsourced bibliographic metadata management. In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2012. (SAC '12), p. 962–964. ISBN 978-1-4503-0857-1. Disponível em: <<http://doi.acm.org/10.1145/2245276.2245462>>. Citado na página 38.

GRANITZER, M. et al. A comparison of layout based bibliographic metadata extraction techniques. In: *Proceedings of the 2Nd International Conference on Web Intelligence, Mining and Semantics*. New York, NY, USA: ACM, 2012. (WIMS '12), p. 19:1–19:8. ISBN 978-1-4503-0915-8. Disponível em: <<http://doi.acm.org/10.1145/2254129.2254154>>. Citado 4 vezes nas páginas 38, 39, 43 e 47.

GROUP, T. O. *The Open Group Base Specifications Issue 7*. 2013. Disponível em: <<http://pubs.opengroup.org/onlinepubs/9699919799/>>. Citado na página 19.

HAN, H. et al. Automatic document metadata extraction using support vector machines. In: *Digital Libraries, 2003. Proceedings. 2003 Joint Conference on*. [S.l.: s.n.], 2003. p. 37–48. Citado 5 vezes nas páginas 23, 25, 32, 40 e 43.

HAN, H. et al. Rule-based word clustering for document metadata extraction. In: *Proceedings of the 2005 ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2005. (SAC '05), p. 1049–1053. ISBN 1-58113-964-0. Disponível em: <<http://doi.acm.org/10.1145/1066677.1066917>>. Citado 3 vezes nas páginas 30, 31 e 32.

HENNING, V.; REICHELT, J. Mendeley - a last.fm for research? In: *Proceedings of the 2008 Fourth IEEE International Conference on eScience*. Washington, DC, USA: IEEE Computer Society, 2008. (eSCIENCE '08), p. 327–328. ISBN 978-0-7695-3535-7. Disponível em: <<http://dx.doi.org/10.1109/eScience.2008.128>>. Citado na página 43.

KERN, R. et al. Teambeam meta-data extraction from scientific literature. *D-Lib Magazine*, Corporation for National Research Initiatives, v. 18, n. 7, p. 1, 2012. Citado na página 41.

KLEENE, S. C. Representation of events in nerve nets and finite automata. *Automata Studies*, p. 3–42, 1956. Citado na página 18.

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 17.

LAFFERTY, J. D.; MCCALLUM, A.; PEREIRA, F. C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. (ICML '01), p. 282–289. ISBN 1-55860-778-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=645530.655813>>. Citado 3 vezes nas páginas 30, 33 e 34.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of Machine Learning*. [S.l.]: The MIT Press, 2012. ISBN 026201825X, 9780262018258. Citado na página 14.

OLIVER, I. *Programming Classics: Implementing the World's Best Algorithms*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993. ISBN 0-13-100413-1. Citado na página 59.

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 7 vezes nas páginas 9, 34, 35, 36, 37, 41 e 50.

RABINER, L. R.; JUANG, B. H. An introduction to hidden markov models. *IEEE ASSp Magazine*, 1986. Citado na página 26.

SARAWAGI, S. Information extraction. *Found. Trends databases*, Now Publishers Inc., Hanover, MA, USA, v. 1, n. 3, p. 261–377, mar. 2008. ISSN 1931-7883. Disponível em: <<http://dx.doi.org/10.1561/19000000003>>. Citado na página 16.

SEYMORE, K.; MCCALLUM, A.; ROSENFELD, R. Learning hidden markov model structure for information extraction. In: *In AAAI 99 Workshop on Machine Learning for Information Extraction*. [S.l.: s.n.], 1999. p. 37–42. Citado 5 vezes nas páginas 23, 27, 28, 29 e 35.

THOMPSON, K. Programming techniques: Regular expression search algorithm. *Commun. ACM*, ACM, New York, NY, USA, v. 11, n. 6, p. 419–422, jun. 1968. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/363347.363387>>. Citado na página 19.

TKACZYK, D. et al. Cermine – automatic extraction of metadata and references from scientific literature. In: *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*. [S.l.: s.n.], 2014. p. 217–221. Citado 2 vezes nas páginas 39 e 41.

WEIBEL, S. The dublin core: A simple content description 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 17.

ZHANG, N. R. Hidden markov models for information extraction. 2001. Citado 3 vezes nas páginas 8, 29 e 30.

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.