HEITOR DE SOUZA GANZELI

METODOLOGIA PARA A ANÁLISE DA QUALIDADE DE WEB SITES BASEADA EM
TÉCNICAS DE APRENDIZADO DE MÁQUINA

São Paulo
2014

HEITOR DE SOUZA GANZELI

METODOLOGIA PARA A ANÁLISE DA QUALIDADE DE WEB SITES BASEADA EM
TÉCNICAS DE APRENDIZADO DE MÁQUINA

Dissertação apresentada à Escola
Politécnica da Universidade de São Paulo
para obtenção do título de Mestre em
Engenharia Elétrica

São Paulo
2014

HEITOR DE SOUZA GANZELI

METODOLOGIA PARA A ANÁLISE DA QUALIDADE DE WEB SITES BASEADA EM
TÉCNICAS DE APRENDIZADO DE MÁQUINA

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de Mestre em
Engenharia Elétrica

Área de Concentração:
Engenharia de Computação

Orientador: Profa. Dra. Graça Bressan

São Paulo
2014

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, de abril de 2014.

Assinatura do autor _____

Assinatura do orientador _____

Catálogo-na-publicação

Ganzeli, Heitor de Souza

**Metodologia para a análise da qualidade de web sites baseada em técnicas de aprendizado de máquina / H.S. Ganzeli. -- versão corr. -- São Paulo, 2014.
148 p.**

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

**1.Web sites (Qualidade) 2. Aprendizado computacional
3.Padrões abertos 4.Padrões web I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.**

RESUMO

A Web é a aplicação mais popular da Internet e, desde sua criação, gerou mudanças de diversas maneiras na vida das pessoas. Esse é um dos motivos que a transformou em objeto de estudo de diversas pesquisas de cunho social, tecnológico econômico e político. A metodologia descrita nesta dissertação pode ser entendida como uma extensão do projeto TIC Web, que foi desenvolvido como parceria entre o NIC.br, o escritório do W3C Brasil e o instituto InWeb para estudar características de qualidade da Web Brasileira. Nesse sentido, a presente metodologia possui o objetivo de automatizar análises de domínios e *sites* Web, principalmente com base nos resultados sobre a Web Governamental Brasileira obtidos pelo TIC Web. Ou seja, o presente trabalho se foca na definição e aplicação de metodologia baseada em técnicas de aprendizado de máquina para a automatização das análises de domínios Web, visando praticidade na execução da categorização de *sites* Web segundo critérios relacionados à qualidade percebida por seus usuários. Os tópicos aqui discutidos compreendem: a importância dos padrões abertos e elementos de desempenho para a determinação da qualidade de um *site*; fundamentos de aprendizado de máquina; o detalhamento das ferramentas utilizadas para coletar e extrair informações dos *sites*, bem como dos atributos e indicadores por elas adquiridos; a metodologia proposta, incluindo a descrição de diversos algoritmos utilizados; e, um caso de uso demonstrando sua aplicabilidade. Além disso, propõe-se como parte da metodologia de análise a utilização dos resultados de seus resultados para realizar a avaliação de *sites* segundo sua qualidade percebida.

Palavras-chave: padrões abertos, padrões Web, qualidade de *sites*, aprendizado de máquina.

ABSTRACT

The World Wide Web is the most popular application throughout the Internet and, since its creation, it has changed people's lives in lots of ways, hence, it has become subject to several social, technological, economical and political researches. The methodology described in the present text may be understood as an extension of the TIC Web project, which was developed by a partnership among NIC.br, Brazilian W3C office and the InWeb institute in order to study some quality related issues about the Brazilian Web. Accordingly, the methodology presented in this work aims to automate analyses of Web domains and sites, mainly based on the results over the Brazilian Governmental Web obtained by TIC Web. In other words, the present project focus on the definition and use of a methodology dependent on machine learning in order to automate the analyses of extracted data, having the goal of easing the classification of Web sites according to the quality perceived by their users. Some of the discussed topics are as follows: the importance of Open Standards and performance features to defy the quality of a *site*; basics of machine learning; details of the tool applied to extract Web sites data, as well as its acquired parameters and indicators; the proposed methodology, including the description of applied algorithms; and a use case evincing its applicability. Additionally, it is proposed, as part of the methodology, the utilization of the results obtained by the domain analysis to evaluate other websites in accordance to their perceived quality.

Key words: open standards, Web standards, web quality, machine learning.

LISTA DE ILUSTRAÇÕES

Figura 1.1 – Mapa mental da estrutura do presente trabalho.....	26
Figura 2.1 - Protocolos e redes do modelo TCP/IP	28
Figura 3.1 - Estrutura macroscópica da Web	41
Figura 4.1 – Sequência de etapas proposta pela metodologia.....	48
Figura 5.1 - Gráfico da porcentagem de páginas da Web Governamental Brasileira em conformidade com os padrões HTML e de acessibilidade de 2010 a 2012.	70
Figura 5.2 - Gráfico com resultados de testes de sincronização de tempo via requisições HTTP para as coletas realizadas de 2010 a 2012.....	71
Figura 5.3 - Gráfico com resultados de testes de sincronização de tempo via requisições NTP para as coletas realizadas em 2011 e 2012.	72
Figura 5.4 - Gráfico contendo a distribuição da porcentagem de <i>sites</i> entre os tipos de servidores registrados em 2010 a 2012.	74
Figura 5.5 - Gráfico da porcentagem de <i>sites</i> localizados no Brasil e no exterior para as coletas executadas no ano de 2010 e 2012	74
Figura 5.6 - Código da função de normalização da matriz de atributos dos dados a serem analisados	83
Figura 5.7 – Código da função de cálculo do PCA (Principal Component Analysis). ..	84
Figura 5.8 – Gráficos indicando os autovalores para cada um dos autovetores encontrados. Neles, a linha vertical indica o atributo limite a ser utilizado. Assim, a cor verde indica os atributos que foram incluídos e, a vermelha indicada aqueles descartados no processo de redução dimensional. Da esquerda para a direita e de cima para baixo, os gráficos correspondem às listas de atributos 1, 2, 3 e 4 indicadas na Seção 5.2.2.	85
Figura 5.9 – Código da função utilizada para executar o algoritmo K-means.	86
Figura 5.10 - Gráficos 2D referentes à lista 1 para a execução, com máximo de 5 grupos, do algoritmo K-means.	89
Figura 5.11 - Gráficos 3D referentes à lista 1 para a execução, com máximo de 5 grupos, do algoritmo K-means.	90
Figura 5.12 - Gráficos 2D referentes à lista 4, com atributos relacionados à estrutura dos <i>sites</i> , para a execução, com máximo de 3 grupos, do algoritmo K-means;.....	91
Figura Apêndice 2.1 – Gráficos 2D referentes à lista 1 para a execução com máximo de 3 grupos. a. Gráfico o algoritmo K-means; b. Gráfico 3D para o algoritmo EM. ..	107

Figura Apêndice 2.2 – Gráficos 3D referentes à lista 1 para a execução com máximo de 3 grupos. a. Gráfico 3D para o algoritmo K-means; b. Gráfico 3D para o algoritmo EM.....	108
Figura Apêndice 2.3 – Gráficos 2D referentes à lista 1 para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.	109
Figura Apêndice 2.4 – Gráficos 3D referentes à lista 1 para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.	110
Figura Apêndice 2.5 – Gráficos 2D referentes à lista 2, com atributos relacionados à validação do código das páginas, para a execução com máximo de 3 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.	111
Figura Apêndice 2.6 – Gráficos 3D referentes à lista 2, com atributos relacionados à validação do código das páginas, para a execução com máximo de 3 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.	112
Figura Apêndice 2.7 – Gráficos 2D referentes à lista 2, com atributos relacionados à validação do código das páginas, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.	113
Figura Apêndice 2.8 – Gráficos 3D referentes à lista 2, com atributos relacionados à validação do código das páginas, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.	114
Figura Apêndice 2.9 – Gráficos 2D referentes à lista 3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, para a execução com máximo de 3 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.	115
Figura Apêndice 2.10 – Gráficos 3D referentes à lista 3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, para a execução com máximo de 3 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.	116
Figura Apêndice 2.11 – Gráficos 2D referentes à lista3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.	117
Figura Apêndice 2.12 – Gráficos 3D referentes à lista3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.	118
Figura Apêndice 2.13 – Gráficos 2D referentes à lista 4, com atributos relacionados à estrutura dos sites, para a execução com máximo de 3 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.....	119
Figura Apêndice 2.14 – Gráficos 3D referentes à lista 4, com atributos relacionados à estrutura dos sites, para a execução com máximo de 3 grupos. a. Gráfico para o	

algoritmo K-means; b. Gráfico para o algoritmo EM..... 120

Figura Apêndice 2.15 – Gráficos 2D referentes à lista 4, com atributos relacionados à estrutura dos *sites*, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM..... 121

Figura Apêndice 2.16 – Gráficos 3D referentes à lista 4, com atributos relacionados à estrutura dos *sites*, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM..... 122

Figura Apêndice 6.1 - Diagrama que representa o ciclo de funcionamento do WIRE
..... 144

LISTA DE TABELAS

Tabela 5.1 - Volume e quantidade de <i>site</i> e páginas por ano de realização da coleta.	68
Tabela 5.2 - Resultados dos testes de suporte ao IPv6 realizados em 2011 e 2012.	73
Tabela 5.3- Atributos adicionados à unidade de pesquisa “ <i>site</i> ” devido à consolidação dos atributos referentes à unidade de pesquisa “páginas”	76
Tabela 5.4 - Descrição dos atributos a serem utilizados em cada um das listas selecionadas.	79
Tabela 5.5 – Relação dos números de <i>sites</i> classificados em cada grupo nas execuções dos algoritmos K-means e EM para os números máximo de 3 e 5 grupo e para cada uma das listas descritas na Seção 5.2.2.	88
Tabela 5.6 – Definição dos grupos criados para cada uma das listas de atributos especificadas na Seção 5.2.2 para a execução do método K-means com limite de 3 grupos.	93
Tabela 5.7 - Classificação e quantidade de <i>sites</i> dos grupos criados para cada uma das listas de atributos especificadas na Seção 5.2.2 para a execução do método K-means com limite de 3 grupos.....	94
Tabela 5.8 – Resultados das análises dos 5 <i>sites</i> mais populares da Internet Governamental Brasileira segundo critérios relacionados à percepção de qualidade do usuário.....	95
Tabela Apêndice 3.1 – Ordem de relevância dos atributos da lista 1 para as classificações resultantes da utilização dos métodos K-means e EM para 3 e 5 grupos.	123
Tabela Apêndice 3.2 – Ordem de relevância dos atributos da lista 2, com atributos relacionados à validação do código das páginas, para as classificações resultantes da utilização dos métodos K-means e EM para 3 e 5 grupos.	124
Tabela Apêndice 3.3 – Ordem de relevância dos atributos da lista 3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, para as classificações resultantes da utilização dos métodos K-means e EM para 3 e 5 grupos.....	125
Tabela Apêndice 3.4 – Ordem de relevância dos atributos da lista 4, com atributos relacionados à estrutura dos <i>sites</i> , para as classificações resultantes da utilização dos métodos K-means e EM para 3 e 5 grupos.	125
Tabela Apêndice 4.1 – Atributos mais relevantes do grupo dos 20 <i>sites</i> do governo brasileiro mais populares segundo o <i>site</i> Alexa.com para a execução do algoritmo K-means para a lista 1 de atributos e com máximo de 3 grupos.	127
Tabela Apêndice 4.2 – Atributos mais relevantes do grupo dos 20 <i>sites</i> do governo	

brasileiro mais populares segundo o *site* Alexa.com para a execução do algoritmo K-means para a lista 2, com atributos relacionados à validação do código das páginas, e com máximo de 3 grupos. 128

Tabela 4.3 Apêndice – Atributos mais relevantes do grupo dos 20 *sites* do governo brasileiro mais populares segundo o *site* Alexa.com para a execução do algoritmo K-means para a lista 3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, e com máximo de 3 grupos. 129

Tabela Apêndice 4.4 – Atributos mais relevantes do grupo dos 20 *sites* do governo brasileiro mais populares segundo o *site* Alexa.com para a execução do algoritmo K-means para a lista 4, com atributos relacionados à estrutura dos *sites*, e com máximo de 3 grupos. 130

Tabela Apêndice 4.5 – Atributos mais relevantes de cada um dos grupos encontrados na execução do algoritmo K-means para a lista 1 de atributos e com máximo de 3 grupos. 131

Tabela Apêndice 4.6 – Atributos mais relevantes de cada um dos grupos encontrados na execução do algoritmo K-means para a lista 2, com atributos relacionados à validação do código das páginas, e com máximo de 3 grupos. 131

Tabela Apêndice 4.7 – Atributos mais relevantes de cada um dos grupos encontrados na execução do algoritmo K-means para a lista 3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, e com máximo de 3 grupos. 132

Tabela Apêndice 4.8 – Atributos mais relevantes de cada um dos grupos encontrados na execução do algoritmo K-means para a lista 4, com atributos relacionados à estrutura dos *sites*, e com máximo de 3 grupos. 132

Tabela Apêndice 5.1 – Conjunto de indicadores analisados pelo projeto TIC Web para a unidade de pesquisa *site*. 134

Tabela Apêndice 5.2 – Conjunto de indicadores analisados pelo projeto TIC Web para a unidade de pesquisa página. 137

Tabela Apêndice 5.3 - Conjunto de atributos consolidados de páginas que se pretende incluir aos dados de *sites*. 140

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
ASES	Avaliador e Simulador de Acessibilidade de Sítios
ATAG	Guia de Acessibilidade para Ferramentas de Desenvolvimento
CEPTRO.br	Centro de Estudos e Pesquisas em Tecnologias de Redes e Operações
CGI.br	Comitê Gestor da Internet no Brasil
CSS	<i>Cascading Style Sheets</i>
ConNeCTOR	<i>Convenient Network Characteristics Testing Organized Routines</i>
DNS	Sistema de Nomes de Domínio
EM	<i>Expectation Maximization</i>
E-MAG	Modelo de Acessibilidade de Governo Eletrônico
E-PING	Padrões de Interoperabilidade de Governo Eletrônico
FA	<i>Factor Analysis</i>
FEA	<i>Feature Extraction Algorithm</i>
FSA	<i>Feature Selection Algorithm</i>
FTP	<i>File Transfer Protocol</i>
FSS	<i>Feature Subsection Selection</i>
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Protocolo Internet
LVF	Las Vegas Filter
MIME	<i>Multipurpose Internet Mail Extensions</i>
Microsoft IIS	<i>Microsoft Internet Information Services</i>
MLE	<i>Maximum Likelihood Estimation</i>
mRMR	<i>min-Redundancy Max-Relevance</i>
NIC.br	Núcleo de Informação e Coordenação do Ponto BR
NTP	<i>Network Time Protocol</i>
OSI	<i>Open Systems Interconnection Model</i>

PCA	<i>Principal Component Analysis</i>
RTT	<i>Round Trip Time</i>
TIC Web	Indicadores para desenvolvimento de Tecnologias de Informação e Comunicação – Web
TLD	Domínio de Topo (<i>Top Level Domain</i>)
UAAG	Guia de Acessibilidade para Ferramentas Utilizadas por Pessoas com Deficiência
URL	Localizador-Padrão de Recursos (<i>Uniform Resource Locator</i>)
UTC	Tempo Universal Coordenado (<i>Coordinated Universal Time</i>)
W3C	Consórcio World Wide Web
WAB	<i>Web Accessibility Barrier</i>
WAI	<i>Web Accessibility Initiative</i>
WCAG	Guia de Acessibilidade para Conteúdos Web
WebApp	Aplicativo Web
WIRE	<i>Web Information Retrieval Environment</i>
XHTML	<i>Extensible HyperText Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	21
1.1	OBJETIVOS	23
1.2	MOTIVAÇÃO E RELEVÂNCIA.....	24
1.3	CONTRIBUIÇÕES ESPERADAS	24
1.4	ORGANIZAÇÃO DA DISSERTAÇÃO	25
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	PADRÕES ABERTOS	27
2.2	PADRÕES WEB.....	28
2.3	PADRÕES DE ACESSIBILIDADE	30
2.4	PADRÕES DE REDE	33
2.5	PADRÕES DE SINCRONIZAÇÃO DE TEMPO	33
2.6	TEMPO DE RESPOSTA	34
2.7	LOCALIZAÇÃO GEOGRÁFICA	35
2.8	APRENDIZADO DE MÁQUINA.....	36
3	TRABALHOS CORRELATOS	39
3.1	PESQUISAS QUANTITATIVAS	39
3.2	PESQUISAS QUALITATIVAS.....	43
4	METODOLOGIA DE ANÁLISE DE CONJUNTOS DE SITES.....	47
4.1	ADAPTAÇÃO DE ATRIBUTOS	49
4.2	SELEÇÃO MANUAL DE ATRIBUTOS.....	50
4.3	NORMALIZAÇÃO DE ATRIBUTOS.....	51
4.4	REDUÇÃO DIMENSIONAL.....	52
4.5	AGRUPAMENTO DE ELEMENTOS	57
4.6	VISUALIZAÇÃO GRÁFICA	62
4.7	ANÁLISE DE INDICADORES	64
4.8	GENERALIZAÇÃO DOS CLASSIFICADORES	65
5	ESTUDO DE CASO	67
5.1	CARACTERÍSTICAS DA WEB GOVERNAMENTAL BRASILEIRA	67
5.1.1	<i>Tamanho da Web</i>	68
5.1.2	<i>Validação de páginas</i>	69
5.1.3	<i>Sincronização dos relógios dos servidores</i>	70
5.1.4	<i>IPv6</i>	72
5.1.5	<i>Tipo de servidor</i>	73
5.2	APLICAÇÃO DA METODOLOGIA	75
5.2.1	<i>Adaptação de atributos</i>	75

5.2.2	<i>Seleção manual de atributos</i>	78
5.2.3	<i>Normalização de atributo</i>	82
5.2.4	<i>Redução dimensional</i>	83
5.2.5	<i>Agrupamento de elementos</i>	85
5.2.6	<i>Visualização gráfica</i>	88
5.2.7	<i>Análise de indicadores</i>	91
5.2.8	<i>Generalização dos classificadores</i>	94
6	CONSIDERAÇÕES FINAIS E CONCLUSÕES	97
7	REFERÊNCIAS BIBLIOGRAFICAS	99
APÊNDICE 1	CÓDIGO DA FUNÇÃO EMCLASS	103
APÊNDICE 2	GRÁFICOS RESULTANTES DA EXECUÇÃO DOS ALGORITMOS K-MEANS E EXPECTATION-MAXIMIZATION	107
APÊNDICE 3	TABELAS INDICATIVAS DA ORDEM DE RELEVÂNCIA DOS ATRIBUTOS	123
APÊNDICE 4	TABELAS EXEMPLIFICANDO A COMPOSIÇÃO DOS GRUPOS	127
APÊNDICE 5	ESPECIFICAÇÃO DE INDICADORES	133
5.1	ESCOPO DA ANÁLISE	133
5.2	INDICADORES DE SITES	134
5.3	INDICADORES DE PÁGINAS	137
APÊNDICE 6	FERRAMENTAS DE EXTRAÇÃO DE ATRIBUTOS	143
6.1	WEBCRAWLER	143
6.2	FERRAMENTA DE ANÁLISE	145
6.2.1	<i>Validação de páginas Web</i>	146
6.2.2	<i>Avaliação de sincronização do relógio dos servidores</i>	146
6.2.3	<i>Testes Relacionados a Servidores Web</i>	147

1 INTRODUÇÃO

A importância da Internet para toda a sociedade é indiscutível. Ela serve de plataforma para inúmeras aplicações pelo mundo e possibilita a criação de novos e inovadores métodos de comunicação. Por sua vez, a Web pode ser entendida como a parte Internet acessível através de *browsers*, programas responsáveis por tornar certos tipos de linguagens e códigos em conteúdos visualizáveis por usuários. Além disso, é ela a tecnologia responsável pela popularização e crescimento da Internet.

Levando isso em conta, o Comitê Gestor da Internet Brasileira (CGI.br) e o Núcleo de Informação e Coordenação do Ponto BR (NIC.br), através do escritório do Consórcio World Wide Web (W3C) Brasil e do Centro de Estudos e Pesquisas em Tecnologias de Redes e Operações (CEPTRO.br), iniciaram um projeto chamado “Indicadores para desenvolvimento de Tecnologias de Informação e Comunicação – Web” (TIC Web) com o objetivo de melhor entender certas características tecnológicas da Web Brasileira. Como exemplo de objeto de estudo, cita-se a verificação da adoção de boas práticas e de padrões abertos nos sítios brasileiros. Mais especificamente, são analisados domínios Web segundo a adoção de determinados padrões abertos relacionados à percepção de qualidade dos *sítes* por seus usuários, agrupando-se, para tanto, uma série de indicadores com os quais se defina e diferencie domínios Web.

Apesar de outros trabalhos já terem discutido diversos dos pontos aqui abordados, como mostrará o Capítulo 3, a comparação dos resultados é difícil devido a tais análises focarem-se na caracterização estrutural e amostral de domínios Web, enquanto o TIC Web e, por consequência, o presente trabalho tentarem realizar análises sobre dados censitários e periódicos obtidos não apenas sobre a camada de aplicação, mas também de outras camadas da rede.

Como forma de avaliar as ferramentas e hipóteses a serem utilizadas no projeto, foram realizadas análises sobre a Web Governamental Brasileira, constituída por *sítes* sob o domínio *.gov.br*. Os resultados alcançados pelo TIC Web foram publicados na conferência Webmedia’12 em artigo escrito por Ganzeli, Moreiras, & Bressan (2012). A opção de tal domínio Web como ponto de partida foi justificada por suas características singulares. Apesar de ser um domínio com poucos *sítes*, esses, são controlados por entidades governamentais e, portanto,

estão sujeitos a diversos regulamentos e normas. Em particular, lhes é exigida compatibilidade com uma série de padrões de interoperabilidade descritos pelo documento “Padrões de Interoperabilidade de Governo Eletrônico” (E-PING) criado pelo Ministério do planejamento (2012) que incluem o “Modelo de Acessibilidade de Governo Eletrônico” (E-MAG) desenvolvido pelo Ministério do Planejamento & Secretaria de Logística e Tecnologia da Informação (2011). Esse modelo consiste de um guia de acessibilidade eletrônica criado em 2007 como consequência das Leis Federais 10.098 e 10.048 e do Decreto 5.296, os quais estabeleceram o período de um ano para que todos os portais do governo estivessem aderentes a padrões de acessibilidade. O conjunto dessas ações governamentais mostra motivação para tornar a Web Governamental um lugar mais acessível, e reforça a validade do projeto TIC Web como um meio de expor *sites* não conformes com tais padrões.

O presente trabalho é uma extensão de tal projeto, e se foca na definição e aplicação de metodologia baseada em técnicas de aprendizado de máquina para a automatização das análises dos dados extraídos, visando praticidade na execução da categorização de *sites* Web segundo critérios relacionados à qualidade percebida por seus usuários. Essa proposta foi motivada pela da necessidade intrínseca que o TIC Web possuía de aprimorar seu processo de análise de dados, até então realizado primordialmente de forma artesanal, sendo que apenas a comparação entre domínios e a verificação tendências globais eram possíveis. Com a adição de ferramentas mais inteligentes ao processo, novas análises podem ser feitas e resultados mais aprofundados e úteis podem ser adquiridos.

Um exemplo é a generalização dos resultados obtidos da classificação dos elementos da base de dados para a realização de avaliações de outros *sites* Web. Isso corrobora com uma tendência crescente da avaliação de qualidade e desempenho que *sites*, principalmente comerciais, vêm apresentando nos últimos anos. Ou seja, pretende-se criar ferramental, tecnológico e metodológico, para que se possa melhor extrair informações dos dados obtidos pelo TIC Web e utiliza-los em prol da comunidade.

1.1 Objetivos

O presente trabalho propõe, como principal objetivo, o desenvolvimento de uma metodologia para a realização automatizada de análises indicativas da qualidade de Web *sites*. Seu diferencial está na utilização de técnicas de aprendizado de máquina para a análise de dados de diferentes grupos de Web *sites* e posterior confecção classificadores capazes de tal avaliação.

Para poder ser aplicada sob diversos cenários, essa metodologia precisava ser descrita de forma clara e cada um de seus aspectos, tanto aqueles relacionados com a avaliação de *sites* quanto os do próprio processamento dos dados, deveriam ser justificados. Assim, os modelos criados podem, ser utilizados para a avaliação de novos *sites* em diversos cenários, como regulações governamentais ou negociações empresariais que compreendam desenvolvimento de aplicações Web.

Prende-se que este trabalho possa ser por pessoas ou entidades interessadas em estudos de Web *sites* para criar classificadores de qualidade para seus *sites* ou para *sites* de terceiros. Além disso, desenvolvedores Web poderiam utilizar os serviços de classificação para medir e melhorar a qualidade de seus próprios *sites*.

Alguns objetivos específicos pretendidos com este trabalho foram:

- O estudo de características que definam a qualidade ou falta de qualidade de um *site*;
- O estudo de técnicas de aprendizado de máquina adequados à análise e caracterização de *sites*;
- O estudo de metodologias de análise semelhantes, nas quais a presente proposta poderia se respaldar;
- O desenvolvimento de ferramental para a aplicação da metodologia proposta;
- A realização de um estudo de caso, como base nos dados obtidos do projeto TIC Web, como forma de validar não apenas a metodologia descrita, mas também a seleção dos algoritmos realizada.

1.2 Motivação e relevância

A motivação do projeto TIC Web vem da necessidade da obtenção de dados estatísticos sobre algumas das principais tecnologias abertas recomendadas pelo CGI.br e pelo escritório do W3C Brasil. Por tecnologias abertas entende-se aquelas baseadas em padrões abertos que, por sua vez, são especificados por comitês com participação aberta a interessados e que possuem a garantia de que sua implementação não acarretará em custos por royalties.

O CGI.br, desde de sua criação em 1995, tem como propósito a coordenação e integração de iniciativas de serviços Internet no Brasil, assim, a visualização do situação da Web Brasileira é importante por possibilitar a verificação da difusão de suas recomendações e, conseqüentemente, a identificação daquelas que necessitam mais dedicação para prosperar.

Quanto à motivação do escritório do W3C Brasil, cita-se, além suas metas gerais de melhoria da interoperabilidade na Web, seu comprometimento em torná-la acessível para uma vasta quantidade pessoas e dispositivos.

O presente trabalho compartilha tais motivações e acrescenta a necessidade da criação de melhores estratégias, metodologias e ferramentais que viabilizem a realização de estudos aprofundados sobre as informações adquiridas. Isso é importante pois, no estágio atual do TIC Web, consegue-se apenas analisar as características de domínios de forma agregada e, conseqüentemente, perde-se a capacidade da individualização dos resultados, útil para a classificação de *sites* e determinação de agrupamentos (*clusters*) de *sites* com características semelhantes.

Destaca-se assim, como principais melhorias a serem atingidas, um melhor entendimento das características de domínios, a individualização das análise, a definição de metodologia para a análise grupos de *sites* e a possibilidade da aplicação dos conhecimentos obtidos para a classificação de *sites* externos aos domínios sob análise.

1.3 Contribuições esperadas

Cita-se como principal contribuição do presente trabalho a definição de metodologia para a análise de domínios de *sites* e posterior classificação de *sites*

segundo sua qualidade. Essa definição precisava ser feita de forma clara para que pudesse ser replicada e comparada por outros no futuro. Outra contribuição consiste da extensão do TIC Web, com o aprimoramento do processo de análise desse projeto e com a possibilidade do aumento de sua repercussão com a classificação de *sítes* com bases nos resultados obtidos de suas coletas. Além disso, menciona-se também como contribuições a seleção e explicação de algoritmos como de forma validar a metodologia e facilitar sua aplicação à novos conjuntos de dados.

1.4 Organização da dissertação

Nesta dissertação será apresentado, no Capítulo 2, um conjunto de conceitos essenciais com os quais este trabalho está relacionado. No capítulo seguinte, está disposta a revisão bibliográfica com a apresentação de trabalhos relevantes nas áreas de pesquisas quantitativas e qualitativas para a caracterização de *sítes* Web.

No Capítulo 4, a metodologia aqui proposta é definida. Ao todo foram desenvolvidas oito etapas de análise e, para cada uma, as técnicas utilizadas são detalhadas. No capítulo 5 seguinte está um estudo de caso com a aplicação da metodologia. Ele está dividido em duas partes principais, sendo que a primeira apresenta o conjunto de dados resultante do trabalho TIC Web, como forma de melhor contextualizar os resultados obtidos no presente projeto. A segunda parte desse quinto capítulo apresenta a execução da metodologia do quarto capítulo. Para melhorar a fluidez do texto, a descrição dos indicadores considerados e das ferramentas utilizadas pelo TIC Web para a aquisição dos dados, foram incluídas respectivamente nos Apêndices 5 e 6.

Por último, o Capítulo 6 destina-se à discussão da aplicabilidade do trabalho bem como dos resultados obtidos.

A Figura 1.1 sumariza a estrutura do presente trabalho.

Figura 1.1 – Mapa mental da estrutura do presente trabalho

2 FUNDAMENTAÇÃO TEÓRICA

Como forma de justificar sua utilização dos indicadores do projeto TIC Web, neste capítulo são apresentados os principais conceitos e padrões à eles relacionados. Começando pela definição de padrões abertos, serão abordados também padrões Web, de acessibilidade e de rede, além de definições mais técnicas, como o tempo de resposta a requisições Web. Por fim, conceitos básicos de aprendizado de máquina são tratados devido à sua relevância na descrição da metodologia proposta.

2.1 Padrões abertos

Segundo Tanenbaum (2003), a Internet pode ser definida como uma vasta coleção de redes que utilizam protocolos comuns para prover determinados serviços. Isso só é possível pois tais protocolos de comunicação foram previamente acordados e padronizados, permitindo que cada uma das redes de computadores existentes no mundo consiga conversar com as outras. Padrões abertos são aqueles especificados por comitês com partição aberta a qualquer parte interessada e, além disso, eles garantem que seu emprego não exige custos extras por royalties. Como definido pelo W3C em artigo escrito por Dardailler (2007), padrões abertos proporcionam transparência, aumento de relevância, abertura, imparcialidade, consenso, disponibilidade e facilidade de manutenção.

Os protocolos utilizados por uma rede podem ser categorizados em camadas Internet de acordo com suas funções. Para reduzir sua complexidade de desenvolvimento, tais camadas são organizadas em forma de pilha. Um modelo de referência que explica o funcionamento dessa arquitetura é chamado de modelo “*Open Systems Interconnection Model*” (OSI) descrito em Recommendation (1994). Ele define sete camadas e as funções que por elas devem ser desempenhadas, deixando em aberto os protocolos necessários para implementá-las. Assim, tanto protocolos abertos quanto proprietários podem ser escolhidos. Apesar das implementações desse modelo praticamente não serem mais utilizados, ele ainda pode ser generalizado para o entendimento de outras arquiteturas. Na Internet, foi implantado um modelo similar, também organizado em camadas, chamado TCP/IP,

que define as seguintes camadas não totalmente correspondentes às do modelo anterior: aplicação, transporte, rede e enlace. A Figura 2.1 contém um diagrama dessa última arquitetura e exemplifica alguns dos protocolos que podem compor uma comunicação.

Figura 2.1 - Protocolos e redes do modelo TCP/IP

Fonte: Tanenbaum (2003)

O projeto TIC Web se foca na avaliação da utilização de padrões contemporâneos e abertos nas camadas de rede e aplicação entre os *sítes* da Web brasileira. O artigo de Chen, Hong, & Shen (2005) discute a importância da utilização do padrão HTML na construção de aplicações Web como forma de garantir unicidade da apresentação do documento em diferentes plataformas. O mesmo argumento é válido para os outros padrões e protocolos abertos. De forma geral, sua aplicação possibilita a comunicação entre os diversos sistemas presentes na Internet, provendo também os outros benefícios anteriormente citados.

2.2 Padrões Web

No modelo TCP/IP, é a camada de aplicação que engloba os diversos tipos de aplicações distribuídas que são disponibilizadas na Internet. Alguns exemplos são os sistemas de e-mail e, principalmente, a Web. Tamaña gama de aplicações só foi atingida devido à variedade de protocolos suportados por essa camada. O *Hypertext*

Transfer Protocol (HTTP), o *Hypertext Transfer Protocol Secure* (HTTPS) e o *File Transfer Protocol* (FTP), são alguns exemplos.

A Web é provavelmente a aplicação Internet mais conhecida e utilizada pela população. De certo ponto de vista, ela é um conjunto de documentos interligados descritos por estruturas padronizadas que permitam sua apresentação em browsers. Tais estruturas, para existir, precisam de entidades responsáveis por deliberar sobre sua padronização. No caso dos documentos Web tal entidade é o W3C, o qual é responsável por diversos dos padrões Web abertos, e.g., *Hypertext Markup Language* (HTML) e *Cascading Style Sheets* (CSS). Mais especificamente, o HTML possibilita:

- Publicação de documentos on-line compostos por cabeçalhos, textos, tabelas, listas, imagens, dentre outros;
- Acesso on-line a informações via hiperlinks;
- Transações entre serviços remotos para busca de informações, reservas, compra produtos, dentre outros; e,
- Inclusão, nos documentos, de objetos multimídia como planilhas, vídeos, músicas ou outras aplicações.

Salienta-se, no entanto, que caso padrões Web não existissem, a visualização do conteúdo on-line seria muito mais complicada, já que cada documento poderia ter seu próprio conjunto de marcações e ser dependente de aplicações específicas para visualizá-los. Ou seja, o emprego de padrões abertos na Web, e na Internet como um todo, consiste numa maneira eficaz de se preservar a consistência do funcionamento das aplicações independentemente da plataforma sobre a qual elas estão sendo executadas, característica cada vez mais importante dada a crescente quantidade de plataformas e dispositivos.

Apesar da popularidade do HTML, aplicar plenamente suas regras não é tarefa fácil, como citado no artigo de Bauer & Scharl (2000), que discorre sobre as dificuldades na aplicação de processos de avaliação manual da qualidade de Web sites organizacionais. Documentos HTML costumam ser relativamente grandes e ter conteúdo modificado frequentemente tanto em sua criação quanto manutenção, o que acarreta em constantes de erros validação. Levando isso em consideração, o W3C desenvolveu e mantém uma ferramenta de auxílio a desenvolvedores,

chamada *W3C Markup Validator*¹, que expõe os erros de codificação presentes nas páginas. Outras ferramentas como o *HTML Tidy*² além da validação realizam também a correção dos erros de forma automática, sempre que possível.

Assim, a conformidade com padrões Web traz diversos benefícios ao ecossistema Web. Porém, apesar de sua importância, não existe muita preocupação em aplicá-los na prática, como visto no trabalho de Ofuonye, Beatty, Dick, & Miller (2010). Com a verificação da aderência a tais padrões pelo presente projeto, consegue-se manter registro da evolução de sua adoção e verificar a eficácia dos esforços realizados principalmente pelo W3C de popularização desses padrões.

2.3 Padrões de acessibilidade

O termo acessibilidade possui múltiplos significados dependendo do contexto em que é utilizado. Segundo definição oficial do W3C por Henry (2006), ele está relacionado à capacidade de pessoas com diferentes graus de aptidão, ou inaptidão, acessar determinados conteúdos. As recomendações acerca do desenvolvimento de *sites* acessíveis se baseiam nos seguintes princípios descritos por Caldwell, Cooper, Reid, & Vanderheiden (2006):

- A interface de usuário e o conteúdo publicado devem ser apresentados de maneira que todos aqueles que os acessem possam percebê-los;
- A interface de usuário deve ser operável, ou seja, ela não deve conter formas de interação que seus usuários não sejam capazes de executar;
- O conteúdo transmitido deve ser entendível; e,
- O conteúdo transmitido deve ser suficientemente robusto para que possa ser interpretado por uma ampla variedade de dispositivos e plataformas, incluindo sistemas com tecnologias de apoio à deficientes.

Levando esses princípios em consideração, um grupo de trabalho pertencente ao W3C, chamado *Web Accessibility Initiative* (WAI), criou 3 guias para auxiliar no desenvolvimento de aplicações acessíveis: Guia de Acessibilidade para Conteúdos

¹ Disponível em: <http://validator.w3.org/>. Acessado em: 18 de abril de 2014

² Disponível em: <http://tidy.sourceforge.net/>. Acessado em: 18 de abril de 2014

Web (WCAG); Guia de Acessibilidade para Ferramentas de Desenvolvimento (ATAG); e, Guia de Acessibilidade para Ferramentas Utilizadas por Pessoas com Deficiência (UAAG).

O WCAG trata principalmente dos pontos de verificação que as páginas Web devem cumprir para que possam ser adequadamente acessados por pessoas com deficiências. Nesse sentido, ele define três níveis de prioridade para cada desses pontos:

- **Prioridade 1:** Compreende os pontos de verificação mais essenciais para a garantia da acessibilidade básica do *site*. Os pontos de prioridade um são aqueles que os criadores de conteúdo Web devem satisfazer inteiramente. Se não os satisfizerem, um ou mais grupos de usuários ficarão impossibilitados de acessar as informações contidas no documento. Assim, sua presença é requisito básico para que determinados grupos de pessoas sejam capazes de acessar documentos disponíveis na Web. Exemplo: “prover texto equivalente a cada elemento não textual”;
- **Prioridade 2:** Compreende pontos de verificação que os criadores de conteúdo deveriam satisfazer. Se não os satisfizerem, um ou mais grupos de usuários terão dificuldade para acessar as informações contidas no documento. A presença desses pontos promove, então, a remoção de barreiras que atrapalhem o acesso a páginas Web. Exemplo: “Utilizar de forma adequada arquivos de estrutura e folhas de estilo”; e,
- **Prioridade 3:** Compreende os pontos de verificação que os criadores de conteúdo na Web podem satisfazer. Se não os satisfizerem, um ou mais grupos de usuários poderão ter um pouco de dificuldade para acessar informações contidas nas páginas. A presença de pontos de verificação de prioridade 3 melhora o acesso às páginas Web. Exemplo: “Especificar por extenso acrônimos na primeira vez que aparecem nos textos”.

A partir da análise de todos os pontos de verificação descritos no WCAG, uma determinada página Web pode ser classificada como não acessível ou acessível nível “A”, “AA” ou “AAA”. Para ser considerada acessível de nível “A”, ela deve cumprir todos os pontos de verificação de prioridade 1. Para nível “AA”, ela deve cumprir todos dos pontos das prioridades 1 e 2. Finalmente, para o nível “AAA”, ela

deve cumprir todos os pontos de verificação. Quando uma página não atende a nenhum desses três casos, ele é considerado não acessível. Ou seja, um ou mais grupos de usuários ficam impossibilitados de acessar suas informações.

Apesar de o WCAG ser um dos guias de acessibilidade mais referenciados, o Governo brasileiro concluiu, em 2007, o E-MAG com base nesse primeiro. Em teoria, ele complementa o guia anterior com a adição de especificidades das instituições governamentais e das características de seus públicos alvos. No entanto, o trabalho de Bach, Ferreira, Silveira, & Nunes (2009) comprova a compatibilidade entre eles.

ATAG é o guia que define como ferramentas de autoria, utilizadas por desenvolvedores para produzir conteúdo Web, deveriam ajudar no desenvolvimento de conteúdo acessível. Ele explica, também, os componentes que as ferramentas de autoria deveriam conter para prover acessibilidade aos próprios desenvolvedores.

Por fim, o UAAG mostra como projetar as aplicações normalmente utilizadas por pessoas com deficiências como, por exemplo, browsers e tocadores de mídia.

Apesar de todas essas regras serem voltadas para um pequeno nicho da população, esse não é o único beneficiário. Em geral, tais regras ajudam na construção de *sites* bem estruturados que podem ser facilmente portados para novas tecnologias. Por exemplo, um aplicativo Web (WebApp) que siga convenções de estilo e *layout* modernos pode facilmente ser acessado por *browsers* de computadores e de celulares sem perda na qualidade do conteúdo. Além disso, a melhor estruturação do *site* facilita sua utilização por grupos de pessoas com baixo nível de instrução tecnológica, que compartilham de algumas das dificuldades de pessoas com deficiência. O artigo de Hull (2004) discorre sobre tais questões.

A partir de uma rápida busca na Web, diversas ferramentas de validação de acessibilidade podem ser encontradas. O WAI fornece uma lista³ contendo algumas delas. No entanto, dado que algumas das regras recomendadas são bastante subjetivas, como o exemplo fornecido para o nível 'AAA', e não podem ser bem avaliadas por esse tipo de ferramenta, a presença de pessoas com deficiências nas etapas de teste das aplicações é imprescindível.

³ Disponível em: <http://www.w3.org/WAI/ER/tools/complete>. Acessado em: 18 de abril de 2014

2.4 Padrões de rede

O Protocolo Internet (IP) é o protocolo adotado pela camada de rede na pilha TCP/IP. Atualmente ele está sofrendo processo de transição da versão 4 para a 6, conhecida por IPv6 que foi descrita na RFC 2460 escrita por Deering & Hinden (1998). Essa versão é, na verdade, um novo protocolo e começou a ser desenvolvida nos anos 90 como solução para a exaustão do espaço de endereçamento do IPv4, pois, com a popularização da Internet e aumento do número de dispositivos conectados, os 4,29 bilhões de endereços disponíveis para mapear interfaces de rede começaram a se mostrar insuficientes.

Com o IPv6, ao invés da utilização de um campo de apenas 32 bits para identificar as máquinas na rede, foram designados 128 bits para tal função. Esse é um dos principais motivos que torna tais protocolos incompatíveis e que, conseqüentemente, dificulta o processo de transição. Contudo, para que a Internet possa continuar crescendo, essa mudança é inevitável. A não adoção desse protocolo pode fazer com que um crescente número de usuários, conectados apenas via IPv6, não consiga acessar os serviços disponibilizados exclusivamente sob o protocolo antigo.

Além dessa motivação, o IPv6 também agrega algumas vantagens à camada de rede, incluindo melhorias em desempenho, devido ao fato de os campos do cabeçalho serem fixos; e mobilidade, que passou a integrar o conjunto básico de funcionalidades do protocolo.

2.5 Padrões de sincronização de tempo

Outro protocolo da camada de aplicação que merece destaque é o *Network Time Protocol* (NTP), apresentado por Rybaczky (2005). Apesar da palavra “protocolo” em seu nome, ele pode ser descrito como sendo 3 coisas distintas, um algoritmo de sincronização de tempo, um protocolo de comunicação e uma aplicação responsável por deixar o relógio de um dispositivo sincronizado com o horário *Coordinated Universal Time* (UTC).

Suas vantagens quando comparado a outros protocolos de sincronização incluem portabilidade, baixo consumo de energia e pequeno *offset* (diferença de tempo com relação ao horário oficial).

O tratamento de incidentes de segurança e de crimes cibernéticos, é o ponto chave que reforça a necessidade dos mecanismos de sincronização de tempo em dispositivos. O rastreamento de invasões em análises forenses torna-se inviável, caso o relógio dos dispositivos envolvidos estivessem defasados entre si. Além disso, a sincronização de tempo é fundamental também na coordenação de eventos em aplicações distribuídas e em diversas outras, como gerenciadores de e-mails e mensagens, que muitas vezes dependem da precisão do relógio interno das máquinas.

2.6 Tempo de resposta

O tempo de resposta é mais um importante indicador na avaliação da qualidade de serviço provida por um *site*. Os artigos de Palmer (2002); Rose & Straub (2001); Savoia (2001) discorrem sobre tal importância como um dos componentes para uma boa percepção da qualidade de um serviço por parte de seus usuários. Em particular, o artigo de Savoia (2001) propõe a existência de quatro regras gerais sobre o desempenho de determinado *site*:

- Lei da aderência (*The Law of Stickiness*) – um usuário tende a continuar utilizando um mesmo *site*, porém ele não é leal. Quando esse *site* apresentar problemas, o usuário pode facilmente mudar para um serviço concorrente e se manter nele;
- Lei da Perspectiva do usuário (*The Law of User Perspective*) – A performance de um Web *site* deve ser medida a partir do ponto de vista do usuário do *site* e não da rede do provedor de conteúdo;
- Lei da Responsabilidade (*The Law of Responsibility*) – Apesar de existirem diversos fatores que influenciam o desempenho de um *site*, a culpa por uma experiência ruim é sempre do *site*.
- Lei da Expectativa (*The Law of Expectations*) – A satisfação de um usuário com um determinado serviço depende de sua expectativa para com esse o serviço e isso é diretamente condicionado as suas experiências.

Disso, é interessante notar que não se coloca uma meta padrão que um *site* deva atingir para que sua performance seja considerada boa. Porém, é importante que ele esteja sempre melhorando e que consiga, até certo ponto, acompanhar ou minimamente superar os serviços concorrentes.

Esse mesmo artigo cita a seguinte formula para quantificar os principais fatores que influenciam o tempo de download de uma página:

$$\begin{aligned} & \frac{\text{Tamanho da Página}}{\text{Largura de Banda Mínima}} + (RTT \times \text{Ciclos}) \\ & + \text{Tempo de processamento do Servidor} \\ & + \text{Tempo de processamento do Cliente} \\ & = \text{Tempo de Resposta da Página} \end{aligned} \quad 2-1$$

Conforme será descrito na Seção 6.2, foi desenvolvida para o projeto TIC Web uma ferramenta de análise de *sites* que possui entre suas funcionalidades a análise do tempo de resposta dos *sites*. Para realizar a medição, essa ferramenta executa uma requisição HTTP HEAD ao servidor. Segundo a equação 2-1 apresentada, esse teste almeja a verificação do *Round Trip Time* (RTT) somado ao tempo de processamento do Servidor. Isso acontece pois, como na requisição do tipo HEAD apenas o cabeçalho da página é enviado, o atraso referente ao tamanho da página e ao número de ciclos de transmissão podem ser ignorados. O tamanho das páginas é considerado em um indicador isolado. Contudo, a largura de banda, o número de ciclos e o tempo de processamento do cliente não puderam ser medidos pelas ferramentas do projeto.

2.7 Localização geográfica

Conforme apresentado na seção anterior, o tempo de resposta é um fator relevante para a melhora da experiência de uso de um determinado *site* e um dos elementos que mais influencia nessa latência é a distância entre o servidor de determinada aplicação e seu cliente. Simplificadamente, para que exista comunicação entre um usuário e um servidor, é necessária a troca de mensagens entre eles e, dada a existência de um limite físico da velocidade com a qual tais

mensagens podem ser trafegadas, quanto maior a separação entre as partes, maior será a latência da troca.

Apesar de a distância entre algumas cidades ou até mesmo estados não representar um acréscimo significativo no tempo de trânsito de uma requisição, a distância entre continentes de fato representa. Até mesmo uma conexão entre São Paulo e a Florida, que estão diretamente interligados através de fibras ópticas ultramarinas, adiciona em torno de 150 milissegundos no tempo de resposta, fazendo com que qualquer atraso extra já seja perceptível. Por isso, é importante que os serviços voltados ao público brasileiro estejam alocados no Brasil.

Quando se avalia a Web Governamental, adicionam-se também requisitos segurança e privacidade que os sistemas devem atender e nasce a dúvida de até qual ponto se pode confiar os dados relativos à população brasileira a entidades estrangeiras.

2.8 Aprendizado de máquina

Mudando o foco da discussão dos elementos relacionados à qualidade dos *sites*, esta seção aborda brevemente alguns princípios de aprendizado de máquina importantes para a definição da metodologia de análise proposta. Segundo Mitchell (1997), aprendizado de máquina pode ser definido como a ciência que estuda a construção de computadores capazes de melhorar a forma como desempenham determinada tarefa a partir do ganho de experiência. Nesse sentido, ele define um problema de aprendizado de máquina como sendo aquele em que determinado programa aprende com determinada experiência E , a respeito de determinada tarefa T , se seu desempenho, P , na realização de T melhora devido a E .

Um exemplo prático é um sistema antisspam, que, com base na observação da marcação de mensagens como *spam* por seus usuários, aprende a filtrar as mensagens recebidas.

Algoritmos de aprendizado de máquina podem ser divididos em diversas categorias, sendo que duas das principais são o aprendizado supervisionado e o não supervisionado. Na primeira, os algoritmos são ensinados, por meio de exemplos, a realizar determinada função, enquanto que na segunda, eles devem aprendê-las por si próprios.

Ou seja, em algoritmos de aprendizado de máquina supervisionado, as bases de dados utilizadas para o treinamento contêm informações prévias sobre os resultados esperados e, com isso, o algoritmo aprende a executar classificação similar. Exemplos desse tipo são: algoritmos de regressão, os quais a partir de um conjunto de pontos em um gráfico tentam prever a relação entre duas ou mais variáveis; e, algoritmos de classificação, que se focam em decidir à qual classe determinado ponto pertence, dado um conjunto de elementos previamente classificados.

Em algoritmos não supervisionados, tal classificação prévia dos dados não existe. Cabe aos próprios algoritmos a decisão sobre quais elementos possuem maior afinidade e quais os grupos, também chamados de aglomerados, devem ser formados para separação dos dados. Alguns exemplos de problemas aos quais esse tipo de algoritmo pode ser aplicado são: a agrupamento de pessoas conhecidas em redes sociais; e, a segmentação de consumidores na avaliação de vendas de determinado mercado.

Para a realização das análises propostas neste projeto, entende-se que a aplicação de algoritmos de aprendizado de máquina não supervisionados seja mais adequada, uma vez que a base de dados em questão não possui qualquer classificação prévia dos *sites*. Além disso, a descoberta de agrupamentos torna-se viável, permitindo segmentação dos elementos com base em suas características de qualidade. Contudo, como será visto na descrição da metodologia no Capítulo 4, a utilização conjunta de técnicas de supervisionadas pode auxiliar a pós análise dos resultados e grupos encontrados no processo principal.

3 TRABALHOS CORRELATOS

Trabalhos que analisam as características e o funcionamento de *sítes* e domínios Web não são recentes. Os trabalhos de Bailey, Craswell, & Hawking (2003); Castillo, Starosta, & Sydow (2007); Castillo (2005); Edward, Lavoie, & Patrick (2001); Lawrence & Giles (2000); Modesto, Junior, & Ziviani (2005); Tolosa, Bordignon, Baeza Yates, & Castillo (2007) são alguns exemplos. Os principais motivadores dessas pesquisas são: avaliação de estudos arquiteturas de softwares de coletas, estudo de comportamentos sociais e tecnológicos, estudos linguísticos, entre outros. Contudo, destaca-se, o entendimento da Web e o aperfeiçoamento de ferramentas utilizadas por buscadores Web, como o Google ou o Bing, conforme mostrado nos artigos de Edward et al. (2001); Lawrence & Giles (2000).

Seguindo os objetivos discutidos no capítulo anterior, de analisar a Web segundo a adoção de certos padrões Internet abertos relacionados à percepção de qualidade dos *sítes* Web, os indicadores a serem estudados neste trabalho podem ser divididos em duas subcategorias. A primeira engloba indicadores quantitativos, como a contagem de páginas e *sítes* já, a segunda possui, cunho qualitativo e, engloba indicadores como o suporte ao protocolo IPv6, e a localização geográfica dos servidores. Como essas categorias não são habitualmente estudadas em conjunto, os trabalhos referenciados foram divididos entre as seções 3.1 e 3.2.

3.1 Pesquisas Quantitativas

Alguns dos trabalhos que serviram de base para a escolha dos indicadores quantitativos do presente trabalho foram: Castillo et al. (2007); Castillo (2005); Lawrence & Giles (2000); Mendes, Mosley, & Counsell (2002); Modesto et al. (2005); Tolosa et al. (2007). Em particular, os trabalhos de Modesto et al. (2005); Tolosa et al. (2007) formaram as principais referências para o desenvolvimento da metodologia deste projeto. Tais pesquisas mostraram a possibilidade da utilização de buscadores Web para observar e diferenciar domínios Web. Porém, devido às limitações de tempo, banda Internet e espaço de armazenamento, ambas se constituíram de pesquisas amostrais e analisaram apenas uma pequena parcela dos *sítes* sob os domínios Web em estudo.

O trabalho de Lawrence & Giles (2000) apresenta uma distinta metodologia para a amostragem de *sites*. Ao invés de buscar uma semente relevante de *sites* para iniciar o processo de coleta, ele propõe uma busca de servidores Web através da realização de requisições HTTP a endereços IP selecionados de forma aleatória. A técnica se mostrou bastante conveniente por permitir uma análise amostral da Web como um todo. Porém, com a evolução da Web, a utilização de técnica similar para realizar tal estimativa torna-se questionável, uma vez que se tornou comum o caso de um mesmo servidor responder por múltiplos *sites* não necessariamente acessíveis conhecendo-se unicamente o endereço IP desse servidor. Ainda, o início da utilização do IPv6 torna inviável pesquisa pelo novo espectro de endereços, já que a porcentagem de endereços com serviços Web, em relação ao total de números do espectro, tende a ser significativamente menor do que no IPv4.

O estudo da Web brasileira realizado por Modesto et al. (2005) apresentou também os indicadores mais relevantes para pesquisas quantitativas, além dos recursos básicos para se realizar pesquisas desse porte. Os artigos de Castillo et al. (2007); Modesto et al. (2005); Tolosa et al. (2007) possuem características bastante semelhantes, já que todos eles fazem uso da ferramenta *Web Information Retrieval Environment* (WIRE) para analisar o conjunto de *sites* sob a Web de determinado país. Os principais indicadores neles apresentados são:

- Tamanho da Web em número de *sites* e páginas;
- Quantidade de páginas únicas, dinâmicas e estáticas;
- Quantidade de páginas por *Multipurpose Internet Mail Extensions* (MIME) *Type*, conjunto de identificadores do tipo de determinado conteúdo;
- Idade dos documentos;
- Quantidade de páginas por código de resposta HTTP;
- Quantidade de links externos ao escopo de análise;
- Porcentagem de páginas pertencente à cada uma das estruturas macroscópicas da Web, definidas para diferenciar tipos de *sites* de acordo com seus links, conforme diagrama apresentado na Figura 3.1;

- Quantidade de páginas por linguagem do código fonte; e,
- Quantidade de links encontrados por tipo.

Figura 3.1 - Estrutura macroscópica da Web

Fonte: Modesto et al. (2005)

Apesar de esses três artigos apresentarem interessantes resultados para seus respectivos domínios analisados, a escolha da profundidade máxima das páginas a serem coletadas e a escolha das sementes iniciais não são bem fundamentadas e, em geral, sua capacidade de representar a Web de um determinado país pode ser questionada.

O artigo de Mendes et al. (2002) apresenta uma detalhada discussão de atributos quantitativos que poderiam ser utilizados para caracterizar e uma aplicação Web. Segundo classificação nele apresentada, os indicadores que serão utilizados para o presente estudo, mostrados no Apêndice 5, se enquadram na categoria de métricas de comprimento. Observa-se, no entanto, que caso as análises que se deseja realizar não apresentem dados satisfatórios, a adição dos outros atributos citados seria interessante. Os atributos mais relevantes apresentados compreendem:

- Número de páginas;
- Número de arquivos de mídia;
- Número de programas;
- Tamanho total da página;
- Tamanho total dos arquivos de mídia;
- Número de linhas de códigos embarcados;
- Número de arquivos de mídia reutilizados;
- Número de programas reutilizados;
- Tamanho total dos arquivos de mídia reutilizados;
- Tamanho total dos programas reutilizados;
- Número total de links internos;
- Número total de links internos por página; e,
- Quantidade de diferentes tipos de mídia por página.

Além da discussão sobre os melhores atributos a serem utilizados, esse artigo também propõe uma tentativa de adaptar uma curva estatística, utilizando métodos de regressão linear e múltipla, para a previsão do tamanho das aplicações web. Porém, a análise não obteve sucesso no estudo de caso realizado.

Como a nomenclatura sugere, os indicadores quantitativos são úteis para que se possa ter conhecimento das dimensões do conteúdo sob estudo. Na prática, eles podem também ser utilizados para a realização de comparações entre os diferentes domínios ou para estudos evolutivos dos próprios domínios, como mostram os artigos de Modesto et al. (2005); Tolosa et al. (2007).

Em resumo, apesar de uma comparação direta entre os resultados do presente trabalho e os dos artigos apresentados nesta seção não ser possível, esses trabalhos ajudaram tanto a mostrar a importância de diversos dos atributos utilizados, como também apresentaram uma gama de metodologias de coleta de atributos de Web sites.

3.2 Pesquisas Qualitativas

Esta segunda classe de indicadores é mais relevante para a realização de pesquisas analíticas sobre um domínio Web, podendo ser direcionada para o estudo de comportamentos ou tendências tecnológicas e sociais. Alguns trabalhos com resultados interessantes nessa área são: Bauer & Scharl (2000); Canali, Cova, & Vigna (2011); Chen et al. (2005); Edward et al. (2001); Eppler & Muenzenmayer (2002); Freire, Castro, & Fortes (2009); Katerattanakul & Siau (1999); Muller & Schwarz (2011); Sloan, Gregor, Rowan, & Booth (2000); Yoo & Donthu (2001).

De diversas maneiras essas publicações demonstram a importância de se avaliar a qualidade de um *site*, e como isso melhora a forma como um *site* é visto por seus usuários. O artigo de Eppler & Muenzenmayer (2002), por exemplo, apresenta alguns dos problemas relacionados à falta de qualidade de um *site*:

- Informações desatualizadas;
- Navegação confusa;
- Quantidade exagerada de informações na página inicial;
- Inconsistência de estilo entre as páginas;
- Infraestrutura não confiável;
- Falhas de segurança;
- Presença de textos grandes de difícil compreensão.

Esse e outros artigos, como os escritos por Katerattanakul & Siau (1999); Yoo & Donthu (2001), consolidam os tipos de problemas mais comuns em diversas classes de acordo com a forma como interferem na percepção de qualidade dos *sites*, facilitando o trabalho de análise e posterior correção dos mesmos. O trabalho de Eppler & Muenzenmayer (2002) vai além, ao classificar e sugerir ferramentas que possam ser utilizadas para sanar os tipos de problemas citados:

- Monitores de performance;
- Analisadores de *site*;
- Analisadores de tráfego;
- Mineradores Web; e,
- Ferramentas de *feedback* de usuários.

Apesar de esses artigos conterem interessantes resultados e metodologias para a análise de web *sites*, em sua maioria as ferramentas são voltadas para o uso interno de uma empresa ou por desenvolvedores como forma de avaliação de um *site* ou pequeno grupo de *sites*.

O intuito do presente projeto, no entanto, é estudar um conjunto de características de qualidade em grandes grupos de páginas, o que inviabiliza o uso de diversas das ferramentas citadas. Por exemplo, as ferramentas de análise de tráfego de um *site* não podem ser utilizadas sem que se tenha acesso ao sistema ou à rede desse *site*. Consequentemente, a classe de ferramentas formadas por analisadores de *sites* são as mais adequadas. Exemplos desse tipo de ferramenta compreendem analisadores de links quebrados, de falhas em formulários, da falta de atributos em marcações HTML, dentre outros.

Nesse sentido, os artigos de Chen et al. (2005); Muller & Schwarz (2011); Sloan et al. (2000) discorrem sobre a importância da validação de código para a melhoria da apresentação de um *site*, tanto por parte dos padrões HTML, quanto dos padrões de acessibilidade. Em particular, o artigo de Chen et al. (2005) mostra um estudo com páginas Web de diversas fontes, que obteve como resultado a informação de que apenas 5% das páginas avaliadas serem aderentes aos padrões HTML definidos pelo W3C. Apesar desse estudo não ser estatisticamente válido para a generalização à outros domínios, devido à falta de métodos que comprovem a relevância da amostra para o escopo (toda a Web), ele contém forte indício da falta de preocupação com a conformidade com os padrões.

Em outra direção, o artigo de Freire et al. (2009) apresenta pesquisa sobre a acessibilidade em *sites* governamentais, em especial as do governo brasileiro. Nele, a não adoção de indicadores de acessibilidade mais simplificados como a

verificação dos níveis 'A', 'AA' e 'AAA' propostos pelo WCAG, é justificada por eles pouco informarem sobre o real estado dos *sites* não acessíveis. Com isso, é proposta a utilização de uma métrica correlata chamada *Web Accessibility Barrier* (WAB) descrita por Parmanto & Zeng (2005) que pondera a quantidade de erros existentes e seus respectivos níveis de prioridade, conforme descrito na Equação 3-1. Nessa pesquisa, foi utilizado o conteúdo de páginas dos governos estaduais presente na ferramenta *Internet Archive*⁴ para a avaliação do histórico de acessibilidade da Web Governamental Brasileira. É importante salientar que tal ferramenta não mantém histórico de todos os *sites* e que mesmo dos *sites* mantidos, nem todas as suas páginas são armazenadas.

$$WAB_{site} = \frac{\sum_{p=1}^{NP} \sum_{possível\ barreira\ b \in WCAG\ 1.0} \left(\frac{B_{bp}}{P_{bp}} \right) (W_b)}{NP} \quad 3-1$$

Onde,

NP é o número total de páginas do *site*;

B_{bp} é o número de barreiras (erros segundo o WCAG) do tipo b encontradas em uma página p ;

P_{bp} é o número de barreiras em potencial do tipo b encontradas em uma página p , ou seja, trata-se do número máximo de erros que um determinado tipo de erro pode ocorrer em uma dada página; e,

W_b é o peso associado a cada tipo de barreira, segundo ordem inversa de seus valores de prioridade descritos no WCAG

Já o artigo de Canali et al. (2011) mostra e discute uma ferramenta, chamada *prophiler*, capaz de filtrar de forma eficiente a existência de elementos maliciosos em páginas Web, um importante item para a avaliação da segurança em *sites* Web. Essa ferramenta é comparada com outra, Wepawet apresentado por Cova, Kruegel, & Vigna (2010). Apesar da primeira não ser tão robusta para a garantia da existência

⁴ Disponível em: <http://www.archive.org/>. Acessado em: 18 de abril de 2014

ou não de código malicioso em uma página quanto a segunda, ela possui desempenho significativamente maior na identificação de casos de páginas seguras e pode ser utilizada como filtro para diminuir a carga dessa segunda quando se deseja analisar um grande montante de páginas.

Também pode-se citar o artigo de Bauer & Scharl (2000) por apresentar uma proposta similar à do TIC Web, com a automatização da análise da qualidade de *sites* de organizações sem fins lucrativos e seu posterior agrupamento em classes. Nele, discute-se também a importância da utilização de *frameworks* para avaliação de *sites* e como os processos de automáticos se destacam dos métodos manuais frequentemente utilizados.

Por sua vez, o artigo de Edward et al. (2001) propõe a utilização dos metadados fornecidos pelas próprias páginas para melhor compreender amostras. Nesse trabalho, são detalhados os tipos de metadados mais frequentes em páginas Web, diferenciando aqueles costumeiramente incluídos por ferramentas de geração de código automático daqueles usualmente adicionados por desenvolvedores, como a inserção de palavras chaves que descrevam o conteúdo dos documentos. No entanto, principalmente devido à falta de padronização na maneira como são incluídos, pouco se conseguiu concluir sobre sua utilidade na descrição de páginas e generalização para grupos ou domínios de *sites*.

Em contraposição aos artigos relativos à estudos quantitativos, os artigos dessa seção, propõe análises mais aprofundadas às características que venham a definir a qualidade de um *site*. Em sua maioria, no entanto, não conseguem propor metodologias que consigam lidar com grandes quantidades de *sites*.

4 METODOLOGIA DE ANÁLISE DE CONJUNTOS DE SITES

A presente proposta constitui-se de um aprofundamento dos estudos sobre os dados obtidos pelo projeto TIC Web, principalmente no que diz respeito à busca e identificação de grupos de *sites* com características semelhantes. Não apenas ela utilizou as informações geradas pelo projeto, como também, foram estudadas formas de processá-los para a aplicação de técnicas de aprendizado de máquina. Consequentemente, o método foi desenvolvido com base nos indicadores do TIC Web descritos no Apêndice 5. Isso, no entanto, não impede que ele seja aplicado para outros conjuntos de indicadores ou atributos.

Assim, esse estudo mostrou-se relevante tanto para a construção do método aprofundado de análise dos dados adquiridos pelo TIC Web quanto para sua posterior utilização em processo de classificação de *sites* que demonstre a experiência de uso percebida de um *site*. Ainda, auxilia na indicação de grupos de *sites* nos quais se devem focar os esforços para a melhoria da qualidade.

Para atingir seus objetivos, foram estudadas e comparadas técnicas de aprendizado de máquina não supervisionado, capazes de identificar agrupamentos de *sites* com características comuns. Isso exigiu a utilização de alguns sistemas de análise estatística e de mineração de dados e, com esse intuito os sistemas Octave⁵, R⁶ e Weka⁷, já consagrados no meio acadêmico, foram considerados. Tanto o Octave quanto o R são sistemas de análise estatística de dados que facilitam seu manuseio e visualização. Já o Weka é um programa *open source* bastante utilizado em pesquisas acadêmicas por apresentar diversos algoritmos de análise de dados e de aprendizado de máquina já implementados.

O artigo de Bauer & Scharl (2000) ajuda a definir os elementos a serem especificados em metodologias para análise e categorização de *sites* e possui uma precisa definição para cada uma das alternativas de análises de *sites* Web: classificações manuais, análises textuais, classificações por técnicas estatísticas, redes neurais não supervisionadas e redes neurais supervisionadas. Embora possua embasamento teórico bastante completo, o conjunto de *sites* analisado foi reduzido,

⁵ Disponível em: <http://www.gnu.org/software/octave/>. Acessado em: 18 de abril de 2014

⁶ Disponível em: <http://www.r-project.org/>. Acessado em: 18 de abril de 2014

⁷ Disponível em: <http://www.cs.waikato.ac.nz/ml/weka/index.html>. Acessado em: 18 de abril de 2014

prejudicando a generalização dos resultados. Com base nesse e em outros artigos citados no Capítulo 3, as etapas do processo de análise, mostradas na Figura 4.1 puderam ser montadas.

Figura 4.1 – Sequência de etapas proposta pela metodologia

A última etapa, de generalização, foi pensada como uma alternativa para melhor aplicar os resultados obtidos com as análises realizadas. A partir de tais resultados, é possível a criação um índice de qualidade que consiga nivelar *sites* de acordo com suas características de qualidade. Com essa validação, desenvolvedores teriam meios para comparar a qualidade de seus *sites* e projetar melhorias. Ainda, a classificação *off-line* dos *sites* de certos domínios, principalmente do *.gov.br*, poderia ser utilizada para a organização do foco de ação de políticas públicas voltadas à melhoria da qualidade da informação ofertada aos cidadãos.

Os itens a seguir descrevem o significado de cada uma dessas etapas, quais tarefas cada uma engloba, sua importância no contexto da análise e os artefatos que devem ser gerados em sua execução. No capítulo seguinte, de número 1, esses passos são colocados em prática para realização da avaliação da Web Governamental Brasileira.

4.1 Adaptação de atributos

É fato que, dada a origem e metodologia de extração dos atributos dos *sites* do governo brasileiro, a diversidade de tipos de atributos obtidos é relativamente grande. Isso se torna um problema pois, em geral, algoritmos de aprendizado de máquina não supervisionado não possuem suporte adequado ao tratamento de sequências de caracteres. Assim, esta fase de adaptação de atributos objetiva a padronização do formato dos dados a serem utilizados na análise de modo que todos possuam uma representação numérica.

Um exemplo é o indicador “tipo de servidor”, pois apesar de na prática existirem poucas opções popularmente utilizadas, como o *Apache Server* ou *Microsoft IIS*, os registros cadastrados contêm respostas completas enviadas pelos servidores pesquisados. Ou seja, ao invés de conterem somente o nome da aplicação, eles contêm também suas versões e até módulos ativos. Isso é prejudicial às técnicas de análise por aumentar significativamente a quantidade de categorias de servidores de aplicação. Para adaptar esse tipo de atributo deve-se primeiramente identificar conjuntos de valores frequentes nos dados e enumerá-los. Caso necessário, esses grupos podem agregar valores semelhantes e não necessariamente iguais, tornando-os mais representativos. Isso pode ser visto no agrupamento das diversas versões de servidores Apache encontrados em um único grupo que os identifique. Note que, de certo ponto de vista, essa categorização prévia dos atributos é considerada uma técnica de aprendizado de máquina supervisionada, pois utiliza conhecimento prévio sobre as alternativas de tipos de *sites* para dividir os registros com base no atributo original.

Especificamente para esta pesquisa, dada a necessidade da utilização das informações obtidas para as unidades de páginas, os dados dessa unidade de pesquisa foram agrupados e recompilados para compor novos atributos aos *sites* a que pertencem.

Podem ocorrer casos em que a adaptação dos valores originais para unidades numéricas não faça sentido, como é o caso dos nomes de *sites* e seus endereços IPv4 e IPv6. Nesses casos, essas variáveis devem ser diretamente removidas da base.

O banco de dados com os novos campos que representem os grupos criados é o principal artefato desta etapa. Todavia, é interessante também descrever quais dos atributos foram removidos, quais adaptados e o que foi realizado para a execução de tal transformação.

Esta etapa pode ser explicitada pela equação 4-1. Nela, a função $A(X)$ de adaptação dos atributos é responsável realizar todas as transformações necessárias aos valores originais de cada registro de forma que eles correspondam ao novo conjunto de atributos adaptados p' .

$$X_{n \times p'}^{adaptado} = A(X_{n \times p}^{original}) \quad 4-1$$

Onde,

$X_{n \times p}^{original}$ é matriz de valores para cada um dos p atributos originais da base de dados com n registros de *sites*;

$A(X)$ é a função de adaptação dos atributos;

$X_{n \times p'}^{adaptado}$ é a matriz resultante de valores para cada novo atributo p' para a todos os n registros de *sites* da base de dados.

4.2 Seleção manual de atributos

A seguir, vem a seleção manual dos atributos que serão utilizados na análise e, como o próprio nome sugere, ela compreende a escolha das variáveis que serão utilizadas para avaliar e caracterizar os *sites* pesquisados.

A importância dessa fase decorre do fato de ela determinar o escopo da pesquisa e os artefatos gerados influenciarem diretamente a forma como os resultados finais deverão ser interpretados. Caso, por exemplo, nela fossem selecionados preferencialmente atributos relacionados à acessibilidade do *site*, os resultados da análise ficariam mais direcionados à essa característica. Em contra partida, se os atributos estiverem mais envolvidos com variáveis de desempenho, a interpretação dos resultados seria completamente alterada.

Considerando o escopo de variáveis descrito no Apêndice 5, decidiu-se no presente projeto por fazer a comparação dos resultados atingidos a partir da escolha de 4 conjuntos de atributos, um com todos os adquiridos, um com atributos sobre validação de código, outro sobre a infraestrutura dos servidores dos *sites* e, um último grupo sobre a estrutura dos *sites*. Assim, os proventos desta etapa consistem de uma lista dos atributos selecionados para cada um dos casos.

Tomando-se como base a lista de atributos adaptados p' resultante da etapa anterior, nesta etapa deve-se selecionar $p'_1, p'_2, \dots, p'_m$ listas de atributos para que se consiga avaliar m diferentes aspectos da base dados. Com a especificação de tais listas, deve-se gerar as novas bases de dados $X_{n \times p'_1}^{adaptado}, X_{n \times p'_2}^{adaptado}, \dots, X_{n \times p'_m}^{adaptado}$ correspondentes a elas. O processo de geração é simples e exige apenas a seleção das colunas referentes a cada um dos atributos descritos nas listas $p'_1, p'_2, \dots, p'_m$.

4.3 Normalização de atributos

Não apenas a diversidade de tipos de atributos, mas também a disparidade entre seus valores altera a eficiência da execução dos algoritmos de aprendizado computacional. Dada a dependência de passos iterativos por parte dos algoritmos a serem utilizados, pode ser matematicamente comprovado que sua convergência é mais lenta caso as escalas de grandeza dos atributos sob análise sejam díspares. Isso é particularmente verdadeiro para os métodos que envolvem a técnica de descida do gradiente descrita por Mitchell (1997). Nesse sentido, segundo Jain & Dubes (1988), a normalização de atributos configura boa solução ao deixar as médias de todos os atributos em torno de zero e suas faixas de valores entre -1 e 1.

Vale salientar que a realização desta etapa melhora a qualidade dos resultados, pois, caso atributos com faixas de valores muito diferentes sejam utilizados, o nível de relevância entre eles também fica distorcido, deixando os resultados mais suscetíveis às variações de atributos com escalas de grandeza maiores.

A Equação 4-2 compreende uma maneira de se executar a normalização dos atributos numéricos de determinada base de dados.

$$x_{ij} = \frac{x_{ij}^* - \mu_j}{s_j} \quad 4-2$$

onde,

x_{ij} é o valor final do elemento i de um determinado atributo j;

x_{ij}^* é o valor original do elemento i de um determinado atributo j;

μ_j é a média dos valores do atributo j, e;

s_j é a variância dos valores do atributo j, alternativamente, esse valor pode corresponder à diferença entre os valores máximo e mínimo do atributo j.

Assim, se aplicarmos essa formula a todas as bases de dados resultantes teremos as novas matriz de valores $X_{n \times p'_1}^{normalizado}$, $X_{n \times p'_2}^{normalizado}$, ..., $X_{n \times p'_m}^{normalizado}$. Esse processo é explicitado por:

$$X_{n \times p'_l}^{normalizado} = N(X_{n \times p'_l}^{adaptado}) \quad 4-3$$

Onde,

l é um valor de 1 a m ;

$N(X)$ é a função que aplica a Equação 4-2 de normalização aos valores da matriz X ;

$X_{n \times l}^{adaptado}$ é a matriz de valores adaptados e selecionados pela lista de atributos p'_l ;

$X_{n \times p'_l}^{normalizado}$ é a matriz resultante, de valores normalizados, para a lista de atributos p'_l .

4.4 Redução dimensional

Quando se lida com grande volume de dados de treinamento ou grandes quantidades de atributos ou, ainda, ambos, é fácil de se chegar a situações em que a análise dos dados por meio das técnicas de aprendizado de máquina passem a

ser excessivamente custosas computacionalmente ou que o excesso de atributos acabe influenciando negativamente os resultados obtidos ou ainda que a visualização dos resultados fique prejudicada. Para evitar o descarte incorreto de atributos, técnicas de priorização de atributos ajudam a selecionar as dimensões mais relevantes para a análise. Ou seja, os atributos ou grupos de atributos que melhor conseguem diferenciar os dados são identificados. A partir da definição da quantidade adequada de atributos que se queira trabalhar, a seleção dos mais úteis ao processo fica facilitada.

Segundo Pudil & Hovovicova (1998), os métodos de redução dimensional podem ser divididos em duas categorias. A primeira, chamada de seleção de atributos, em inglês denominada *Feature Selection Algorithms* (FSAs), corresponde a algoritmos com o objetivo de escolher o subconjunto de atributos que melhor aperfeiçoe determinado critério de avaliação de custo entre todos os subconjuntos de mesmo tamanho. Já a segunda categoria, extração de atributos, também denominada por *Feature Extraction Algorithms* (FEAs), engloba algoritmos que transformam o conjunto total de atributos em um novo, não diretamente equivalente, mas de dimensionalidade menor.

A primeira categoria é utilizada quando os atributos originais são relevantes para a análise dos dados e vão futuramente alterar o processo de aquisição. É o caso pesquisas na área medicinal, nas quais a quantidade de atributos pode atrasar significativamente o processo de coletá-los. Já a segunda, é utilizada quando se almeja maior qualidade na discriminação dos dados e quando a interpretação dos resultados não precisa ser detalhada por atributo.

Exemplos de FSAs são: *Las Vegas Filter* (LVF) descrito por Liu & Setiono (1996), *Relief Algorithm* descrito por Kira & Rendell (1992) e *min-Redundancy Max-Relevance* (mRMR) descrito por Peng, Long, & Ding (2005). No artigo de Molina, Belanche, & Nebot (2002) está consolidada a descrição e comparação de uma grande variedade de métodos de seleção de atributos existentes na literatura. Quanto à extração de atributos, pode-se citar *Principal Component Analysis* (PCA) descrito por Jolliffe (2005) e *Factor Analysis* (FA) descrito por Mardia, Kent, & Bibby (1995).

Principalmente devido aos requisitos de qualidade na separação dos agrupamentos do presente projeto, a categoria de redução dimensional FEA foi

escolhida. Outro ponto que influenciou a decisão foi a facilidade de se coletar grandes volumes de atributos para as unidades pesquisadas quando se comparados a outras áreas de estudo. Isso é consequência das coletas serem automatizadas e da natureza das informações coletadas, puramente objetivas e sem intervenção manual.

Para exemplificar o funcionamento desse tipo de algoritmo, o PCA foi selecionado. Seguindo o objetivo dos FEAs, esse algoritmo se foca em reduzir a dimensionalidade de determinada base de dados mantendo, a medida do possível, as variações presentes por toda a base. Isso é feito com a transformação do conjunto de variáveis à um novo, não correlacionado e ordenado segundo sua relevância. Mais sobre o funcionamento desse mecanismo pode ser encontrado na Seção 1.1 do livro de Jolliffe (2005). Basicamente, ele pode ser descrito pelas seguintes etapas:

1. Tradução dos dados

Nesta etapa os valores devem ser normalizados para que seu centro fique zerado. No presente projeto esse processamento foi descrito na Seção 4.4, referente à normalização dos atributos, e, portanto, não se faz necessária.

2. Cálculo da matriz de covariâncias

Para cada uma das m listas previamente selecionadas, as matrizes de covariâncias podem ser calculadas pela Equação 4-4:

$$\Sigma_{p'_l * p'_l} = \frac{X_{n * p'_l}^{normalizado'} * X_{n * p'_l}^{normalizado}}{n} \quad 4-4$$

Onde,

$\Sigma_{p'_l * p'_l}$ é a matriz de covariâncias para dos valores para a lista de atributos p'_l ;

l é um valor de 1 a m ;

n é o número de *sites* da base de dados;

$X_{n * p'_l}^{normalizado}$ é a matriz resultante, de valores normalizados, para a lista de atributos p'_l ;

$X_{n * p'_l}^{normalizado'}$ é a transposta da matriz $X_{n * p'_l}^{normalizado}$

3. Calculo dos autovalores

Estes também podem ser chamados de componentes principais, dando origem ao nome do algoritmo. Pode ser matematicamente provado que os autovetores e autovalores de determinada matriz podem ser encontrados com a aplicação da Decomposição em Valores Singulares (SVD) na matriz de covariância dos dados.

O SVD pode ser descrito pela seguinte decomposição de uma determinada matriz de covariâncias Σ :

$$SVD(\Sigma) = U * S * V \quad 4-5$$

Onde,

Σ é a matriz de covariâncias;

U e V são equivalentes e cada uma de suas colunas representa um autovetor da matriz X sob análise;

S é a matriz singular diagonal que representa os valores singulares, ou autovalores, de cada um dos autovetores descritos pelas colunas de U ou V .

Como exemplo de aplicação, pode-se mostrar a seguinte matriz diagonal:

$$S = \begin{bmatrix} 3,4576 * 10^4 & 0 & 0 \\ 0 & 3,1959 * 10^4 & 0 \\ 0 & 0 & 3,3793 * 10^2 \end{bmatrix} \quad 4-6$$

Como os valores dessa matriz representam a relevância de cada um dos autovetores, se essa matriz fosse obtida em um processo de redução dimensional de 3 para 2 atributos, a relevância dos atributos selecionados seria de:

$$\frac{3,4576 * 10^4 + 3,1959 * 10^4}{3,4576 * 10^4 + 3,1959 * 10^4 + 3,3793 * 10^2} = 99,5\% \quad 4-7$$

4. Redução dimensional

Por último, deve-se escolher a quantidade de autovetores para os quais as matrizes de dados $X_{n \times p}^{normalizado}$ serão projetadas. Não existe um consenso sobre o melhor valor a ser escolhido. Pode-se priorizar a manutenção da relevância dos atributos ou escolher um valor arbitrário passível de ser mostrado graficamente. A Equação a seguir resume esse processo:

$$X_{n*d_l}^{reduzido} = X_{n*p'_l}^{normalizado} * U_{p'_l*d_l}$$

4-8

Onde,

$X_{n*d_l}^{reduzido}$ é a nova matriz de valores com d_l atributos para a lista p'_l previamente selecionada;

$X_{n*p'_l}^{normalizado}$ é a matriz de valores normalizados para a lista de atributos p'_l ;

$U_{p'_l*d_l}$ é a matriz dos d_k autovetores de $X_{n*p'_l}^{normalizado}$;

d_l é o número de dimensões que a matriz $X_{n*p'_l}^{reduzido}$ terá;

l é um valor de 1 a m , tal que m é o número de listas previamente selecionado.

Independentemente da escolha do algoritmo, o resultado desta análise consiste em uma base de dados nova, modificada e com dimensionalidade reduzida, $X_{n*d_l}^{reduzido}$, pronta para ser utilizada por algoritmos de detecção de agrupamentos.

4.5 Agrupamento de elementos

Um dos principais objetivos da presente pesquisa é conseguir identificar conjuntos, ou aglomerados, de *sites* com características comuns dentre os *sites* observados na coleta realizada pelo projeto TIC Web. No campo de aprendizado de máquina, existem diversas técnicas e métodos para a realização desse tipo de tarefa. No presente projeto, duas técnicas foram estudadas e seus resultados foram comparados a identificação daquele que melhor se aplica aos conjuntos de dados em questão. O *K-means* descrito por Hartigan & Wong (1979) que se baseia na distância euclidiana entre os registros para decidir em qual agrupamento cada um pertence e, o algoritmo *Expectation Maximization* (EM) descrito por Dempster, Laird, & Rubin (1977), baseado em curvas gaussianas.

O K-means é um algoritmo iterativo bastante simples. Ele usa o conceito de centroides para a caracterização dos grupos definidos, sendo um centroide um vetor com valores para cada um dos atributos que definem os elementos na base de dados, ou seja, ele representa um determinado ponto no espaço definido pelo conjunto de atributos em questão. Assim, a partir um conjunto inicial de centroides de tamanho igual ao número de agrupamentos que se deseja encontrar, a cada passo da iteração todos os elementos da base de dados são classificados em um dos grupos de acordo com sua distância euclidiana a cada um dos centroides. A seguir, antes do fim do passo da iteração, a posição de cada centroide é atualizada para corresponder à posição média dos elementos a ele atribuídos. O ciclo iterativo segue com novas classificações dos dados seguidas de atualizações das posições dos centroides até que os resultados converjam. A condição de parada pode ser definida por um número máximo de iterações a serem executadas ou pela verificação de que os resultados atingiram uma posição estável, sem mais alterações das classes dos elementos.

Nesse algoritmo, o grupo inicial de centroides compõe fator determinante no encontro da melhor solução para agrupar dos dados, uma vez que o algoritmo é susceptível à se prender à mínimos locais de sua função de custo. Para minimizar a existência de tal problema, o algoritmo é executado múltiplas vezes com escolhas aleatórias dos centroides iniciais e aquele conjunto que apresentar menor valor para a função de custo, dada pela Equação 4-9, é escolhido como solução.

$$J = \sum_{z \in \tau} \sum_{x \in C_z} \|x - z\|^2 \quad 4-9$$

Onde,

τ é o conjunto de centroides;

C_z o conjunto de elementos agrupados pelo centroide z .

Já o EM é mais complexo. O artigo de Figueiredo & Jain (2002) apresenta uma forma de utilizar o método, normalmente aplicado no cálculo de máxima verossimilhança (*Maximum Likelihood Estimation* - MLE), como uma técnica iterativa de aglomeração de elementos. Ele é descrito por um processo iterativo para estimar

um vetor θ com de variáveis desconhecidas de um determinado registro $x \in X$, no caso, a média e a matriz de covariâncias. Formalmente, quer-se a estimativa:

$$\hat{\theta}(t) = \{\mu_j(t), \Sigma_j(t)\} \quad 4-10$$

Onde,

$\mu_j(t)$ é a média dos elementos de um agrupamento j no passo de iteração t ;

$\Sigma_j(t)$ é a matriz de covariâncias de um agrupamento j no passo de iteração t ;

$j = 1, \dots, M$ tal que M é igual ao número de agrupamento que se deseja encontrar

t é o passo de iteração e varia em $0, 1, \dots$

A inicialização de $\hat{\theta}(t = 0)$ é randômica e o processamento no passo de iteração pode ser dividido em dois

- **passo-E (*Expectation*)**: a partir de um conjunto de registros, X , e da estimativa atual das variáveis desconhecida, $\hat{\theta}(t)$, este passo é responsável pelo cálculo da probabilidade *a posteriori* de que um determinado *site* $x \in X$ pertença ao agrupamento C_j segundo a formula:

$$P(C_j|X) = \frac{|\Sigma_j(t)|^{-\frac{1}{2}} e^{\eta_j} P_j(t)}{\sum_{k=1}^M \left(|\Sigma_k(t)|^{-\frac{1}{2}} e^{\eta_k} P_k(t) \right)} \quad 4-11$$

Onde,

$$\eta_i = -\frac{1}{2} (X - \mu_i(t))^T \Sigma_i^{-1}(t) (X - \mu_i(t))$$

- **passo-M (*Maximization*)**, responsável pela atualização das variáveis desconhecidas para o passo seguinte da iteração, $\hat{\theta}(t + 1)$. Para isso as seguintes fórmulas são utilizadas:

$$\mu_j(t+1) = \frac{\sum_{k=1}^N P(C_j|x_k)x_k}{\sum_{k=1}^N P(C_j|x_k)} \quad 4-12$$

Onde,

N é o número de *sites* do conjunto X sob análise.

$$\Sigma_j(t+1) = \frac{\sum_{k=1}^N P(C_j|x_k) \left(x_k - \mu_j(t)\right) \left(x_k - \mu_j(t)\right)^T}{\sum_{k=1}^N P(C_j|x_k)} \quad 4-13$$

$$P_j(t+1) = \frac{1}{N} \sum_{k=1}^N P(C_j|x_k) \quad 4-14$$

Esse ciclo é executado até que seja verdade a inequação descrita em 4-15. E, ao fim do processo, a Equação 4-11 é utilizada para a classificação dos *sites* em seus respectivos agrupamentos.

$$\|\hat{\theta}(t+1) - \hat{\theta}(t)\| < \varepsilon \quad 4-15$$

Onde,

a operação $\| \cdot \|$ foi implementada como sendo a distância euclidiana entre os vetores $\mu(t+1)$ e $\mu(t)$;

ε um valor limite escolhido arbitrariamente para definir a convergência dos valores.

Nesta etapa do projeto, é feita a verificação da eficiência desses algoritmos ao serem aplicados a cada uma das bases dados selecionadas previamente, $X_{n \times d_l}^{reduzido}$, com $l = 1, \dots, m$ e m sendo o número de listas de atributos sob estudo. Além disso, como esses algoritmos são bastante sensíveis à seleção de seus parâmetros, principalmente o número de agrupamentos, eles foram feitos testes tanto para o número máximo de 3 grupos quanto para 5 grupos.

Apesar de neste trabalho não serem discutidas métricas para a avaliação do melhor número de grupos a ser utilizado, os valores 3 e 5 foram escolhidos por facilitarem a leitura das faixas de qualidade às quais um *site* poderá ser alocado.

Quanto à segmentação dos dados de treinamento, é importante salientar que para a validação desse procedimento, a separação da base em elementos de treinamento e elementos para a validação cruzada, comum em diversos procedimentos de aprendizado de máquina, não é aplicável devido a não existência informações sobre a categoria a qual cada um dos *sites* se encontra. Contudo, a múltipla execução dos algoritmos ainda se faz útil para impedir que as técnicas converjam para resultados não otimizados. Isso é feito com a comparação da função custo resultantes de cada uma das execuções.

Em decorrência dessa execução tem-se, para cada um dos algoritmos e variações de condições em teste, a classificação dos elementos da base de dados e um conjunto de variáveis que serão utilizadas para a classificação do universo de *sites* sob estudo. No caso do método K-means, esses parâmetros são descritos pelos centroides, enquanto que no método EM, eles são o par composto pelo vetor de média e as matrizes de covariâncias de cada. Assim, esta etapa pode ser explicitada pela equação:

$$[R_{n*1}^i, V^i] = Alg_i(X_{n*d_i}^{reduzido})$$

4-16

Onde,

R_{n*1}^i é vetor com a classificação de cada um dos *sites* em seu respectivo agrupamento para um determinado algoritmo i ;

V^i é o conjunto de variáveis que poderá ser utilizado para a classificação de novos *sites* para o algoritmo i ;

$X_{n*d_i}^{reduzido}$ é a matriz de valores reduzidos com d_i atributos para a lista p'_i previamente selecionada;

Alg_i é um dos algoritmos ou variações de algoritmos utilizados no processo de comparação, sendo $i = 1, \dots, i_{max}$.

No caso do presente projeto $i_{max} = 4$, sendo que os algoritmos aplicados foram o K-means para 3 e 5 grupos e o EM também para 3 e 5 grupos.

4.6 Visualização gráfica

Sendo o conjunto de dados analisados uma base com muitas dimensões, a visualização da distribuição dos *sites* nesse espaço fica prejudicada, uma vez que por métodos usuais consegue-se representar até três dimensões. Isso, no entanto, não minimiza a importância de se conseguir visualizar os dados em formatos diferentes do tabular. Nesse sentido, os FEAs apresentados na seção 4.4, por identificar e ordenar os atributos mais relevantes, são utilizados para delinear gráficos de duas e três dimensões.

Utilizando-se os dois ou três atributos identificados como mais relevantes, nesta etapa são gerados gráficos bidimensionais e tridimensionais que facilitam a visualização das distâncias entre os grupos ou de suas congruências internas. Além disso, caso divisões claras entre os grupos não existam, valida-se a qualidade das separações de fato atingidas pela execução dos algoritmos de aglomeração.

Outra opção aplicada para facilitar a posterior análise dos dados, é utilizar FSAs, também introduzidas na seção 4.4, para identificar os atributos originais mais representativos de cada um dos grupos encontrados e especificar suas características. Na presente metodologia, a técnica mRMR, também apresentada nessa seção, foi aplicada na etapa de Análise para viabilizar a descrição dos grupos segundo os atributos originais de seus respectivos elementos. Tal escolha deve-se à popularidade dessa técnica e sua existência prévia nas bibliotecas das ferramentas de desenvolvimento utilizadas no projeto.

O mRMR pode ser resumido da seguinte forma:

O algoritmo tenta encontrar o subconjunto de m atributos S representado por $\{x_i\}$ tal que a máxima relevância $D(S, c)$ seja encontrada para a classificação em agrupamentos c . Esse valor é estimado por:

$$\max D(S, c), \quad D(S, c) = \frac{1}{|S|} \sum_{x_i \in S} I(x_i; c)$$

Onde,

$I(x_i; c)$ é a função que calcula o valor da informação mútua entre os valores de cada *síte* para o atributo x_i e sua classificação por agrupamento c . O desenvolvimento dessa função foi descrito por Peng et al. (2005).

E para minimizar a redundância entre os x_i atributos selecionados, a seguinte condição também deve ser atendida:

$$\min R(S), \quad R(S) = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i; x_j)$$

Finalmente, o operador $\Phi(D, R)$ que otimiza ambos é chamado de “*minimal-redundancy-max-relevance*” e dá o nome ao algoritmo. Para otimizar tanto D quanto R a seguinte condição deve ser atingida:

$$\max \Phi(D, R), \quad \Phi = D - R$$

Na prática, existem algoritmos incrementais de busca para encontrar valores suficientemente bons.

Uma vez que a etapa de visualização é independente das modificações às bases de dados realizadas nas etapas anteriores do processo, dado que a marcação dos elementos pertencentes a cada grupo já foi realizada, a utilização de ambas as técnicas de visualização é viável dependendo de sua necessidade durante a realização das análises.

4.7 Análise de indicadores

A etapa de análise consiste da avaliação dos métodos aplicados a fim de se determinar o mais adequado para o conjunto dos dados. Além disso, é conveniente a inspeção dos grupos de *sites* formados a partir da identificação dos atributos que mais os distanciam, para que se consiga melhor enxergar uma escala de qualidade entre eles, tendo como base os valores esperados para as variáveis discutidas no Capítulo 2.

Realiza-se, portanto, a inspeção manual de exemplos de cada um dos grupos encontrados, bem como se utiliza algoritmos de redução dimensional com base em seleção de atributos para a verificação da relevância dos atributos originais. Isso é necessário pois os algoritmos de extração de atributo, a exemplo do PCA, realizam mudanças de bases algébricas para atingir seus objetivos, prejudicando a tradução direta de significado dos atributos visualizados graficamente. Nesta etapa o algoritmo de seleção mRMR foi aplicado para a realização dessa ordenação entre os atributos de cada uma das listas de variáveis sob análise. Como esse método necessita da classificação prévia dos elementos da base, sua utilização configura mais um caso de aplicação de técnicas de aprendizado computacional supervisionado.

Com as listas ordenadas de atributos, selecionam-se exemplos de cada um dos grupos formados para a avaliação das características que os definem. Comparando tais definições com as expectativas de valores para *sites* com maior ou menor nível de qualidade percebida, pode-se definir um nível de qualidade que represente o grupo. Para uma divisão em três grupos, a divisão de níveis poderia ser: *sites* bons, *sites* regulares e *sites* ruins.

Desta etapa obtêm-se como resultados uma descrição dos tipos de *sites* pertencentes a cada um dos grupos, a seleção do algoritmo que mais se enquadra às expectativas de divisões dos grupos e os parâmetros para a avaliação de novos *sites*.

4.8 Generalização dos classificadores

A ideia de se criar um índice de qualidade para os *sites* reside principalmente na necessidade que o governo possui de avaliar os *sites* desenvolvidos pelas instituições governamentais e delas exigir requisitos mínimos de qualidade. Esse índice, além de se constituir em um indicativo da qualidade, pode também servir como comparativo entre *sites* ou aplicativos Web, como forma de garantir estabilidade, compatibilidade e portabilidade. Nesse sentido, os resultados adquiridos na execução da metodologia de análise baseada em algoritmos de aprendizado de máquina podem ser generalizados para a avaliação da qualidade de outros *sites*.

Apesar de já existirem na Web ferramentas especializadas na avaliação e testes de *sites*, como o *Page Speed Insights*⁸ ou o *YSlow*⁹, elas se focam na análise de desempenho dos *sites*, enquanto o índice proposto avalia de forma mais abrangente a usabilidade do *site* e aderência a padrões abertos, sendo portanto, complementar à tais ferramentas. Outras propostas, como as dos artigos Katerattanakul & Siau (1999); Yoo & Donthu (2001) também não se enquadrariam por envolverem processos manuais e, conseqüentemente não escalarem para as necessidades do presente trabalho.

Na prática, o processo de classificação de *sites* com base nos atributos obtidos na avaliação da Web Governamental brasileira é semelhante ao de análise da base. A partir da extração do conjunto de atributos de um *site* com as ferramentas de coleta descritas no Apêndice 6, o conjunto de variáveis deve:

- Ser adaptado, segundo as regras definidas pelo passo 1 da metodologia;
- Ser selecionado, segundo as listas de atributos definidas pelo passo 2;
- Ser normalizado, segundo médias e desvio padrão encontrados no passo 3;
- Passar pela transformação de redução dimensional com a utilização das matrizes de covariância, autovalor e autovetor definidas no passo 4;

⁸ Disponível em: <https://developers.google.com/speed/pagespeed/insights>. Acessado em: 18 de abril de 2014

⁹ Disponível em: <http://developer.yahoo.com/yslow/>. Acessado em: 18 de abril de 2014

- Passar pelos processo de classificação de grupo, segundo resultados dos algoritmos escolhidos no quinto passo;
- Receber as classificações de qualidade, segundo os grupos em que se encontra e seus respectivos níveis de qualidade definidos no passo 7; e,
- Opcionalmente, receber um nível único de qualidade segundo composição dos resultados obtidos para cada um dos conjuntos de atributos sob análise.

Como forma de implementação desse analisador, pensou-se no desenvolvimento de uma ferramenta similar ao *W3C Markup Validator* apresentado na Seção 6.2.1 do Apêndice 6. Nesse caso, a diferença seria a possibilidade de melhorias no processo de avaliação devido à continuidade do processo de coleta de dados e adição de atributos conseguidos com o projeto TIC Web. Assim, de tempos em tempos, com a inclusão de novos *sites* à base de dados, os parâmetros utilizados para a classificação dos *sites* poderiam ser refinados com novas execuções do processo de classificação dos *sites*. Essa implementação, entretanto, não entrou no escopo desta dissertação e é proposta como sugestão para a realização em trabalhos futuros.

5 ESTUDO DE CASO

Neste Capítulo é apresentada uma demonstração da execução da metodologia proposta no Capítulo 4 para os dados obtidos do estudo sobre a Web Governamental Brasileira realizado no projeto TIC Web. Para melhor contextualização do escopo de dados em análise, o Item 5.1 apresenta a série histórica dos resultados obtidos nesse projeto. Os Apêndices 5 e 6 descrevem os indicadores e ferramentas utilizadas para a aquisição desses dados, ajudando na compreensão das limitações e características da execução. A segunda parte deste quinto capítulo, seção 5.2 apresenta a execução da metodologia definida no quarto capítulo, utilizando base de dados reais do universo de *sites .gov.br*.

5.1 Características da Web Governamental Brasileira

Apesar do significativo aumento da capacidade computacional e da largura de banda utilizada no TIC Web em comparação com trabalhos citados no Capítulo 3, como o artigo de Modesto et al. (2005), a Web sofreu mudanças ainda maiores, e sua análise por completo continuou exigindo custos de hardware muito elevados. Esse foi um dos fatores que motivou a equipe a começar as análises por domínios menores, como da Web Governamental Brasileira, como forma de aprender sobre as técnicas de extração das características dos *sites* para a posterior aplicação da metodologia de forma amostral em domínios maiores.

Em 2012, no 18º Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia), foi publicado um artigo curto com o título “*ICT Web: Analysis of the Brazilian Governmental Web*” escrito por Ganzeli et al. (2012) que apresenta os resultados preliminares da pesquisa TIC Web. Nesse artigo foi discutida brevemente a importância dos temas relacionados à padrões abertos, conforme aqui apresentado no Capítulo 2 para a posterior exposição dos resultados obtidos nas análises realizadas nos dois primeiros anos de coletas do domínio *.gov.br*.

Os resultados nele apresentados compreendem o tamanho da Web, a validação de páginas, a utilização de IPv6, a sincronização dos relógios dos servidores e os tipos de servidores Web mais utilizados. Nos subitens a seguir está a combinação dos dados desses dois anos, com os referentes à coleta mais recente,

realizada em 2012, que foram utilizados como base para o estudo de caso da presente dissertação.

5.1.1 *Tamanho da Web*

Conforme discutido na Seção 6.1 do Apêndice 6, para a obtenção do número e tamanho total de *sites* e páginas pertencentes à Web Governamental Brasileira, utilizou-se uma modificação do *web crawler* WIRE. Sua semente foi formada pela junção dos registros de domínios do TLD *.gov.br* com outras duas fontes: lista de *sites* providos por instituições governamentais responsáveis por gerir domínios Web dos governos estaduais e municipais, e uma segunda lista formada por buscas a *sites* governamentais no buscador “Google.com”. Tais esforços de enriquecimento da semente serviram como meio de assegurar sua qualidade e abrangência.

A Tabela 5.1 a seguir mostra o volume em Gigabytes das páginas, e as quantidades totais de *sites* e páginas coletadas em cada um dos anos em que a pesquisa foi realizada. É importante salientar que a somatória dos volumes das páginas contabiliza apenas o conteúdo HTML de tais documentos, ignorando elementos como imagens, vídeos e outros aplicativos normalmente referenciados em páginas Web.

Tabela 5.1 - Volume e quantidade de site e páginas por ano de realização da coleta.

Ano	Volume(Gigabytes)	Número total de <i>sites</i>	Número total de páginas válidas
2010	148	11 900	6 300 597
2011	172	12 263	6 874 926
2012	313	14 152	10 863 943

Dessa tabela, é interessante notar que os resultados para o ano de 2012 indicaram um ritmo de mudanças bastante díspar. O volume de páginas sofreu crescimento íngreme, mostrando grande aumento na quantidade de conteúdo publicada no ano. Além disso, nota-se um acréscimo nas relações de tamanho de página e *site* médios, que passaram respectivamente de 26KB para 30KB e de 14MB para 22MB no período. Essa situação pode indicar maior enfoque do governo na utilização da Web em seus serviços. É importante notar também que, a cada ano,

a varredura realizada é mais abrangente devido a inclusão de novos *sites* à semente inicial da coleta.

5.1.2 Validação de páginas

Para todas as páginas baixadas, foram executados os testes de validação de código e de acessibilidade, descritos na Seção 6.2.1 do Apêndice 6, com a finalidade de se obter a proporção das páginas da Web Governamental Brasileira aderentes aos principais padrões abertos de desenvolvimento de páginas Web. Desta forma, cada uma das páginas foi submetida aos processos de validação de código HTML, e de validação de acessibilidade segundo o padrão E-MAG.

A Figura 5.1 mostra os resultados dessa avaliação para as coletas realizadas de 2010 a 2012, as colunas referentes à validação HTML mostram os totais percentuais de páginas que não apresentaram erros para essa validação e, as referentes a validação de acessibilidade, mostram os totais percentuais de páginas com os requisitos mínimos de acessibilidade, ou seja, que receberam nota igual ou maior que 'A'.

No que diz respeito à conformidade com os padrões HTML, observa-se uma progressão com razão um pouco menor do que 1%. Dado que os números de páginas vem aumentando significativamente, essa progressão mostra que não apenas os *sites* do governo estão conseguindo manter a base de páginas válidas, mas que também estão conseguindo difundir o padrão a muitos novos conteúdos.

Em relação à acessibilidade, embora o número percentual de casos tenha baixado de 2010 para 2012, observou-se melhora significativa na quantidade de páginas com um mínimo de acessibilidade, de aproximadamente 384 mil em 2010, ela passou para aproximadamente 505 mil em 2012.

Como os dados obtidos para essas validações são bastante simplificados, por conterem apenas as quantidades de erros alcançados nos testes, uma análise mais aprofundada sobre os tipos de erros mais frequentes fica inviabilizada.

Figura 5.1 - Gráfico da porcentagem de páginas da Web Governamental Brasileira em conformidade com os padrões HTML e de acessibilidade de 2010 a 2012.

5.1.3 Sincronização dos relógios dos servidores

A Figura 5.2 exibe os dados obtidos para a medição da qualidade da sincronização dos relógios dos servidores da Web Governamental Brasileira, baseada nas trocas de mensagens HTTP, para as coletas realizadas de 2010 a 2012. Nesse gráfico, os servidores são separados em 7 categorias de acordo com a diferença de tempo apresentada entre seu relógio e o relógio de uma máquina sincronizada via NTP, correspondendo à referência UTC de tempo. Devido à imprecisão dos testes realizados por esse método, conforme apresentado na Seção 6.2.2 do Apêndice 6, um servidor é considerado sincronizado (sync) com a Hora Legal Brasileira quando seu resultado difere da hora de referência por até 1 segundo.

Para esse indicador, assim como para o anterior, os resultados mostraram uma melhora de um ano para o outro, principalmente no que diz respeito aos servidores com mais de 10 minutos de erro.

Figura 5.2 - Gráfico com resultados de testes de sincronização de tempo via requisições HTTP para as coletas realizadas de 2010 a 2012.

Como o teste de qualidade de sincronização de tempo baseado no protocolo NTP só foi adicionado a partir da coleta de 2011, a Figura 5.3 contém os dados apenas a partir de então. Nele, observa-se claramente a diferença na precisão da medida, da ordem de milissegundos, e da qualidade da sincronização das máquinas que adotam o sistema, com mais de 90% dos casos sincronizados. Todavia, ele possui um abrangência de *sites* significativamente menor do que o teste via HTTP. Em 2012, para exemplificar, 4,14% dos *sites* responderam às requisições via NTP contra 93,78% que forneceram respostas via HTTP. Em termos absolutos, em 2011 532 *sites* responderam a esse tipo de requisição e em 2012, 586 responderam.

Desse gráfico a informação de que cerca de 7% dos *sites* que responderam às requisições NTP não estavam sincronizados, diferença de tempo entre 1 e 10 segundos, foi inusitada, uma vez que a utilização do serviço NTP deveria garantir a sincronia do relógio da máquina com o UTC. Uma possibilidade para a ocorrência desse fato é uma má configuração em algum dos servidores de referência do sistema NTP listados por padrão em sistemas operacionais diversos. Como, em geral, usuários NTP não se preocupam em modificar a configuração padrão disponibilizada, um erro em algum desses servidores se propagaria para múltiplos clientes.

Figura 5.3 - Gráfico com resultados de testes de sincronização de tempo via requisições NTP para as coletas realizadas em 2011 e 2012.

5.1.4 IPv6

Conforme descrito na Seção 6.2.3 do Apêndice 6 a verificação do suporte ao IPv6 representou o teste referente à servidores mais abrangente desta pesquisa. Além da verificação da existência de endereços IPv6 para os domínios pesquisados, foram também apurados as respostas a requisições a eles enviadas e o suporte ao protocolo provido pelos servidores DNS encarregados.

Mesmo com esse teste não sendo suficiente para garantir que a qualidade do acesso via IPv6 fosse igual à do IPv4, ele foi suficiente para mostrar a defasagem da implantação dessa nova tecnologia nos portais do governo. Em 2010 o teste não conseguiu registrar nenhum *site* com suporte, total ou parcial, ao IPv6. Em 2011, apenas 25 *sites* (0,2% do total no ano) apresentaram um mínimo de suporte a esse protocolo e, em 2012, 56 (0,4% do total no ano). A Tabela 5.2 exhibe a distribuição desses *sites*, que apresentaram um mínimo de suporte ao IPv6 nesses dois últimos anos, nas categorias discutidas na Seção 6.2.3 do Apêndice 6.

Tabela 5.2 - Resultados dos testes de suporte ao IPv6 realizados em 2011 e 2012.

	2011	2012
Possui endereço IPv6 em seu domínio principal	24	56
Possui endereço IPv6 em nomes de <i>sites</i> alternativos	6	3
O endereço responde a requisições HTTP GET (possui servidor Web)	3	32
O Endereço responde a requisições PING6	3	30
Os servidores DNS suportam IPv6	20	42
Os servidores DNS dos domínios alternativos suportam IPv6	5	2

É interessante notar que mesmo possuindo endereço IPv6, muitos dos *sites* não eram de fato acessíveis por meio desse protocolo, uma vez que não respondiam à requisições do tipo HTTP GET ao seu endereço IPv6. O fato de que eles não respondiam tais requisições, indica a inexistência de serviço Web ativo por meio do endereço. Dos 24 *sites* registrados em 2011 que possuíam endereços IPv6 em suas URLs principais, apenas 3 respondiam à requisições Web.

Além disso, nota-se a mudança na abordagem de nomes de domínio alternativos. Essa denominação foi adotada para designar domínios utilizados para testar o funcionamento do novo protocolo. No início da pesquisa TIC Web foi identificado que algumas empresas agregavam certos prefixos, como “www6” e “v6”, aos nomes de seus *sites* para verificar possíveis problemas decorrentes da utilização do IPv6 e, é esse o comportamento que está sendo verificado na segunda e sexta linhas da Tabela 5.2. A queda da aplicação de tais nomes supõe uma maturidade dos serviços providos via IPv6, uma vez que não necessitam mais de domínio exclusivo.

5.1.5 Tipo de servidor

O último atributo apresentado no artigo publicado na conferência WebMedia’12, escrito por Ganzeli et al. (2012), foi o tipo de servidor Web adotado pelo *sites*. A Figura 5.4 mostra o cenário obtido para as coletas realizadas nos anos de 2010 a 2012. Com ele, percebe-se que não ocorreram mudanças muito

significativas nesse período, principalmente no que se refere aos líderes do mercado, Apache e Microsoft *Internet Information Services* (Microsoft IIS). Contudo, é interessante observar que servidores alternativos, representados pela categoria outros, vem se tornando cada vez mais populares.

Figura 5.4 - Gráfico contendo a distribuição da porcentagem de sites entre os tipos de servidores registrados em 2010 a 2012.

A Figura 5.5 mostra estabilidade na localização dos servidores dos *sites* do governo brasileiro, indicando que pouco se foi feito no sentido de remover conteúdo possivelmente sigiloso, de servidores no exterior.

Figura 5.5 - Gráfico da porcentagem de sites localizados no Brasil e no exterior para as coletas executadas no ano de 2010 e 2012

5.2 Aplicação da metodologia

Seguindo sequência da metodologia definida no Capítulo 4, as seções a seguir descrevem a execução de cada um de seus passos utilizando, para tanto, o conjunto de dados relativo à coleta do *.gov.br* em 2012.

5.2.1 Adaptação de atributos

No presente projeto, a etapa de adaptação dos atributos teve início com a consolidação, em atributos de *sites*, dos dados relativos a páginas Web. Conforme apresentado no Apêndice 5, para cada uma das coletas, os registros são divididos em duas unidades de pesquisa, “página” e “*site*”. A primeira contém informações relativas às páginas encontradas, como a validação do código *Hypertext Markup Language* (HTML), e é identificada pela própria *Uniform Resource Locator* (URL) de acesso à página. A segunda, possui informações relativas aos *sites*, em geral comuns a todas as suas páginas, como sua localização geográfica e seu tipo de servidor Web e, ela é identificada pela URL do *site*.

Esse passo foi importante devido à dificuldade de se aplicar técnicas de aprendizado de máquina à totalidade dos registros dessa última unidade de pesquisa. Apenas para ilustrar a diferença de escala entre essas duas unidades, a base de *sites* obtida na coleta realizada em 2010 possuía um total de 11,9 mil registros, enquanto a de páginas, 6,3 milhões registros. As quantidades para os outros anos estão descrita na Tabela 5.1, porém as ordens de grandeza são equivalentes. Assim, os atributos descritos na Tabela 5.3 foram criados e adicionados à unidade de pesquisa “*site*”. A descrição dos indicadores originais se encontra no Apêndice 5.

Tabela 5.3- Atributos adicionados à unidade de pesquisa “site” devido à consolidação dos atributos referentes à unidade de pesquisa “páginas”

Atributo	Processo de aquisição
Quantidade de redirecionamentos encontrados	Contabilização de páginas com resposta HTTP Redirect (HTTP status = 3XX)
Quantidade de páginas HTTP OK encontradas	Contabilização de páginas com resposta HTTP OK (HTTP status = 2XX)
Quantidade de links para páginas HTML	Contabilização de páginas identificadas com o MIME “text/html”
Quantidade de links para documentos de texto puro	Contabilização de páginas identificadas com o MIME “text/plain”
Quantidade de links para aplicativos	Contabilização de páginas identificadas com o MIME “application”
Quantidade de páginas com acessibilidade nível A*	Contabilização do número de páginas que não apresentaram erros de prioridade 1 e pelo menos um erro de prioridade 2*
Número médio de erros de nível 1 na validação de acessibilidade*	Cálculo da média de erros de prioridade 1 por página*
Número médio de avisos de nível 1 na validação de acessibilidade*	Cálculo da média de avisos de prioridade 1 por página*
Quantidade de páginas com acessibilidade nível AA*	Contabilização do número de páginas que não apresentaram erros de prioridade 1 e 2 e pelo menos um erro de prioridade 3*
Número médio de erros de nível 2 na validação de acessibilidade*	Cálculo da média de erros de prioridade 2 por página*
Número médio de avisos de nível 2 na validação de acessibilidade*	Contabilização da média de avisos de prioridade 2 por página*
Quantidade de páginas com acessibilidade nível AAA*	Contabilização do número de páginas que não apresentaram erros de prioridade 1, 2 e 3*
Número médio de erros de nível 3 na validação de acessibilidade*	Cálculo da média de erros de prioridade 3 por página*
Número médio de avisos de nível 3 na validação de acessibilidade*	Cálculo da média de avisos de prioridade 3 por página*
Quantidade média de erros na validação HTML por página	Cálculo da média de erros HTML por página
Tamanho médio por página	Cálculo da média dos tamanhos das páginas
Profundidade média	Contabilização da média da profundidade das páginas em relação à raiz do site
Tamanho máximo de página	Identificação do tamanho da maior página do site

Atributo	Processo de aquisição
Tamanho mínimo de página	Identificação do tamanho da menor página do <i>site</i>
Desvio padrão do tamanho das páginas	Cálculo do desvio padrão dos tamanhos das páginas do <i>site</i>
Quantidade de páginas estáticas	Contabilização da quantidade de páginas estáticas segundo critério de identificação do WIRE
Quantidade de páginas dinâmicas	Contabilização da quantidade de páginas dinâmicas segundo critério de identificação do WIRE
Média das idades das páginas	Cálculo da média das idades das páginas
Idade mínima das páginas	Identificação idade da página mais recente do <i>site</i>
Idade máxima das páginas	Identificação idade da página mais antiga do <i>site</i>
Profundidade máxima das páginas	Identificação profundidade máxima das páginas
Profundidade média das páginas	Cálculo da profundidade média das páginas

** A descrição de níveis e prioridades de acessibilidade se encontra na Seção 2.3*

Além da consolidação dos dados provenientes das páginas, alguns dos atributos dos *sites* também tiveram de ser transformados ou removidos devido ao seu formato ou conteúdo serem inadequados ao processamento realizado pelas técnicas de aprendizado de máquina. Os seguintes atributos foram removidos graças à sua irrelevância ao processo de análise:

- Nome do *site*;
- Domínio;
- Endereço IPv4;
- Endereço IPv6; e
- Grupo em que se encontra na estrutura macroscópica da Web, conforme descritos na Figura 3.1.

Já os itens a seguir foram transformados em atributos numéricos, adequados ao processamento do aprendizado de máquina:

- **Tipo de Servidor:** dado que não existe relação de superioridade entre os tipos de servidores encontrados, o atributo foi separado em 4 valores independentes. Servidor apache, servidor microsoft_IIS, servidor zope e servidor nginx. A escolha dessas variáveis foi determinada pela popularidade desses tipos de servidores conforme mostrado nos resultados da Seção 5.1.5. Cada um dos atributos criados pode receber valores positivos ou negativos, dependendo do *site* registrado utilizar ou não o tipo de servidor respectivo ao atributo. O caso de nenhum desses atributos ser positivo, indica que o *site* utiliza outro tipo de servidor.
- **Localização geográfica:** Separada em 2 valores independentes. *Site* localizado no Brasil e localização não identificada. O funcionamento desses novos atributos booleanos é similar ao descrito para os novos atributos indicativos do tipo de servidor. Com essa mudança, a localização de servidores no exterior foi agrupada em um único conjunto de valores. Isso, aconteceu pois os classificadores de qualidade a serem gerados são voltados ao público brasileiro.

A Tabela 5.4 mostra o resultado final da adaptação dos atributos.

5.2.2 Seleção manual de atributos

Com a lista completa dos atributos disponíveis para análise pôde-se criar listas com tipos específicos de atributos, para se observar e analisar diferentes nuances dos resultados.

A Tabela 5.4 mostra os atributos resultantes da etapa anterior, bem como as diferentes seleções a serem estudadas. A primeira seleção corresponde a uma lista completa dos atributos disponíveis. A lista 2 corresponde à uma seleção de características voltadas a validação do código das páginas. A terceira, corresponde à

uma seleção de características voltadas a infraestrutura disponibilizada pelo servidor. E, por último, tem-se uma lista voltada à estrutura do *site*. Como diversos desses atributos foram retirados diretamente das unidades de pesquisa “*site*” e “*página*”, uma explicação mais detalhada pode ser encontrada nas seções 5.2 e 5.3 do Apêndice 5.

Tabela 5.4 - Descrição dos atributos a serem utilizados em cada um das listas selecionadas.

Descrição	Nome	Listas			
		1	2	3	4
Nome do <i>site</i>	sitio				
Número de páginas	n_pag	X			X
Tamanho total	total_size	X			X
Tamanho médio das páginas	avg_size	X			X
Tamanho máximo de página	max_pag_size	X			X
Tamanho mínimo de página	min_pag_size	X			X
Desvio padrão do tamanho das páginas	std_pag_size	X			X
<i>Site</i> localizado no Brasil	brazil	X		X	
Localização não identificada	non_indentified	X		X	
Servidor apache	apache	X		X	
Servidor microsoft_IIS	microsoft_IIS	X		X	
Servidor zope	zope	X		X	
Servidor nginx	nginx	X		X	

Descrição	Nome	Listas			
		1	2	3	4
Possui endereço ipv6 no domínio principal	ipv6_address	X		X	
Possui endereço ipv6 em domínio alternativo	alt_address	X		X	
O endereço responde a requisições HTTP GET via ipv6	reply_get	X		X	
O endereço responde a requisições PING6	reply_ping	X		X	
Os servidores DNS do domínio principal suportam ipv6	ns_ipv6_sup	X		X	
Os servidores DNS de algum domínio alternativo suportam ipv6	ns_alt_ipv6_sup	X		X	
Quantidade de páginas sem erros na validação de HTML	count_html_ok	X	X		
Número médio de erros na validação HTML	avg_html_errors	X	X		
Nível de acessibilidade do <i>site</i>	accessibility_lvl	X	X		
Quantidade de páginas com acessibilidade nível A	count_pag_A	X	X		
Número médio de erros de prioridade 1 na validação de acessibilidade	avg_p1_errors	X	X		
Número médio de avisos de prioridade 1 na validação de acessibilidade	avg_p1_warn	X	X		
Quantidade de páginas com acessibilidade nível AA	count_pag_AA	X	X		
Número médio de erros de prioridade 2 na validação de acessibilidade	avg_p2_errors	X	X		
Número médio de avisos de prioridade 2 na validação de acessibilidade	avg_p2_warn	X	X		
Quantidade de páginas com acessibilidade nível AAA	count_pag_AAA	X	X		
Número médio de erros de prioridade 3 na validação de acessibilidade	avg_p3_errors	X	X		
Número médio de avisos de prioridade 3 na validação de acessibilidade	avg_p3_warn	X	X		

Descrição	Nome	Listas			
		1	2	3	4
Diferença de tempo em relação ao UTC medida via requisições HTTP	http_offset	X		X	
Diferença de tempo em relação ao UTC medida via requisições NTP	ntp_offset	X		X	
Stratum do serviço NTP	ntp_stratum	X		X	
Tempo de resposta de uma requisição HTTP	http_delay	X		X	
Tempo de resposta via NTP	ntp_delay	X		X	
Quantidade de páginas estáticas	count_static	X			X
Quantidade de páginas dinâmicas	count_dynamic	X			X
Média das idades das páginas	avg_pag_age	X			X
Idade mínima das páginas	min_pag_age	X			X
Idade máxima das páginas	max_pag_age	X			X
Possui arquivo robots.txt	robots_txt	X			X
Grau de entrada	in_degree	X			X
Grau de saída	out_degree	X			X
Links internos	internal_links	X			X
Profundidade máxima das páginas	max_depth	X			X
Profundidade média das páginas	avg_depth	X			X
Quantidade de redirecionamentos encontrados	count_redirect	X			X
Quantidade de páginas HTTP OK encontradas	count_ok	X			X

Descrição	Nome	Listas			
		1	2	3	4
Quantidade de links para páginas HTML	count_text_html	X			X
Quantidade de links para documentos de texto puro	count_text_plain	X			X
Quantidade de links para aplicativos	count_app	X			X

5.2.3 Normalização de atributo

Com a adaptação dos atributos para valores numéricos, a etapa de normalização pôde ser executada. No caso do presente projeto, a linguagem de programação Octave foi utilizada para carregar as listas de dados, descritas na Tabela 5.4, e aplicar a fórmula 4-2 descrita na Seção 4.3. A Figura 5.6 contém o código utilizado.

Note que a utilização de matrizes nessa linguagem é bastante simplificada e, por isso, ela foi escolhida para o desenvolvimento das soluções aqui presentes.

Figura 5.6 - Código da função de normalização da matriz de atributos dos dados a serem analisados

```

1 function [X_norm, mu, sigma, valid_columns] = featureNormalize(X)
2
3 % FEATURENORMALIZE função para normalizar um uma matriz de atributos X
4 % FEATURENORMALIZE(X) retorna a matriz X normalizada, na qual
5 % a média dos valores de cada parâmetro é zero e o
6 % desvio padrão é 1.
7
8 mu = mean(X);
9 X_norm = bsxfun(@minus, X, mu);
10
11 sigma = std(X_norm);
12
13 X_norm = bsxfun(@rdivide, X_norm, sigma);
14
15 %
16 % caso o registro apresente alguma coluna nula, ele deve
17 % ser desconsiderado e as variáveis estatísticas são recalculadas
18 % após a remoção
19 %
20 valid_columns = sum(isnan(X_norm)) == 0;
21 X_norm = X_norm(:,valid_columns);
22
23 mu = mu(valid_columns);
24 sigma = sigma(:,valid_columns);
25
26 end

```

5.2.4 Redução dimensional

Como não é foco do presente trabalho o estudo da redução dimensional da base de dados, optou-se por não realizar comparações entre os diferentes métodos apresentados na Seção 4.4 e por aplicar apenas um deles. A Figura 5.7 a apresenta o código da função PCA em linguagem Octave utilizada para aplicar o método às bases de dados previamente normalizadas. Nesse código é utilizada a função *svd* pertencente à biblioteca padrão da linguagem para a implementação do algoritmo segundo os passos descritos na Seção 4.4.

Depois de executado o algoritmo, com as matrizes de autovalores e autovetores montadas, deve-se determinar a dimensionalidade com a qual se irá trabalhar. Como a matriz de autovetores está ordenada de acordo com a relevância da dimensão, os vetores a serem escolhidos seguem sua ordenação.

Figura 5.7 – Código da função de cálculo do PCA (Principal Component Analysis).

```

1 function [U, S] = pca(X)
2 % PCA executa o método Principal Component Analysis em um
3 % conjunto de dados X.
4 %
5 % [U, S] = pca(X) calcula os autovetores da matriz de
6 % covariâncias de X e retorna os autovetores U, e os
7 % autovalores na diagonal) em S
8 %
9
10 % Inicialização das variáveis
11 [m, n] = size(X);
12
13 U = zeros(n);
14 S = zeros(n);
15
16 % Cálculo da matriz de covariâncias
17 Sigma = X'*X/m;
18
19 % Cálculo dos autovetores e autovalores
20 [U,S,V] = svd(Sigma);
21
22 end

```

Como visto, não existe um consenso para a determinação do número adequado de vetores a serem utilizados, porém a matriz de autovalores pode ser entendida como uma medida da relevância de cada um dos autovetores. Assim, decidiu-se limitar a quantidade de dimensões quando a soma dos autovalores dos atributos selecionados atingisse 90% da soma total dos autovalores dos atributos pertencentes à lista em questão. Nesse processo, os atributos foram selecionados em ordem decrescente de relevância. A porcentagem de 90% foi escolhida por conseguir manter parte significativa da relevância total dos atributos e, ao mesmo tempo, ignorar atributos que venham causar distorções no processo.

A Figura 5.8 apresenta os resultados da seleção da dimensionalidade para as quatro bases de dados indicadas na Seção 5.2.2. Nesses gráficos, a linha vertical indica o limite de dimensões considerado. Lembrando que esse limite foi definido para que 90% do valor total da relevância dos atributos em questão fosse considerado no processo. Assim, os limites ficaram em 27, 9, 10 e 9 atributos respectivamente. A partir dessa figura, nota-se também que em todos os quatro casos os formatos dos gráficos são semelhantes, seguindo o modelo de cauda longa.

Nesse estágio, além da redução do número de dimensões, ocorre um ganho de qualidade dos dados à medida que removeu distorções que poderiam ter sido causadas pela adição de atributos irrelevantes. Isso é importante dado ser comum que esse tipo de distorção altere os resultados do reconhecimento de padrões.

Figura 5.8 – Gráficos indicando os autovalores para cada um dos autovetores encontrados. Neles, a linha vertical indica o atributo limite a ser utilizado. Assim, a cor verde indica os atributos que foram incluídos e a vermelha indicada aqueles descartados no processo de redução dimensional. Da esquerda para a direita e de cima para baixo, os gráficos correspondem às listas de atributos 1, 2, 3 e 4 indicadas na Seção 5.2.2.

5.2.5 Agrupamento de elementos

Para esta etapa os algoritmos de aprendizado computacional descritos na Seção 4.5 foram implementados também em linguagem Octave, com o objetivo de facilitar a utilização das bases de dados previamente modificadas, bem como de

agilizar a geração de gráficos para a etapa seguinte do processo. A Figura 5.9 mostra o código utilizado para a execução do algoritmo K-means.

Figura 5.9 – Código da função utilizada para executar o algoritmo K-means.

```

1 function [centroids, idx] = runkMeans(X, initial_centroids, max_iters)
2 % A função RUNKMEANS roda o algoritmo K-Means em uma matriz X,
3 % na qual cada uma de suas linhas representa um elemento e as colunas,
4 % seus respectivos parâmetros.
5 % initial_centroids corresponde à matriz com a posição inicial dos centroides.
6 % max_iters corresponde ao número máximo de iterações que o algoritmo realizará
7 % E, os valores retornados, são os centroides encontrados e um índice contendo a
8 % classificação de cada elemento de X
9
10 % Inicialização dos valores
11 [m n] = size(X);
12 K = size(initial_centroids, 1);
13 centroids = initial_centroids;
14 previous_centroids = centroids;
15 idx = zeros(m, 1);
16
17 % Execução do K-Means
18 for i=1:max_iters
19
20     % para cada elemento, encontra-se o centroid mais próximo
21     idx = findClosestCentroids(X, centroids);
22
23     % Dada a nova classificação, calculam-se os novos centroides
24     centroids = computeCentroids(X, idx, K);
25 endfor
26 end

```

Por ser mais complexo, o algoritmo EM foi implementado com base em uma biblioteca de código aberto para o reconhecimento de padrões em imagens desenvolvida em C++ de nome TerraLib¹⁰. O código desenvolvido está apresentado no Apêndice 1. Assim como o algoritmo K-means, o EM também é susceptível ao encontro de mínimos locais para a função de custo da solução. Assim, sua inicialização também é feita de modo aleatório, como indica o bloco composto pelas linhas de 28 à 48 de seu código. A múltipla execução do algoritmo também é realizada até que se encontre melhor solução. Contudo, como esse processamento é muito mais custoso do que o descrito pelo algoritmo anterior, dada sua intensa

¹⁰ Disponível em: <http://www.terralib.org/html/devel/html/index.html>. Acessado em: 18 de abril de 2014

manipulação de matrizes, apenas 10 tentativas foram executadas para cada caso, contra 1000 do K-means.

Para todas as listas descritas na Seção 5.2.2, ambos os algoritmos foram executados para números máximos de 3 e 5 grupos. A

Tabela 5.5 mostra a quantidade de sites alocados para cada um dos grupos. Nela, é interessante notar a discrepância entre os comportamentos dos algoritmos. Enquanto o K-means conseguiu delimitar elementos para cada um dos grupos criados, o EM, devido a sua característica de análise de curvas gaussianas, só conseguiu encontrar regras suficientes para diferenciar os elementos para a primeira lista de atributos. Isso invalida a utilização desse último algoritmo para as análises mais aprofundadas, uma vez que não é possível encontrar elementos que definam os grupos.

Observa-se também que para as listas de 1 a 3 os tamanhos dos grupos formados possuem uma característica similar. Nesses três casos, o maior grupo agrega a quase totalidade dos *sites* e o menor praticamente não possui representantes. A quarta lista se mostra um pouco diferente, pois o menor dos grupos possui quantidade de *sites* mais relevante, mas ela também reforça a composição com um grupo majoritário. Ambos os casos são indicativos de que poucos *sites* conseguem se destacar independentemente da lista de atributos escolhida. Tudo isso é válido para a execução do K-means com 3 e 5 grupos.

Um terceiro ponto passível de comparação é o das diferenças entre os resultados para 3 e 5 grupos. Em ambos a existência de um grupo com quantidade de elementos muito superior aos outros, sugere que poucos são os *sites* que conseguem se diferenciar com relação às características testadas. A lista de número 4, referente às características estruturais do *site* se diferencia um pouco por apresentar uma distribuição um pouco mais equilibrada entre os grupos encontrados. Contudo apenas com essas informações, conclusões mais aprofundadas sobre o nível de qualidade selecionado por tais agrupamentos são difíceis de serem feitas.

Tabela 5.5 – Relação dos números de sites classificados em cada grupo nas execuções dos algoritmos K-means e EM para os números máximo de 3 e 5 grupo e para cada uma das listas descritas na Seção 5.2.2.

Algoritmo	N Grupos	Listas			
		1	2	3	4
K-means	3	1076	626	3552	10675
		13022	195	10566	2295
		54	13331	34	1182
	5	54	195	40	847
		59	91	54	2045
		10287	2993	9987	996
		1530	10714	590	7359
		2222	159	3481	2905
	3	947	0	0	0
		0	0	0	0
		13205	14152	14152	14152
EM	3	936	0	0	0
		0	0	0	0
		0	14152	0	0
	5	0	0	14152	0
		13216	0	0	14152

5.2.6 Visualização gráfica

Na etapa de visualização gráfica utilizou-se o PCA, descrito na seção 4.4, para selecionar os atributos mais relevantes e assim traçar gráficos de 2 e 3 dimensões. Para traça-los, cada um dos grupos encontrados durante a execução de técnicas de aprendizado computacional não supervisionado foi designado com uma cor. Para facilitar a visualização e deixar os gráficos mais limpos, apenas uma amostra aleatória de 1000 dos 14152 elementos foi desenhada nos gráficos. As figuras do Apêndice 2 mostram os resultados obtidos para cada uma das quatro listas de atributos analisadas.

Desses gráficos, nota-se a alta concentração da maioria dos pontos, comprovando a similaridade das características dos sites analisados, conforme os dados da

Tabela 5.5 sugeriram. Contudo, devido às características do método de aprendizado computacional baseado em distâncias euclidianas presentes no algoritmo K-means, certa diferenciação é possível. Nesse sentido, os gráficos referentes ao algoritmo EM ficam em sua maioria monocromáticos devido a ele não ter conseguido separar os dados em múltiplos grupos.

Nas Figuras de 2.1 a 2.4 do Apêndice 2 têm-se as visualizações dos resultados para a lista 1 de atributos. Conforme descrito na Seção 5.2.2, essa é a lista mais genérica, contendo todos os atributos adquiridos na presente pesquisa. A Figura Apêndice 2.3.a e a Figura Apêndice 2.4.a são exemplos de como podem ser criadas divisões entre a qualidade percebida dos *sites*. Tais figuras foram copiadas respectivamente nas Figuras 5.10 e 5.11 para facilitar a visualização. Apesar de não haver limiares bem definidos entre os dados, esse gráficos mostram uma clara divisão espacial entre os grupos os elementos dos 3 principais grupos encontrados. Ainda assim, uma análise mais aprofundada de quais elementos estão presentes em cada um dos grupos seria necessária para que se pudesse caracterizá-los adequadamente.

Figura 5.10 - Gráficos 2D referentes à lista 1 para a execução, com máximo de 5 grupos, do algoritmo K-means.

Figura 5.11 - Gráficos 3D referentes à lista 1 para a execução, com máximo de 5 grupos, do algoritmo K-means.

Resultados em 3D referentes à listal utilizado K-means

Nas Figuras de 2.5 a 2.8 também do Apêndice 2, podem ser observados os gráficos referentes à segunda lista de atributos, referentes à qualidade do código das páginas pertencentes aos *sites* do escopo pesquisado. Delas, nota-se um espalhamento dos elementos no espaço ainda mais precário. Mesmo o algoritmo K-means executado para cinco centroides conseguiu delimitar apenas dois grupos mais significativos de elementos. Ao relacionar tais resultados com as informações providas pela Figura 5.8 percebem-se os efeitos do eixo principal ser significativamente mais relevante que o restante.

As distribuições mais peculiares podem ser observadas nos gráficos das Figuras de 2.9 a 2.12, do Apêndice 2, referentes aos resultados da terceira lista, com atributos sobre os servidores dos *sites* encontrados. Nelas, a despeito de alguns elementos discrepantes, todos os pontos estão bem concentrados num mesmo local. Neste caso, a opção de 5 grupos, Figura Apêndice 2.10.a, se mostrou mais adequada por conseguir isolar os casos extremos e ainda criar certa diferenciação entre os *sites* do grupo principal.

Os últimos resultados, figuras 2.13 a 2.16 do Apêndice 2, são relativos à lista de número quatro, com atributos referentes aos dados estruturais dos *sites*. Esses

resultados são particularmente interessantes por apresentarem uma distinção mais clara entre os grupos formados, ao menos para a execução do algoritmo *K-means*. Na Figura Apêndice 2.13.a, copiada na Figura 5.12, pode-se notar o formato de um bumerangue, no qual cada asa possui um grupo de elementos e o cotovelo, um terceiro grupo.

Figura 5.12 - Gráficos 2D referentes à lista 4, com atributos relacionados à estrutura dos sites, para a execução, com máximo de 3 grupos, do algoritmo *K-means*;

5.2.7 Análise de indicadores

Como visto na seção anterior, apesar de já ter sido obtida uma forma consistente para a definição e predição de qual grupo um determinado *site* pertence, sem uma análise mais aprofundada, pouco se pôde concluir sobre as características que definem esses grupos. Como a projeção dos dados via algoritmo PCA impede a interpretação semântica dos grupos formados, o retorno à base original teve de ser feito.

Para identificar as características dos grupos, uma opção é a utilização de FSAs, como os descritos na seção 4.4. Com a classificação em grupos já realizada,

pôde-se utilizar a função *Feature Subsection Selection* (FSS)¹¹ da Linguagem Octave baseada em mRMR, um método de aprendizado computacional supervisionado descrito na Seção 4.6. As tabelas presentes no Apêndice 3 contém os resultados da ordenação dos atributos executada para cada uma das listas definidas na Seção 5.2.2.

A Primeira coisa que se nota ao analisá-las é a similaridade das colunas referentes ao algoritmo EM. Isso se deve ao fato desse algoritmo não ter conseguido separar, na maior parte dos casos, os elementos em múltiplos grupos.

Na Tabela Apêndice 3.1 referente à ordenação da lista completa de atributos, é interessante notar que os primeiros valores tanto para 3 quanto para 5 grupos nos resultados do algoritmo K-means estão relacionados à infraestrutura dos servidores e à estrutura dos *sites*. Outra observação que se pode fazer é que nas outras listas esses atributos não se sobressaltaram devido à combinação com outros valores.

Com relação a esses resultados, é fácil perceber a utilidade de métodos automatizados para a realização não apenas da busca por padrões nos dados, mas também para a definição da relevância dos atributos. Contudo, como soluções automáticas estão longe da perfeição, caso se tenha conhecimento prévio da importância de determinado atributo, adaptações podem ser feitas forçando um maior peso a esse atributo.

Tanto para caracterizar os grupos formados quanto para mostrar alguns exemplos de *sites*, as tabelas do Apêndice 4 foram criadas. As tabelas de 4.5 a 4.8 desse apêndice são compilações das listas dos atributos mais relevantes, segundo ordenação descrita no Apêndice 3, para representantes aleatórios de cada um dos grupos encontrados na execução do algoritmo K-means com máximo de 3 grupos para cada uma das 4 listas de atributos selecionadas na Seção 5.2.2. Apesar de resultados importantes para o presente trabalho, as tabelas presentes nos Apêndices 3 e 4 tiveram de ser separadas do corpo principal do texto para melhorar sua organização.

A partir dessas tabelas, foram criadas definições para cada um dos grupos encontrados e, essas puderam ser posteriormente utilizadas em um processo de

¹¹ Disponível em <http://octave.sourceforge.net/nan/function/fss.html>. Acessado em: 18 de abril de 2014

classificação de *sites*. A Tabela 5.6 contém um exemplo dos resultados para a execução do algoritmo K-means para 3 grupos.

Tabela 5.6 – Definição dos grupos criados para cada uma das listas de atributos especificadas na Seção 5.2.2 para a execução do método K-means com limite de 3 grupos.

Grupo	Lista			
	1	2	3	4
1	<i>Sites sem suporte a IPv6 e amplo número de páginas, em sua maioria válidas</i>	<i>Sites constituídos por páginas com muitos erros na validação de acessibilidade e HTML</i>	<i>Sites sem suporte à IPv6 que utilizam o servidor web Microsoft IIS</i>	<i>Sites com reduzido número de páginas e em sua maioria recentemente atualizadas</i>
2	<i>Sites sem suporte à IPv6 com reduzido número de páginas, em sua maioria válidas</i>	<i>Sites completamente acessíveis em algum dos níveis do E-MAG</i>	<i>Sites sem suporte à IPv6 que não utilizam servidor web Microsoft IIS</i>	<i>Sites com maioria de páginas desatualizadas</i>
3	<i>Sites com suporte a IPv6</i>	<i>Sites constituídos por algumas páginas acessíveis ou com reduzido números de erros de validação</i>	<i>Sites que respondem à requisições HTTP via IPv6</i>	<i>Sites com amplo número de páginas e em sua maioria recentemente atualizadas</i>

Além disso, para classificação e comparação entre os *sites* é interessante que se defina à quais desses grupos um *site* com alto padrão de qualidade deve se enquadrar. Segundo as definições das características para que um determinado *site* melhore sua experiência de uso, discutidas no Capítulo 2, a Tabela 5.7 pôde ser montada. Nela foram também incluídos os tamanhos de cada um dos grupos indicados, segundo valores presentes na

Tabela 5.5. Nota-se que, conforme visto na Seção 5.1, os *sites* da Web Governamental Brasileira se concentram nas categorias com baixos níveis de qualidade. Assim, o nivelamento dos grupos também poderia ter sido feito com base nessa informação previa do comportamento geral da base de forma relativamente confiável.

Tabela 5.7 - Classificação e quantidade de sites dos grupos criados para cada uma das listas de atributos especificadas na Seção 5.2.2 para a execução do método K-means com limite de 3 grupos.

Grupo	Lista							
	1		2		3		4	
	Nível	Qtd Sites	Nível	Qtd Sites	Nível	Qtd Sites	Nível	Qtd Sites
1	Regular	1076	Ruim	626	Ruim	3552	Bom	10675
2	Regular	13022	Bom	195	Regular	10566	Ruim	2295
3	Bom	54	Regular	13331	Bom	34	Bom	1182

Apesar de somente o caso da execução do método K-means com número máximo de 3 grupos ter sido detalhado, os demais casos podem receber o mesmo tratamento. Ao final, pode-se avaliar também a composição dos resultados para múltiplos casos na avaliação de *sites*.

5.2.8 Generalização dos classificadores

Com o intuito de aplicar na prática os resultados obtidos para a classificação dos *sites*, as tabelas de 4.1 a 4.4 do Apêndice 4 foram criadas. Elas são compilações das listas dos atributos mais relevantes, segundo ordenação descrita no Apêndice 3, para os 20 *sites* mais populares do governo brasileiro. A

popularidade do *site* foi originada da lista do primeiro milhão de *sites* mais populares que é disponibilizada pela empresa Alexa¹² obtida no dia 23 de novembro de 2013.

Seguindo o processo descrito na Seção 4.8, a Tabela 5.8 mostra como seriam classificadas as experiências de uso dos 5 *sites* mais populares do governo brasileiro, segundo a lista divulgada no *site Alexa.com* no dia 23 de novembro de 2013, caso os critérios obtidos da Web Governamental Brasileira fossem aplicados em sua avaliação.

Tabela 5.8 – Resultados das análises dos 5 sites mais populares da Internet Governamental Brasileira segundo critérios relacionados à percepção de qualidade do usuário

Site	Lista			
	1	2	3	4
caixa.gov.br	Regular	Ruim	Ruim	Bom
fazenda.gov.br	Regular	Regular	Ruim	Bom
pr.gov.br	Regular	Regular	Regular	Ruim
mec.gov.br	Regular	Regular	Regular	Bom
mg.gov.br	Regular	Ruim	Regular	Bom

Nota-se que, apesar da lista 1 de atributos, composta por todas as variáveis obtidas, ter classificado tais *sites* como regulares, sua composição com os resultados de outras listas, com temas específicos, os nivelou de forma diferente.

¹² Disponível em: <http://www.alexa.com/>. Acessado em: 18 de abril de 2014

6 CONSIDERAÇÕES FINAIS E CONCLUSÕES

O presente trabalho definiu e exemplificou metodologia para a análise da qualidade de web *sites*. A principal motivação para o seu desenvolvimento veio do TIC Web, um projeto que envolve diversas áreas de pesquisa relacionadas às tecnologias de criação e avaliação de um *site* Web. Seus resultados, embora abrangentes, careciam de ferramental analítico mais aprofundado. Nesse sentido, foi proposta uma extensão dessa pesquisa com a utilização mais extensiva de ferramentas de aprendizado de máquina, como meio para individualizar as análises e possibilitar a identificação de grupos de *sites* que compartilhem certas características de qualidade.

Como o termo “qualidade” carrega diversas interpretações, definiu-se aqui um conjunto de atributos e métricas para representá-lo. Cada um desses atributos foi contextualizado no cenário do projeto TIC Web e sua relevância foi defendida com base não apenas na importância dos padrões abertos para a Internet, mas também em uma vasta literatura acadêmica. Também, para fundamentar teoricamente o presente trabalho, foram adicionados diversos exemplos de pesquisas na área, mostrando e comparando seus principais indicadores e resultados. Com isso, foi possível melhor compreender as necessidades do projeto e compilar um conjunto bastante interessante de sugestões de indicadores a serem incluídos, como o índice de acessibilidade WAB. A descrição das ferramentas de coleta empregadas foi incluída tanto para mostrar o processo de aquisição dos dados quanto para justificar suas limitações.

Especificou-se, então, a metodologia de análise, mantendo-a coerente com outras metodologias de análise baseadas em aprendizado de máquina, como a descrita no artigo de Bauer & Scharl (2000), porém, adequando-a às especificidades do projeto. Como exemplo, cita-se a necessidade da consolidação dos atributos relacionados a páginas Web em atributos de *sites*. Essa especificação tentou apresentar de forma clara e detalhada as etapas e algoritmos necessários para sua aplicação em determinado conjunto de *sites*.

Para uma melhor compreensão dos indicadores e da situação atual dos *sites*, na Seção 5.1 foram apresentados e discutidos os principais resultados obtidos pela

execução das coletas do projeto TIC WEB sobre a Web Governamental Brasileira no período de 2010 a 2012. Tais resultados explicitaram diversos problemas relacionados aos indicadores de qualidade aplicados, principalmente com relação à porcentagem de páginas conformes com os padrões de acessibilidade, e ajudaram a melhorar as definições dos agrupamentos encontrados durante o estudo de caso.

Com relação aos resultados práticos do presente projeto, aplicou-se a metodologia especificada no Capítulo 4 à base de dados relativa à análise dos *sites* governamentais realizada pelo TIC Web em 2012. Nesse processo conseguiu-se aplicar e comparar os métodos K-means e EM, bem como utilizar os algoritmos de redução dimensional tanto para aprimorar a qualidade da base de dados antes da execução dos algoritmos de aglomeração, quanto para selecionar os atributos mais relevantes para a geração de gráficos e categorização de agrupamentos encontrados. Ao final do processo, mostrou-se também um exemplo da classificação de *sites* segundo sua qualidade.

A comparação dos métodos de aprendizado computacional aplicados foi interessante por mostrar que o algoritmo K-means, mesmo sendo mais simples conseguiu melhor dividir os *sites* em grupos distintos, possibilitando a utilização dos resultados para a classificação da qualidade de outros *sites*.

De forma geral, o este trabalho conseguiu atingir as contribuições propostas, principalmente com definição e aplicação da metodologia para a análise de domínios de *sites* e posterior classificação de *sites* segundo sua qualidade. Conforme pressuposto, ela foi definida de forma clara para que pudesse ser replicada e comparada por outros no futuro. Ainda mostrou como estender as análises sob os conjuntos de dados adquiridos pelo TIC Web e, com o estudo de caso, validou a metodologia.

Por tudo isso, o presente trabalho conseguiu abrir grande margem para o desenvolvimento de trabalhos futuros. Algumas opções óbvias seriam: a comparação da eficiência de modificações aos algoritmos de aglomeração selecionados ou mesmo de novos algoritmos; a inclusão de novos atributos à base de dados; o teste de novas listas de atributos; e, a construção de aplicação capaz de utilizar os resultados das análises para a classificação de *sites*. Com relação à extração de atributos, seria interessante também a avaliação do uso e qualidade de imagens e vídeos, uma vez que esses vêm se tornando cada vez mais frequentes.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- Bach, C. F., Ferreira, S. B. L., Silveira, D. S., & Nunes, R. R. (2009). Diretrizes de acessibilidade: uma abordagem comparativa entre WCAG e E-MAG. *Revista Eletrônica de Sistemas de Informação*, ISSN 1677-3071, 8(1). doi:10.5329
- Bailey, P., Craswell, N., & Hawking, D. (2003). Engineering a multi-purpose test collection for Web retrieval experiments. *Information Processing & Management*, 39(6), 853–871. doi:10.1016/S0306-4573(02)00084-5
- Bauer, C., & Scharl, A. (2000). Quantitative evaluation of Web site content and structure. *Internet Research*, 10(1), 31–44. doi:10.1108/10662240010312138
- Cafarella, M., & Cutting, D. (2004). Building Nutch. *Queue*, 2(2), 54–61. doi:10.1145/988392.988408
- Caldwell, B., Cooper, M., Reid, L. G., & Vanderheiden, G. (2006). Understanding WCAG 2.0 W3C Working Draft. *W3C World Wide Web Consortium*. Acessado : 20 de abril de 2014, de <http://www.w3.org/TR/UNDERSTANDING-WCAG20/>
- Canali, D., Cova, M., & Vigna, G. (2011). Prophiler : A Fast Filter for the Large-Scale Detection of Malicious Web Pages Categories and Subject Descriptors. In *Proceedings of the 20th international conference on World Wide Web* (p. 197–206). Hyderabad, India: ACM. doi:10.1145/1963405.1963436
- Castillo, C. (2005). Effective web crawling. *ACM SIGIR Forum*, 39(1), 55–56. doi:10.1145/1067268.1067287
- Castillo, C., Starosta, B., & Sydow, M. (2007). Crawl.pl: Measuring Statistical and Structural Properties of the Polish Web. *Studia Informatica*, 1(8), 43–73.
- Chen, S., Hong, D., & Shen, V. Y. (2005). An Experimental Study on Validation Problems with Existing HTML Webpages. In *Proc. International Conference on Internet Computing ICOMP 2005* (p. 373–379).
- Cova, M., Kruegel, C., & Vigna, G. (2010). Detection and analysis of drive-by-download attacks and malicious JavaScript code. *Proceedings of the 19th International Conference on World Wide Web - WWW '10*, 281. doi:10.1145/1772690.1772720
- Dardailler, D. (2007). W3C and Open Standard. Acessado : 20 de abril de 2014, de <http://www.w3.org/2005/09/dd-osd.html>
- Deering, S., & Hinden, R. (1998). RFC 2460: Internet Protocol, Version 6 (IPv6) Specification. *RFC 2460*. IETF. doi:10.1109/MSP.2008.65
- Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1), 1–38.

- Edward, T. O., Lavoie, B. F., & Patrick, D. (2001). Web Characterization Project. *Journal of Library Administration*, 34(3-4), 359–374. doi:10.1300/J111v34n03_17
- Eppler, M. J., & Muenzenmayer, P. (2002). Measuring Information Quality in The Web Context: A survey of State-of-the-Art Instruments and an Application Methodology. In *Proceedings of the Seventh International Conference of Information Quality* (p. 187–196).
- Fielding, R., Irvine, U., Gettys, J., Compaq/W3C, Mogul, J., Compaq, ... Berners-Lee, T. (1999). RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1. IETF.
- Figueiredo, M. A. T., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3), 1–16.
- Freire, A. P., Castro, M. de, & Fortes, R. P. de M. (2009). Accessibility of Brazilian state government websites: a quantitative analysis between 1996 and 2007. *Revista de Administração Pública*, 43(2), 395–414. doi:10.1590/S0034-76122009000200006
- Ganzeli, H. D. S., Moreiras, A. M., & Bressan, G. (2012). ICT Web : Analysis of the Brazilian Governmental Web. In *WebMedia '12 Proceedings of the 18th Brazilian symposium on Multimedia and the web* (p. 383–386). São Paulo - SP, Brazil: ACM New York, NY, USA. doi:10.1145/2382636.2382715
- Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1), 100–108.
- Henry, S. L. (2006). Introduction to Web Accessibility. Acessado : 20 de abril de 2014, de <http://www.w3.org/WAI/intro/accessibility.php>
- Hull, L. (2004). Accessibility: it's not just for disabilities any more. *ACM Interactions*, 11(2), 36–41. doi:10.1145/971258.971270
- Jain, A. K., & Dubes, R. C. (1988). *Algorithms for Clustering Data*. (B. Martine, Org.). New Jersey: Prentice Hall.
- Jolliffe, I. (2005). *Principal component analysis* (Second edi.). New York, New York, USA: Springer-Verlag New York, Inc.
- Katerattanakul, P., & Siau, K. (1999). Measuring Information Quality of Web Sites: Development of an Instrument. In *Proceedings of the 20th international conference on Information Systems* (p. 279–285). Charlotte, North Carolina, USA: Association for Information Systems. doi:10.1145/352925.352951
- Kira, K., & Rendell, L. (1992). A Practical Approach to Feature Selection. In *Proc. of the 9th Int. Conf. on Machine Learning* (p. 249–256). Aberdeen, Scotland: Morgan Kaufmann.

- Lawrence, S., & Giles, C. L. (2000). Accessibility of information on the web. *ACM Intelligence*, 11(1), 32–39.
- Liu, H., & Setiono, R. (1996). A Probabilistic Approach to Feature Selection: a Filter Solution. In *Proc. of the 13th Int. Conf. on Machine Learning* (p. 319–327). Morgan Kaufmann.
- Mardia, K. V., Kent, J. T., & Bibby, J. M. (1995). *Multivariate Analysis (Probability and Mathematical Statistics)* (First Edit., p. 518). Academic Press.
- Mendes, E., Mosley, N., & Counsell, S. (2002). Comparison of Web size measures for predicting Web design and authoring effort. In *Software, IEE Proceedings* (Vol. 149, p. 86–92). doi:10.1049/ip-sen:20020337
- Ministério do planejamento. (2012). Padrões de Interoperabilidade de Governo Eletrônico. *Ministério do planejamento, orçamento e gestão*. Acessado : 22 de maio de 2012, de <http://eping.governoeletronico.gov.br/>
- Ministério do Planejamento, & Secretaria de Logística e Tecnologia da Informação. (2011). *Modelo de Acessibilidade em Governo Eletrônico* (p. 69). Brasília: MP, SLTI, 2011.
- Mitchell, T. M. (1997). *Machine Learning*. (N. Je. Traub, Joseph F and Grosz, Barbara J and Lampson, Butler W and Nilsson, Org.) (p. 432). McGraw-Hill.
- Modesto, M., Junior, Á. P., & Ziviani, N. (2005). Um novo retrato da web brasileira. In *XXV Congresso da Sociedade Brasileira de computação* (p. 2005–2017).
- Mohr, G., Stack, M., Ranitovic, I., Avery, D., & Kimpton, M. (2004). An Introduction to Heritrix: An Open Source Archival Quality Web Crawler. In *4th International Web Archiving Workshop*.
- Molina, L., Belanche, L., & Nebot, À. (2002). Feature selection algorithms: A survey and experimental evaluation. In *IEEE International Conference on Data Mining* (p. 306 – 313). doi:10.1109/ICDM.2002.1183917
- Muller, A., & Schwarz, M. (2011). HTML Validation of Context-Free Languages. In M. Hofmann (Org.), *Foundations of Software Science and Computational Structures 14th International Conference, FOSSACS 2011* (p. 426–440). Springer Berlin Heidelberg. doi:10.1007/978-3-642-19805-2_29
- Ofuonye, E., Beatty, P., Dick, S., & Miller, J. (2010). Prevalence and classification of web page defects. *Online Information Review*, 34(1), 160–174. doi:10.1108/14684521011024182
- Palmer, J. W. (2002). Performance Metrics. *Information Systems Research*, 13(2), 151–167.

- Parmanto, B., & Zeng, X. (2005). Metric for Web accessibility evaluation. *Journal of the American Society for Information Science and Technology*, 56(13), 1394–1404. doi:10.1002/asi.20233
- Peng, H., Long, F., & Ding, C. (2005). Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8), 1226–1238. doi:10.1109/TPAMI.2005.159
- Pudil, P., & Hovovicova, J. (1998). Novel methods for feature subset selection with respect to problem knowledge. *Feature Extraction, Construction and Selection*, 453, 101–116. doi:10.1007/978-1-4615-5725-8_7
- Recommendation, I. (1994). 200 (1994)| ISO/IEC 7498-1: 1994. *Information Technology--Open Systems Interconnection--Basic Reference Model: The Basic Model*.
- Rose, G. M., & Straub, D. W. (2001). The Effect of Download Time on Consumer Attitude Toward the e-Service Retailer. *E-Service Journal*, 1(1), 55–76. doi:10.1353/esj.2001.0005
- Rybaczyk, P. (2005). *Expert Network Time Protocol: An Experience in Time with NTP*. New York, New York, USA: Apress.
- Savoia, A. (2001). Web Page Response Time. *Software Testing and Quality Engineering Magazine*, (August), 48–53.
- Sloan, D., Gregor, P., Rowan, M., & Booth, P. (2000). Accessible accessibility. In *Proceedings on the 2000 conference on Universal Usability - CUU '00* (p. 96–101). New York, New York, USA: ACM Press. doi:10.1145/355460.355480
- Tanenbaum, A. S. (2003). *Computer networks* (4th editio., p. 891). Prentice Hall PTR.
- Tolosa, G., Bordignon, F., Baeza Yates, R., & Castillo, C. (2007). Characterization of the Argentinian Web. *Cybermetrics: International Journal of Scientometrics, Informetrics and Bibliometrics*, 11(1).
- Yoo, B., & Donthu, N. (2001). Developing a Scale to Measure the Perceived Quality of An Internet Shopping Site (SITEQUAL). *Quarterly Journal of Electronic Commerce*, 2(1), 31–47.

APÊNDICE 1 CÓDIGO DA FUNÇÃO EMCLASS

Código da função EMClass responsável pela execução do algoritmo de classificação *Expectation Maximization* (EM) descrito na Seção 4.5.

```

1 function [means, covars, P, idx, variation] = EMclass (X, M)
2 % A função EMclass roda o algoritmo Expectation-Maximization em uma matriz X,
3 % na qual cada uma de suas linhas representa um elemento e as colunas,
4 % seus respectivos parâmetros.
5 % E, M corresponde ao número de classes a ser encontrados.
6 % os valores retornados são correspondem respectivamente à matriz de médias
7 % para cada um dos grupos, suas matrizes de covariâncias, um vetor com a
8 % probabilidade de cada grupo, um índice com a classificação de cada elemento
9 % após a execução de cada algoritmo e um valor de variação correspondente ao
10 % custo obtido para a solução
11
12 % inicialização das variáveis e constantes
13 total_attributes = size (X ,2);
14 MAX_ITERATIONS = 10;
15 SHIFT_THRESHOLD = .2;
16 N = size(X,1);
17 START_COVAR = 1000;
18 variations = zeros (M, 1);
19 covars = zeros (total_attributes, total_attributes, M);
20 old_covars = zeros (total_attributes, total_attributes, M);
21 means = zeros (total_attributes, M);
22 old_means = zeros (total_attributes, M);
23 original_X = X;
24 P = zeros (M, 1);
25 P_C_x = zeros (N, M);
26 original_P_C_x = zeros (N, M);
27
28 % Inicialização das matrizes de covariâncias, médias e Probabilidades
29 tmp_covar = diag (ones (total_attributes, 1) .* START_COVAR);
30
31 for j = 1:M
32     tmp_mean = rand(total_attributes,1) ;
33
34     means (:, j) = tmp_mean;
35     old_means (:, j) = tmp_mean;
36     covars (:, :, j) = tmp_covar;
37     old_covars (:, :, j) = tmp_covar;
38
39     P (j) = 1/M;
40
41 endfor
42
43
44 % início da estimativa dos valores
45 num_iterations = 0;
46 stop_condition = 0;
47
48 while stop_condition == 0

```

```

49 % Cálculo da inversa e determinante da matriz covar para agilizar o processamento
50 covars_inv = zeros (total_attributes, total_attributes, M);
51 covars_det = zeros(M,1);
52 for j = 1:M
53     covars_det(j) = det (covars (:, :, j));
54
55     if (covars_det(j) == 0)
56         covars (:, :, j) = tmp_covar;
57         covars_det (j) = det (covars (:, :, j));
58     endif
59     covars_inv (:, :, j) = inv (covars (:, :, j));
60
61 endfor
62
63 % Cálculo do P(C|x)
64 for j = 1:M
65     for k = 1:N
66
67         denom = 0;
68         for k1 = 1:M
69             tmp_matrix = X (k) - means (:, k1)';
70             tmp_prod = tmp_matrix * covars_inv (:, :, k1) * tmp_matrix';
71             denom += (covars_det(k1) ^ -.5) * exp (-.5 * tmp_prod) * P (k1);
72         endfor
73
74         tmp_matrix = X (k) - means (:, j)';
75         tmp_prod = tmp_matrix * covars_inv(:,j) * tmp_matrix';
76
77         P_C_x (k, j) = (covars_det(j) ^ -.5) * exp(-.5 * tmp_prod) * P (j) / denom;
78
79     endfor
80 endfor
81
82 % Atualização das médias, covariancias e probabilidades
83 for j = 1:M
84     tmp_denom = 0;
85     tmp_num_mean = zeros (size (total_attributes, 1));
86     tmp_num_covar = zeros (size (total_attributes, total_attributes));
87     tmp_sub = 0;
88
89     for k = 1:N
90         % soma para ser utilizada no numerador da média
91         tmp_num_mean = tmp_num_mean + (P_C_x (k, j) .* X (k))';
92
93         % soma para ambos os denominadores
94         tmp_denom += P_C_x (k, j);
95
96         % soma para o numerador da covariancia

```



```

97     tmp_matrix = X (k) - means (:,j)';
98     tmp_prod = tmp_matrix' * tmp_matrix;
99     tmp_num_covar = tmp_num_covar + P_C_x (k, j) .* tmp_prod;
100
101   endfor
102
103   means(:, j) = (1 / tmp_denom) * tmp_num_mean;
104   covars(:, :, j) = (1 / tmp_denom) * tmp_num_covar;
105   P(j) = tmp_denom / N;
106
107   tmp_sub = means (:, j) - old_means (:, j);
108   tmp_prod = tmp_sub' * tmp_sub;
109   variations (j) = sqrt (tmp_prod);
110
111   endfor
112
113   % caso a variação entre dois aglomerados seja muito pequena,
114   % um deles será reinicializado com valores aleatórios
115   shift = rand (total_attributes,1);
116   for j = 1:M
117     for j1 = (j+1):M
118
119       tmp_sub = means (:, j) - means (:, j1);
120       tmp_prod = tmp_sub' * tmp_sub;
121
122       if sqrt(tmp_prod) < SHIFT_THRESHOLD
123         shift = rand (total_attributes,1);
124         means(:, j1) = shift;
125         covars(:, :, j1) = tmp_covar;
126       endif
127
128     endfor
129   endfor
130
131   variation = max(variations);
132
133   old_means = means;
134   old_covars = covars;
135
136   if (++num_iterations >= MAX_ITERATIONS)
137     stop_condition = 1;
138   endif
139
140 endwhile
141
142
143
144 % Classificação dos elementos de acordo com as classes encontradas

```


APÊNDICE 2 GRÁFICOS RESULTANTES DA EXECUÇÃO DOS ALGORITMOS K-MEANS E EXPECTATION-MAXIMIZATION

Figura Apêndice 2.1 – Gráficos 2D referentes à lista 1 para a execução com máximo de 3 grupos. a. Gráfico o algoritmo K-means; b. Gráfico 3D para o algoritmo EM.

Resultados em 2D referentes à lista1 utilizado K-means

a

Resultados em 2D referentes à lista1 utilizado EM

b

Figura Apêndice 2.2 – Gráficos 3D referentes à lista 1 para a execução com máximo de 3 grupos. a. Gráfico 3D para o algoritmo K-means; b. Gráfico 3D para o algoritmo EM.

Resultados em 3D referentes à listal utilizado K-means

a

Resultados em 3D referentes à listal utilizado EM

b

Figura Apêndice 2.3 – Gráficos 2D referentes à lista 1 para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 2D referentes à lista1 utilizado K-means

a

Resultados em 2D referentes à lista1 utilizado EM

b

Figura Apêndice 2.4 – Gráficos 3D referentes à lista 1 para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 3D referentes à listal utilizado K-means

a

Resultados em 3D referentes à listal utilizado EM

b

Figura Apêndice 2.5 – Gráficos 2D referentes à lista 2, com atributos relacionados à validação do código das páginas, para a execução com máximo de 3 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 2D referentes à lista2 utilizado K-means

a

Resultados em 2D referentes à lista2 utilizado EM

b

Figura Apêndice 2.6 – Gráficos 3D referentes à lista 2, com atributos relacionados à validação do código das páginas, para a execução com máximo de 3 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 3D referentes à lista2 utilizado K-means

a

Resultados em 3D referentes à lista2 utilizado EM

b

Figura Apêndice 2.7 – Gráficos 2D referentes à lista 2, com atributos relacionados à validação do código das páginas, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 2D referentes à lista2 utilizado K-means

a

Resultados em 2D referentes à lista2 utilizado EM

b

Figura Apêndice 2.8 – Gráficos 3D referentes à lista 2, com atributos relacionados à validação do código das páginas, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 3D referentes à lista2 utilizado K-means

a

Resultados em 3D referentes à lista2 utilizado EM

b

Figura Apêndice 2.9 – Gráficos 2D referentes à lista 3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, para a execução com máximo de 3 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 2D referentes à lista3 utilizado K-means

a

Resultados em 2D referentes à lista3 utilizado EM

b

Figura Apêndice 2.10 – Gráficos 3D referentes à lista 3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, para a execução com máximo de 3 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 3D referentes à lista3 utilizado K-means

a

Resultados em 3D referentes à lista3 utilizado EM

b

Figura Apêndice 2.11 – Gráficos 2D referentes à lista3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 2D referentes à lista3 utilizado K-means

a

Resultados em 2D referentes à lista3 utilizado EM

b

Figura Apêndice 2.12 – Gráficos 3D referentes à lista3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 3D referentes à lista3 utilizado K-means

a

Resultados em 3D referentes à lista3 utilizado EM

b

Figura Apêndice 2.13 – Gráficos 2D referentes à lista 4, com atributos relacionados à estrutura dos sites, para a execução com máximo de 3 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 2D referentes à lista4 utilizado K-means

a

Resultados em 2D referentes à lista4 utilizado EM

b

Figura Apêndice 2.14 – Gráficos 3D referentes à lista 4, com atributos relacionados à estrutura dos sites, para a execução com máximo de 3 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 3D referentes à lista4 utilizado K-means

a

Resultados em 3D referentes à lista4 utilizado EM

b

Figura Apêndice 2.15 – Gráficos 2D referentes à lista 4, com atributos relacionados à estrutura dos sites, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 2D referentes à lista4 utilizado K-means

a

Resultados em 2D referentes à lista4 utilizado EM

b

Figura Apêndice 2.16 – Gráficos 3D referentes à lista 4, com atributos relacionados à estrutura dos sites, para a execução com máximo de 5 grupos. a. Gráfico para o algoritmo K-means; b. Gráfico para o algoritmo EM.

Resultados em 3D referentes à lista4 utilizado K-means

a

Resultados em 3D referentes à lista4 utilizado EM

b

APÊNDICE 3 TABELAS INDICATIVAS DA ORDEM DE RELEVÂNCIA DOS ATRIBUTOS

Tabela Apêndice 3.1 – Ordem de relevância dos atributos da lista 1 para as classificações resultantes da utilização dos métodos K-means e EM para 3 e 5 grupos.

Posição	Lista 1			
	K-means		EM	
	3 Grupos	5 Grupos	3 Grupos	5 Grupos
1	n_pag	avg_pag_age	n_pag	n_pag
2	ipv6_address	ipv6_address	count_ok	count_ok
3	count_ok	n_pag	brazil	brazil
4	brazil	ntp_offset	avg_depth	max_depth
5	ns_ipv6_sup	non_indentified	apache	apache
6	reply_get	brazil	microsoft_IIS	microsoft_IIS
7	avg_depth	ns_ipv6_sup	nginx	nginx
8	apache	reply_get	ntp_stratum	ntp_stratum
9	microsoft_IIS	std_pag_size	max_pag_size	max_pag_size
10	avg_p1_warn	avg_depth	max_depth	avg_depth
11	alt_address	ntp_stratum	zope	zope
12	nginx	avg_p1_warn	non_indentified	non_indentified
13	internal_links	alt_address	count_html_ok	avg_p1_warn
14	non_indentified	apache	avg_p1_warn	count_html_ok
15	ns_alt_ipv6_sup	microsoft_IIS	robots_txt	robots_txt
16	in_degree	nginx	min_pag_age	min_pag_age
17	avg_p2_errors	ns_alt_ipv6_sup	accessibility_lvl	std_pag_size
18	zope	internal_links	std_pag_size	accessibility_lvl
19	count_html_ok	out_degree	min_pag_size	min_pag_size
20	ntp_stratum	min_pag_age	avg_p1_errors	ns_ipv6_sup
21	max_depth	robots_txt	ns_ipv6_sup	avg_p1_errors
22	reply_ping	total_size	count_static	avg_html_errors
23	count_dynamic	in_degree	count_dynamic	count_static
24	out_degree	max_pag_age	avg_html_errors	count_dynamic
25	max_pag_size	avg_p2_errors	total_size	total_size
26	avg_html_errors	avg_size	avg_size	avg_size
27	avg_p3_warn	reply_ping	in_degree	count_pag_A
28	std_pag_size	zope	avg_p2_errors	in_degree
29	robots_txt	count_html_ok	max_pag_age	avg_p2_errors
30	accessibility_lvl	min_pag_size	http_delay	http_delay
31	count_pag_AAA	avg_html_errors	count_pag_AA	count_pag_AA
32	min_pag_age	avg_p1_errors	count_pag_A	max_pag_age
33	avg_p2_warn	max_pag_size	out_degree	out_degree
34	count_static	accessibility_lvl	avg_p3_errors	avg_p3_warn
35	count_pag_A	http_delay	ipv6_address	avg_pag_age
36	min_pag_size	avg_p3_errors	avg_pag_age	ipv6_address
37	avg_size	max_depth	avg_p3_warn	ns_alt_ipv6_sup

Posição	Lista 1			
	K-means		EM	
	3 Grupos	5 Grupos	3 Grupos	5 Grupos
38	avg_p1_errors	count_dynamic	ns_alt_ipv6_sup	reply_get
39	total_size	ntp_delay	reply_get	avg_p3_errors
40	count_pag_AA	count_static	avg_p2_warn	avg_p2_warn
41	max_pag_age	count_ok	internal_links	count_pag_AAA
42	avg_pag_age	count_pag_AAA	ntp_offset	ntp_offset
43	avg_p3_errors	http_offset	ntp_delay	ntp_delay
44	http_offset	count_pag_A	http_offset	internal_links
45	http_delay	avg_p2_warn	count_pag_AAA	http_offset
46	ntp_delay	count_pag_AA	alt_address	alt_address
47	ntp_offset	avg_p3_warn	reply_ping	reply_ping
48	count_redirect	count_redirect	count_redirect	count_redirect
49	count_text_html	count_text_html	count_text_html	count_text_html
50	count_text_plain	count_text_plain	count_text_plain	count_text_plain
51	count_app	count_app	count_app	count_app

Tabela Apêndice 3.2 – Ordem de relevância dos atributos da lista 2, com atributos relacionados à validação do código das páginas, para as classificações resultantes da utilização dos métodos K-means e EM para 3 e 5 grupos.

Posição	Lista 2			
	K-means		EM	
	3 Grupos	5 Grupos	3 Grupos	5 Grupos
1	accessibility_lvl	accessibility_lvl	avg_html_errors	avg_html_errors
2	avg_p1_warn	avg_p1_warn	avg_p1_warn	avg_p1_warn
3	avg_html_errors	count_html_ok	avg_p2_warn	avg_p2_warn
4	avg_p2_errors	count_pag_A	avg_p3_warn	avg_p3_warn
5	avg_p3_errors	count_pag_AAA	count_html_ok	count_html_ok
6	avg_p2_warn	avg_p2_errors	accessibility_lvl	accessibility_lvl
7	count_html_ok	avg_p2_warn	avg_p3_errors	avg_p3_errors
8	avg_p1_errors	avg_html_errors	avg_p1_errors	avg_p1_errors
9	avg_p3_warn	count_pag_AA	count_pag_A	count_pag_A
10	count_pag_A	avg_p3_warn	avg_p2_errors	avg_p2_errors
11	count_pag_AAA	avg_p3_errors	count_pag_AAA	count_pag_AAA
12	count_pag_AA	avg_p1_errors	count_pag_AA	count_pag_AA

Tabela Apêndice 3.3 – Ordem de relevância dos atributos da lista 3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, para as classificações resultantes da utilização dos métodos K-means e EM para 3 e 5 grupos.

Posição	Lista 3			
	K-means		EM	
	3 Grupos	5 Grupos	3 Grupos	5 Grupos
1	microsoft_IIS	microsoft_IIS	ntp_offset	ntp_offset
2	reply_get	ntp_offset	ntp_delay	ntp_delay
3	brazil	ipv6_address	brazil	brazil
4	non_indentified	nginx	apache	apache
5	apache	zope	microsoft_IIS	microsoft_IIS
6	ntp_offset	brazil	nginx	nginx
7	ntp_delay	apache	non_indentified	non_indentified
8	nginx	ntp_delay	zope	zope
9	zope	ns_ipv6_sup	ntp_stratum	ntp_stratum
10	reply_ping	alt_address	ns_ipv6_sup	ns_ipv6_sup
11	ntp_stratum	reply_get	http_delay	http_delay
12	ns_ipv6_sup	ntp_stratum	ns_alt_ipv6_sup	ns_alt_ipv6_sup
13	http_delay	non_indentified	reply_get	reply_get
14	ipv6_address	reply_ping	ipv6_address	ipv6_address
15	alt_address	ns_alt_ipv6_sup	alt_address	alt_address
16	ns_alt_ipv6_sup	http_delay	http_offset	http_offset
17	http_offset	http_offset	reply_ping	reply_ping

Tabela Apêndice 3.4 – Ordem de relevância dos atributos da lista 4, com atributos relacionados à estrutura dos sites, para as classificações resultantes da utilização dos métodos K-means e EM para 3 e 5 grupos.

Posição	Lista 4			
	K-means		EM	
	3 Grupos	5 Grupos	3 Grupos	5 Grupos
1	avg_pag_age	n_pag	avg_size	avg_size
2	n_pag	avg_pag_age	avg_depth	avg_depth
3	avg_depth	max_pag_size	min_pag_age	min_pag_age
4	avg_size	max_depth	robots_txt	robots_txt
5	count_ok	std_pag_size	total_size	total_size
6	robots_txt	avg_depth	min_pag_size	min_pag_size
7	internal_links	count_ok	in_degree	in_degree
8	out_degree	min_pag_size	std_pag_size	std_pag_size
9	in_degree	min_pag_age	max_pag_age	max_pag_age
10	min_pag_age	internal_links	count_dynamic	count_dynamic
11	max_pag_age	robots_txt	max_pag_size	max_pag_size
12	min_pag_size	in_degree	internal_links	internal_links
13	max_pag_size	count_dynamic	max_depth	max_depth
14	total_size	total_size	out_degree	out_degree
15	count_dynamic	count_static	avg_pag_age	avg_pag_age

Posição	Lista 4			
	K-means		EM	
	3 Grupos	5 Grupos	3 Grupos	5 Grupos
16	count_static	out_degree	n_pag	n_pag
17	std_pag_size	avg_size	count_static	count_static
18	max_depth	max_pag_age	count_ok	count_ok
19	count_redirect	count_redirect	count_redirect	count_redirect
20	count_text_html	count_text_html	count_text_html	count_text_html
21	count_text_plain	count_text_plain	count_text_plain	count_text_plain

APÊNDICE 4 TABELAS EXEMPLIFICANDO A COMPOSIÇÃO DOS GRUPOS

Tabela Apêndice 4.1 – Atributos mais relevantes do grupo dos 20 sites do governo brasileiro mais populares segundo o site Alexa.com para a execução do algoritmo K-means para a lista 1 de atributos e com máximo de 3 grupos.

Sites	lista 1			
	n_pag	ipv6_address	count_ok	grupo
caixa.gov.br	38	0	38	2
fazenda.gov.br	9175	0	9175	1
pr.gov.br	1	0	1	2
mec.gov.br	1	0	1	2
mg.gov.br	3308	0	3308	1
dataprev.gov.br	119	0	119	2
bcb.gov.br	6858	0	6858	1
ibge.gov.br	6630	0	6630	1
previdencia.gov.br	432	0	432	2
ce.gov.br	1	0	1	2
to.gov.br	24	0	24	2
pa.gov.br	115	0	115	2
siapenet.gov.br	462	0	462	2
se.gov.br	351	0	351	2
mds.gov.br	4686	0	4686	1
capes.gov.br	6145	0	6145	1
comprasnet.gov.br	452	0	452	2
ap.gov.br	340	0	340	2
tcu.gov.br	17	0	17	2
mpas.gov.br	224	0	224	2

Tabela Apêndice 4.2 – Atributos mais relevantes do grupo dos 20 sites do governo brasileiro mais populares segundo o site Alexa.com para a execução do algoritmo K-means para a lista 2, com atributos relacionados à validação do código das páginas, e com máximo de 3 grupos.

Sites	lista 2			grupo
	accessibility_lvl	avg_p1_warn	avg_html_errors	
caixa.gov.br	0	348.1	869.4	1
fazenda.gov.br	0	128.4	243.4	3
pr.gov.br	0	0.0	3.0	3
mec.gov.br	0	16.0	1.0	3
mg.gov.br	0	422.7	18.0	1
dataprev.gov.br	0	278.5	925.5	1
bcb.gov.br	0	143.6	31.9	3
ibge.gov.br	0	145.7	120.9	3
previdencia.gov.br	0	128.2	26.5	3
ce.gov.br	0	2.0	3.0	3
to.gov.br	0	41.5	8.3	3
pa.gov.br	0	110.8	65.0	3
siapenet.gov.br	0	50.5	53.8	3
se.gov.br	0	109.4	43.6	3
mds.gov.br	0	260.3	11.4	3
capex.gov.br	0	136.8	1.7	3
comprasnet.gov.br	0	49.5	40.7	3
ap.gov.br	0	38.8	17.8	3
tcu.gov.br	0	18.4	18.9	3
mpas.gov.br	0	146.9	27.8	3

Tabela 4.3 Apêndice – Atributos mais relevantes do grupo dos 20 sites do governo brasileiro mais populares segundo o site Alexa.com para a execução do algoritmo K-means para a lista 3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, e com máximo de 3 grupos.

Sites	Lista 3			
	microsoft_IIS	reply_get	brazil	grupo
caixa.gov.br	1	0	1	1
fazenda.gov.br	1	0	1	1
pr.gov.br	0	0	1	2
mec.gov.br	0	0	1	2
mg.gov.br	0	0	1	2
dataprev.gov.br	1	0	1	1
bcb.gov.br	0	0	1	2
ibge.gov.br	0	0	1	2
previdencia.gov.br	0	0	1	2
ce.gov.br	0	0	1	2
to.gov.br	0	0	1	2
pa.gov.br	1	0	1	1
siapenet.gov.br	1	0	1	1
se.gov.br	0	0	1	2
mds.gov.br	0	0	1	2
capex.gov.br	0	0	1	2
comprasnet.gov.br	0	0	1	2
ap.gov.br	0	0	1	2
tcu.gov.br	1	0	1	1
mpas.gov.br	0	0	1	2

Tabela Apêndice 4.4 – Atributos mais relevantes do grupo dos 20 sites do governo brasileiro mais populares segundo o site Alexa.com para a execução do algoritmo K-means para a lista 4, com atributos relacionados à estrutura dos sites, e com máximo de 3 grupos.

Sites	Lista 4			
	avg_pag_age (dias)	n_pag	avg_depth	grupo
caixa.gov.br	0.00	38	2.76	1
fazenda.gov.br	1308.75	9175	5.32	3
pr.gov.br	935.08	1	1.00	2
mec.gov.br	30.20	1	1.00	1
mg.gov.br	885.54	3308	4.22	3
dataprev.gov.br	1578.35	119	4.45	2
bcb.gov.br	1733.68	6858	6.40	3
ibge.gov.br	324.29	6630	5.01	3
previdencia.gov.br	815.96	432	2.82	1
ce.gov.br	304.76	1	7.00	1
to.gov.br	0.00	24	2.83	1
pa.gov.br	442.09	115	5.00	1
siapenet.gov.br	55.93	462	11.66	1
se.gov.br	0.00	351	3.38	1
mds.gov.br	606.10	4686	4.16	3
capex.gov.br	1.74	6145	3.64	3
comprasnet.gov.br	1863.61	452	4.17	2
ap.gov.br	209.31	340	5.04	1
tcu.gov.br	0.00	17	5.35	1
mpas.gov.br	386.33	224	4.81	1

Tabela Apêndice 4.5 – Atributos mais relevantes de cada um dos grupos encontrados na execução do algoritmo K-means para a lista 1 de atributos e com máximo de 3 grupos.

Sites	lista 1			
	grupo	n_pag	ipv6_address	count_ok
www.crbio-3.gov.br	1	9997	0	9997
www.irga.rs.gov.br		4223	0	4223
www.seduc.mt.gov.br		8341	0	8341
www.camarajoanopolis.sp.gov.br		8131	0	8131
www.geoparkbodoquenapantanal.ms.gov.br		5534	0	5534
www.flexsi.mp.pr.gov.br	2	1	0	1
www.desenvolvimento.saojoaodelrei.mg.gov.br		1	0	1
www.camaratupancireta.rs.gov.br		1	0	1
www.septp.pa.gov.br		1	0	1
saude.wbz.pr.gov.br		1	0	1
majorgercino.sc.gov.br	3	18	1	18
www.camarasbs.sp.gov.br		171	1	171
www.tijucas.sc.gov.br		6241	1	6241
cidadao-unstable.saocarlos.sp.gov.br		1	1	1
www.cmnovalima.mg.gov.br		54	1	54

Tabela Apêndice 4.6 – Atributos mais relevantes de cada um dos grupos encontrados na execução do algoritmo K-means para a lista 2, com atributos relacionados à validação do código das páginas, e com máximo de 3 grupos.

Sites	lista 2			
	grupo	accessibility lvl	avg_p1 warn	avg_html errors
www.crbio-3.gov.br	1	0	510,3	285,2
www.irga.rs.gov.br		0	197,5	205,6
www.cachoeirinha.rs.gov.br		0	1095,9	11,8
www.compras.rs.gov.br		0	141,7	219,4
agencia.sanesul.ms.gov.br		0	995,6	0,0
plprmariahelena.seed.pr.gov.br	2	3	106,0	18,0
cscjardiminterlagos.seed.pr.gov.br		3	93,0	18,5
www.dengue.pr.gov.br		2	54,0	27,0
wbzmiguelnassif.seed.pr.gov.br		3	106,0	18,0
llsmariapereira.seed.pr.gov.br		3	106,0	18,0
www.flexsi.mp.pr.gov.br	3	0	12,0	21,0
www.desenvolvimento.saojoaodelrei.mg.gov.br		0	1,0	1,0
www.camaratupancireta.rs.gov.br		0	0,0	2,0
www.septp.pa.gov.br		0	4,0	15,0
saude.wbz.pr.gov.br		0	0,0	3,0

Tabela Apêndice 4.7 – Atributos mais relevantes de cada um dos grupos encontrados na execução do algoritmo K-means para a lista 3, com atributos relacionados à infraestrutura disponibilizada pelos servidores, e com máximo de 3 grupos.

Sites	Lista 3			
	grupo	microsoft_IIIS	reply_get	brazil
www.septp.pa.gov.br	1	1	0	1
www.siges.es.gov.br		1	0	1
www.eavparquelage.rj.gov.br		1	0	1
www.barradesantarosa.pb.gov.br		1	0	1
www.seduc.mt.gov.br		1	0	1
www.flexsi.mp.pr.gov.br	2	0	0	1
www.desenvolvimento.saojoaodelrei.mg.gov.br		0	0	1
www.camaratupancireta.rs.gov.br		0	0	1
saude.wbz.pr.gov.br		0	0	1
www.crbio-3.gov.br		0	0	1
majorgercino.sc.gov.br	3	0	1	1
www.tijucas.sc.gov.br		0	1	1
www.cmnovalima.mg.gov.br		1	1	0
www.riolandia.sp.gov.br		0	1	1
www.moradanova.mg.gov.br		0	1	0

Tabela Apêndice 4.8 – Atributos mais relevantes de cada um dos grupos encontrados na execução do algoritmo K-means para a lista 4, com atributos relacionados à estrutura dos sites, e com máximo de 3 grupos.

Sites	Lista 4			
	grupo	avg_pag_age (dias)	n_pag	avg_depth
www.flexsi.mp.pr.gov.br	1	24.4	1	2.00
www.desenvolvimento.saojoaodelrei.mg.gov.br		56.7	1	3.00
webmail.posse.go.gov.br		0.0	2	4.00
www.siges.es.gov.br		0.0	1	1.00
webmail.timbo.sc.gov.br		0.0	1	1.00
www.camaratupancireta.rs.gov.br	2	1837.6	1	1.00
www.septp.pa.gov.br		1644.9	1	1.00
saude.wbz.pr.gov.br		935.4	1	1.00
www.redepro.rs.gov.br		2226.0	1	1.00
webmail.srh.ce.gov.br		1423.0	1	1.00
www.crbio-3.gov.br	3	0.0	9997	9.14
www.qualidade.defesacivil.rj.gov.br		0.0	5616	6.57
www.irga.rs.gov.br		0.0	4223	9.50
www.seduc.mt.gov.br		0.0	8341	9.75
www.camarajoanopolis.sp.gov.br		0.0	8131	5.25

APÊNDICE 5 ESPECIFICAÇÃO DE INDICADORES

O projeto TIC Web (Indicadores para desenvolvimento de Tecnologias de Informação e Comunicação – Web) possui como principal objetivo o estudo periódico dos *sites* constituintes da Web Brasileira, e em sua fase inicial, mais especificamente nos da Web Governamental Brasileira, definida pelos sítios sob o domínio *.gov.br*. Ele se distancia dos trabalhos anteriormente citados por não ser uma estudo pontual e pelo fato de executar uma análise censitária dos *sites* pertencentes aos domínio sob inspeção, a fim de conseguir maior precisão nos resultados contra o que poderia ser obtido com uma análise amostral.

Os subitens a seguir descrevem o escopo da pesquisa preliminar do TIC Web sob o *gov.br*, a qual será utilizada como base a realização das análises e avaliação do funcionamento dos métodos de aprendizado de máquina a serem aplicados. Também, neles são especificados os indicadores adotados e uma proposta de novos itens a serem incluídos.

5.1 Escopo da Análise

O projeto TIC Web, foi iniciado em 2009 e começou em 2010 a realizar coletas exaustivas anuais sobre as páginas da Web Governamental Brasileira. Até o presente momento, já foram realizadas três coletas e a presente pesquisa pretende utilizar o conjunto de dados por elas construído.

Dado o tamanho da Web, é importante ressaltar que, para a realização desse tipo de pesquisa, é necessário que se delimite o escopo dos *sites* que se deseja estudar. No caso do TIC Web e, também, da presente dissertação, foi definido que um determinado domínio seria composto por todas as páginas alcançáveis (segundo metodologia de pesquisa posteriormente explicada) sob o *Top Level Domain* (TLD) que representa tal domínio. Como exemplo, podem se citar os TLDs *.com.br*, *.edu.br*, *.net.br* e *.gov.br*.

Para cada uma das coletas, os dados são separados em duas unidades de pesquisa. Uma contém informações relativas às páginas encontradas, como a validação do código *Hypertext Markup Language* (HTML), e é identificada pela própria *Uniform Resource Locator* (URL) de acesso página. Outra, com informações

relativas aos *sites*, geralmente comuns a todas as suas páginas, como sua localização geográfica, tipo de servidor, suporte ao IPv6 ou, ainda, com informações que agrupam os dados de suas respectivas páginas, como a quantidade de páginas aderentes aos padrões HTML. Essa segunda unidade é identificada pela URL do *site*.

5.2 Indicadores de Sites

A unidade de pesquisa **site** é identificada pela URL da página inicial do *site*, definida como sendo a página que contém apenas um caractere '/' após a *string* que define o domínio do *site*, e.g., "www.exemplo.com.br/". Assim, todas as páginas que compartilham de uma mesma *string* inicial são consideradas como pertencentes a um mesmo *site*.

Os atributos avaliados para esta unidade de pesquisa estão listados na Tabela Apêndice 5.1. Para facilitar o entendimento, eles foram relacionados tanto a seus respectivos conceitos correlatos, de acordo com o explicitado nos Capítulos 0 e 1 do corpo principal desta dissertação, quanto com as ferramentas responsáveis por obtê-los, conforme será apreseto no Apêndice 6.

Tabela Apêndice 5.1 – Conjunto de indicadores analisados pelo projeto TIC Web para a unidade de pesquisa *site*.

Indicador	Ferramenta de extração	Conceito Relacionado
Nome do <i>site</i>	WIRE	
TLD	ConNeCTOR	Padrões abertos
Domínio	ConNeCTOR	
Tipo de servidor	ConNeCTOR	
Nível de acessibilidade do <i>site</i>	ConNeCTOR	Padrões de acessibilidade
Endereço IPv4	ConNeCTOR	
Endereço IPv6	ConNeCTOR	Padrões de rede
Possui endereço IPv6 no domínio principal	ConNeCTOR	

Indicador	Ferramenta de extração	Conceito Relacionado
Possui endereço IPv6 em domínio alternativo	ConNeCTOR	
O endereço responde a requisições HTTP GET via IPv6	ConNeCTOR	
O endereço responde a requisições PING6	ConNeCTOR	
Os servidores DNS do domínio principal suportam IPv6	ConNeCTOR	
Os servidores DNS de algum domínio alternativo suportam IPv6	ConNeCTOR	
Localização geográfica	ConNeCTOR	Localização geográfica
Quantidade de páginas sem erros na validação de HTML	ConNeCTOR	Padrões HTML
Diferença de tempo em relação ao UTC medida via HTTP	ConNeCTOR	
Stratum do serviço NTP	ConNeCTOR	Padrões de sincronização de tempo
Diferença de tempo em relação ao UTC medida via NTP	ConNeCTOR	
Tempo de resposta de requisição HTTP	ConNeCTOR	
Tempo de resposta via NTP	ConNeCTOR	Tempo de resposta
Número total de páginas	WIRE	
Tamanho total das páginas	WIRE	
Quantidade de páginas dinâmicas	WIRE	
Quantidade de páginas estáticas	WIRE	
Idade da página mais antiga	WIRE	
Idade da página mais nova	WIRE	Dados quantitativos
Média das idades das páginas	WIRE	
Possui arquivo robots.txt válido	WIRE	
Grau de entrada	WIRE	
Grau de saída	WIRE	
Links internos	WIRE	

Indicador	Ferramenta de extração	Conceito Relacionado
Profundidade máxima das páginas	WIRE	
Grupo em que se encontra na estrutura macroscópica da Web	WIRE	

Desta lista, vale descrever o processo de aquisição dos seguintes atributos:

- **TLD (Top Level Domain):** é extraído da URL do *site*. Para o caso de *sites* governamentais estaduais e municipais, o algoritmo de extração considera a *substring* dessa URL que representa a unidade federativa também é considerada como parte do TLD.
- **Nível de acessibilidade do *site*:** do menor nível para o maior, um *site* pode ser considerado, não acessível, “A”, “AA” e “AAA”, segundo definição encontrada na seção 2.3. Para que ele seja atribuído com a classificação “A”, todas as suas páginas devem ser pelo menos de nível “A”. Para o nível “AA”, todas as páginas devem ser pelo menos de nível “AA”. O equivalente vale para que o *site* receba a classificação “AAA”, todas as suas páginas devem ser de nível “AAA”. Nota-se que essas classificações são exclusivas assim, se um *site* for considerado “AA” ele não é mais um *site* “A”.
- **Possui endereço IPv6 em domínio alternativo:** por domínio alternativo são consideradas modificações na URL original do *site* que adicionem algum prefixo indicativo do uso de IPv6 no *site*. No caso, os prefixos utilizados foram: “www.v6”, “v6”, “www6”, “www.ipv6”, “ipv6”.
- **Localização Geográfica:** obtido com o auxílio de uma biblioteca e base de dados da API GeoIP provida pela empresa MaxMind¹³. Este campo é formado de duas letras correspondentes ao país onde o endereço IP registrado está alocado.
- **Grau de entrada:** contabilização dos *sites* que possuem *links* apontando para este.

¹³ Disponível em: <http://www.maxmind.com/app/ip-location>. Acessado em: 18 de abril de 2014

- **Grau de saída:** contabilização dos *sites* para as quais este *site* possui *links*.
- **Links internos:** quantidade de *links* que apontam para o próprio *site*.
- **Grupo em que se encontra na estrutura macroscópica da Web:** obtido com o WIRE. Segue a definição da estrutura da Web descrita na seção 3.1.

5.3 Indicadores de Páginas

A unidade de pesquisa **página** é identificada pela própria URL da página, e.g., “http://www.exemplo.com.br/local/pagina”. Esse atributo serve também para a diferenciação entre páginas dinâmicas e estáticas, considerando, para tanto, a extensão do documento referenciado e a existência de parâmetros HTTP embutidos na URL. Outra aplicação para essa variável é a identificação do tipo de linguagem na qual a página foi codificada, também definida pela extensão do documento por ela referenciada.

Os atributos avaliados para esta unidade de pesquisa estão listados na Tabela Apêndice 5.2. Para facilitar seu entendimento, eles foram relacionados tanto a seus respectivos conceitos correlatos, de acordo com o explicitado nos Capítulos 0 e 1 do corpo principal desta monografia, quanto às ferramentas responsáveis por obtê-los, conforme será apresentado no Apêndice 6.

Tabela Apêndice 5.2 – Conjunto de indicadores analisados pelo projeto TIC Web para a unidade de pesquisa página.

Indicador	Ferramenta de extração	Conceito Relacionado
Nome do site	WIRE	
Nome da página	WIRE	
Status HTTP	WIRE	Padrões abertos
Tipo do documento	ConNeCTOR	
Tipo MIME	WIRE	
Quantidade de erros de acessibilidade de prioridade 1	ConNeCTOR	
Quantidade de erros de acessibilidade de prioridade 2	ConNeCTOR	Padrões de acessibilidade
Quantidade de erros de acessibilidade de prioridade 3	ConNeCTOR	
Quantidade de avisos de acessibilidade de prioridade 1	ConNeCTOR	

Indicador	Ferramenta de extração	Conceito Relacionado
Quantidade de avisos de acessibilidade de prioridade 2	ConNeCTOR	
Quantidade de avisos de acessibilidade de prioridade 3	ConNeCTOR	
Quantidade de erros na validação HTML	ConNeCTOR	Padrões HTML
Tamanho	WIRE	
Página dinâmica	WIRE	
Idade	WIRE	
Tamanho do conteúdo extraído da página	WIRE	Dados quantitativos
Grau de entrada	WIRE	
Grau de saída	WIRE	
Profundidade	WIRE	

Desta lista, vale descrever o processo de aquisição dos seguintes atributos:

- **Status HTTP:** obtido da resposta à requisição HTTP GET feita à URL da página. Ele indica se a página estava OK (valores entre 200 e 299) ou não e, em caso negativo se ela está localizada em outro local (valores entre 300 e 399), se ela não pôde ser acessada devido à problema com o servidor (valores de 500 a 599), dentre outros problemas que possam ter ocorrido no processo.
- **Tipo de documento:** obtido pelo validador de páginas do W3C a partir de análise de *tag* HTML que identifica a versão de HTML em que a página está escrita. Alguns exemplos são: HTML 5, HTML 4.1 Strict, XHTML 1.0 Transitional e XHTML1.1.
- **Tipo MIME:** obtido tanto pela resposta à requisição HTTP GET feita à URL da página, quanto pela análise de *meta tag* contendo o atributo *content type* existente no cabeçalho da página. Alguns exemplos de tipos são: “text/htm”, “text/html”, “audio/basic” e “application/pdf”
- **Erros e avisos de acessibilidade com prioridades de 1 a 3:** todos esses indicadores referentes à validação de acessibilidade foram obtidos com o

auxílio do Avaliador e Simulador de Acessibilidade de Sítios (ASES) com a opção de verificação do Modelo de Acessibilidade de Governo Eletrônico (E-MAG)

- **Página dinâmica:** para o WIRE uma página é considerada dinâmica quando sua URL possui o caractere '?', que representa a especificação de parâmetros para que o servidor Web gere o código HTML de resposta.
- **Idade:** obtido a partir do cálculo da diferença entre o valor do cabeçalho *last-modified*, da resposta HTTP, e o instante de aquisição da página pelo *crawler*.
- **Tamanho do conteúdo extraído da página:** esse valor é a contabilização do tamanho da página após a extração de *tags* HTML ou outros códigos que a página contenha. Ou seja, ele designa o tamanho do conteúdo textual da página.
- **Grau de entrada:** contabilização das páginas que possuem *links* apontando para esta página.
- **Grau de saída:** contabilização das páginas para as quais esta página possui *links*.
- **Profundidade:** indica a menor quantidade de páginas que deve ser percorrida para que se consiga atingir a página em questão a partir da página inicial do *site*.

Devido a seu volume, esta unidade de pesquisa não foi utilizada para as análises com a aplicação das técnicas de aprendizado de máquina propostas para esta dissertação, os atributos descritos na Tabela Apêndice 5.3 constem de alternativa para a consolidação dos dados na unidade de páginas. Faz-se, no entanto, uma ressalva para o indicador WAB discutido na Seção 3.2 que apesar de sua relevância e aplicabilidade, não pôde ser calculado devido a limitações das ferramentas, incapazes, em seu atual estado, de contabilizar o número potencial de barreiras de uma determinada página.

Tabela Apêndice 5.3 - Conjunto de atributos consolidados de páginas que se pretende incluir aos dados de sites

Indicador	Ferramenta de extração	Conceito Relacionado
Quantidade de páginas OK (HTTP status = 2XX)	WIRE	Padrões abertos
Quantidade de redirecionamentos (HTTP status = 3XX)	WIRE	
Quantidade de erros de requisição (HTTP status = 4XX)	WIRE	
Quantidade de erros de servidor (HTTP status = 5XX)	WIRE	
MIME text/html	WIRE	
MIME text/plain	WIRE	
MIME application	WIRE	
MIME outros	WIRE	
Quantidade de páginas sem erros de prioridade 1	ConNeCTOR	Padrões de acessibilidade
Quantidade média de erros de prioridade 1 por página	ConNeCTOR	
Quantidade mediana de erros de prioridade 1 por página	ConNeCTOR	
Quantidade de páginas sem erros de prioridade 2	ConNeCTOR	
Quantidade média de erros de prioridade 2 por página	ConNeCTOR	
Quantidade mediana de erros de prioridade 2 por página	ConNeCTOR	
Quantidade de páginas sem erros de prioridade 3	ConNeCTOR	
Quantidade média de erros de prioridade 3 por página	ConNeCTOR	
Quantidade mediana de erros de prioridade 3 por página	ConNeCTOR	
Quantidade de páginas sem avisos de prioridade 1	ConNeCTOR	
Quantidade média de avisos de prioridade 1 por página	ConNeCTOR	
Quantidade mediana de avisos de prioridade 1 por página	ConNeCTOR	
Quantidade de páginas sem avisos de prioridade 2	ConNeCTOR	
Quantidade média de avisos de prioridade 2 por página	ConNeCTOR	

Indicador	Ferramenta de extração	Conceito Relacionado
Quantidade mediana de avisos de prioridade 2 por página	ConNeCTOR	
Quantidade de páginas sem avisos de prioridade 3	ConNeCTOR	
Quantidade média de avisos de prioridade 3 por página	ConNeCTOR	
Quantidade mediana de avisos de prioridade 3 por página	ConNeCTOR	
Quantidade média de erros na validação HTML por página	ConNeCTOR	Padrões HTML
Quantidade mediana de erros na validação HTML por página	ConNeCTOR	
Tamanho médio por página	WIRE	
Tamanho mediano por página	WIRE	
Idade mediana das páginas	WIRE	Dados quantitativos
Profundidade mediana das páginas	WIRE	
Profundidade média	WIRE	

APÊNDICE 6 FERRAMENTAS DE EXTRAÇÃO DE ATRIBUTOS

Neste apêndice são abordadas as duas principais ferramentas utilizadas no projeto TIC Web (Indicadores para desenvolvimento de Tecnologias de Informação e Comunicação – Web) para a extração dos indicadores tratados no capítulo anterior. O estudo de suas funcionalidades e atributos é útil para um melhor entendimento das limitações e características do escopo de análise do projeto.

6.1 Webcrawler

Para encontrar e baixar páginas, este projeto utiliza uma versão modificada¹⁴ de um webcrawler chamado WIRE (*Web Information Retrieval Environment*), projeto de software livre desenvolvido sobre plataforma C++ já utilizado em estudos de domínios Web de outros países, como o descrito no artigo de Castillo et al. (2007). A arquitetura desse programa está representada na Figura Apêndice 6.1. Basicamente, a ferramenta é composta por quatro módulos independentes entre si que são executados em ciclos. O primeiro é o *seeder*, responsável pelo carregamento e registro de novas URLs (Localizador-Padrão de Recursos) no sistema. A seguir vem o *manager* que aplica diversas políticas de seleção para escolher quais páginas deverão ser baixadas no ciclo em curso. O terceiro módulo é chamado de *harvester* e possui a função de baixar e armazenar as páginas previamente selecionadas. Por último, o módulo *gatherer* analisa o conteúdo baixado extraíndo *hiperlinks* para a construção da lista de URLs que dará continuidade ao processo.

Para a inicialização desse processo, o *crawler* necessita de uma semente, lista inicial de *sites* a serem percorridos. Idealmente, para a análise completa de um determinado domínio, essa lista deve conter todos os *sites* nele presentes. Todavia, listas como essa são difíceis de serem obtidas uma vez que o Sistema de Nomes de Domínio (DNS) é distribuído e não provê maneira para a obtenção da lista completa. Todavia, o servidor de DNS autoritativo do *.gov.br* permite a execução de requisições recursivas que fornecem os nomes de todos os domínios diretamente a baixo desse

¹⁴ Acessado em: <http://sf.net/projects/wire-nic>

Domínio de Topo (TLD). Os *sites* sob esses domínios tiveram de ser encontrados pelo processo de *crawling* durante a coleta.

Figura Apêndice 6.1 - Diagrama que representa o ciclo de funcionamento do WIRE

Com o carregamento da semente os ciclos de download podem ser iniciados. Seu funcionamento é análogo aos descritos por algoritmos de busca em largura, ou seja, no primeiro ciclo todas as páginas iniciais dos *sites* são baixadas, então, no seguinte, todos os links dessas e assim por diante até que o *site* seja baixado por completo.

Na prática, o processo não é tão simples. O *crawler* precisa implementar diversos mecanismos de controle para evitar *sites* extremamente grandes ou que contenham recursões infinitas. Esse tipo de erro ocorre principalmente em conteúdos dinamicamente gerados, como calendários. Um exemplo de controle que é aplicado é a limitação da profundidade máxima de um *site*.

Outra limitação inerente à mecânica desse tipo de coleta é o fato de o *crawler* não conseguir acessar a chamada Web profunda, acessível apenas por meio de formulários ou senhas de acesso, ou contabilizar a existência de ilhas, páginas que não referenciadas por nenhuma das páginas previamente acessadas.

Outras opções de software que conseguissem realizar esse tipo de busca como o Heritrix apresentado por Mohr, Stack, Ranitovic, Avery, & Kimpton (2004) e o Apache Nutch apresentado por Cafarella & Cutting (2004) também foram inicialmente consideradas, porém o WIRE foi escolhido por ter configuração mais

sim e poder mais facilmente ser adaptado a necessidades do projeto, como a definição de domínios específicos a serem baixados. Além disso, ele já fornecia uma grande quantidade de indicadores, análises e relatórios cruciais ao projeto.

Mesmo o WIRE sendo bastante completo, algumas alterações tiveram de ser feitas. Exemplos foram: um melhor suporte a requisições do tipo HTTP 1.1 (Hypertext Transfer Protocol versão 1.1) em relação ao download de conteúdos com a opção de transferência “*chunked*”; e, o armazenamento das páginas diretamente no sistema de arquivos visando facilitar o acesso da ferramenta de análise que teriam de ser desenvolvidas.

6.2 Ferramenta de Análise

Com a intenção de adicionar novas análises àquelas já oferecidas pelo *webcrawler*, um novo programa, batizado de *Convenient Network Characteristics Testing Organized Routines* (ConNeCTOR), foi desenvolvido. A plataforma escolhida para isso foi a Java e seu código foi publicado como licença de software livre¹⁵. Optou-se por uma arquitetura modular, que permitisse a integração de novas análises e testes à medida que necessários ao projeto. Na versão aqui utilizada, esse sistema consegue realizar as seguintes tarefas:

- Validação de conformidade com padrões HTML (*Hypertext Markup Language*) e de Acessibilidade;
- Descoberta de tipos de servidores;
- Avaliação de sincronização do relógio dos servidores;
- Verificação do tempo de resposta a requisições HTTP;
- Verificação da localização geográfica dos servidores;
- Verificação do TLD ao qual o *site* pertence;
- Verificação de suporte ao IPv6;
- Classificação de extensões das URLs encontradas no processo de *crawling*;

¹⁵ Disponível em: <http://sf.net/projects/connector-nic>. Acessado em: 18 de abril de 2014

6.2.1 Validação de páginas Web

A escolha das ferramentas de avaliação levou em conta a necessidade da utilização de ferramentas oficialmente distribuídas pelas organizações responsáveis pelo desenvolvimento dos protocolos. Assim, o *W3C Markup Validator* e o Avaliador e Simulador de Acessibilidade de Sítios (ASES), foram escolhidos para validar respectivamente códigos HTML e Acessibilidade das páginas.

O *W3C Markup Validator* é um serviço on-line de verificação da conformidade tanto com o padrão HTML quanto com outros padrões Web definidos pelo W3C, como o *Extensible HyperText Markup Language* (XHTML) e o *Cascading Style Sheets* (CSS). Contudo, neste projeto somente códigos HTML e derivados são validados. Como esse validador é um distribuído sob licença de software livre, para integrá-lo ao ConNeCTOR, uma instância local foi configurada e as avaliações foram feitas diretamente sobre ela. Isso permitiu um ganho significativo de performance da aplicação e a possibilidade de tornar a verificação distribuída.

O ASES é um software *open source* mantido pela instituição criadora do E-MAG. Seu propósito é ajudar desenvolvedores a aplicar os diversos elementos listados pelos guias de acessibilidade em seus *sítes*. Para isso, ele realiza verificações de conformidade com o Guia de Acessibilidade para Conteúdos Web versão 1.0 (WCAG 1.0) e Modelo de Acessibilidade de Governo Eletrônico versão 2.0 (E-MAG 2.0). Sua integração com o ConNeCTOR foi bastante diferente daquela realizada com o validador anterior. Como o ASES também foi desenvolvido em Java, ele pôde ser adaptado para funcionar como uma biblioteca do ConNeCTOR. Isso contribuiu para melhora do desempenho dessa validação.

6.2.2 Avaliação de sincronização do relógio dos servidores

Uma das diversas maneiras de se verificar se um servidor Web possui um relógio sincronizado com o Tempo Universal Coordenado (UTC) é comparar a diferença de horários entre eles. O ConNeCTOR implementa dois métodos para realizar tão operação. O primeiro é baseado no sistema *Network Time Protocol* (NTP), ele utiliza um programa para sistemas Linux, chamado *ntpdate*, capaz de executar requisições NTP e obter os dados referentes ao relógio da máquina sob

estudo. Com isso, o teste alcança uma precisão da ordem de microssegundos para suas medidas.

Embora esse teste consiga ótima precisão na avaliação da defasagem de tempo dos relógios, poucos servidores Web são também servidores NTP e respondem às requisições desse tipo, ou seja, apesar de muito preciso, sua abrangência é relativamente pequena. Além disso, o comportamento padrão dos sistemas NTP é funcionar tanto como cliente quanto como servidor NTP, porém em muitos servidores Web as políticas de filtros de portas TCP acabam bloqueando o serviço NTP por ele não ser considerado um serviço essencial.

Tal comportamento fez com que no ConNeCTOR fosse implementado um segundo método de verificação de sincronização de tempo, mais simples, baseado exclusivamente no protocolo HTTP. Nele uma requisição HTTP HEAD é enviada aos servidores na esperança de que eles respondam com seus respectivos horários internos, conforme descrição do protocolo na RFC 2616 escrita por Fielding et al. (1999). Posteriormente, os tempos adquiridos são utilizados no cálculo da diferença de tempo com o UTC.

O ponto negativo desse segundo método é a perda da precisão. Com ele conseguem-se apenas resultados com erros da ordem de alguns segundos. Contudo, isso já é suficiente para indicar, com alto grau de confiabilidade, se um servidor implementa ou não algum mecanismo de sincronização automática de relógio.

6.2.3 Testes Relacionados a Servidores Web

Além dos dados para as medidas de sincronização de tempo outras informações básicas sobre os servidores Web, como tipo de servidor, tempo de resposta e o endereço IP, puderam ser obtidas através do envio de requisições do tipo HTTP HEAD.

Já as informações referentes à localização geográfica do servidor foram conseguidas com o auxílio da *Application Programming Interface* (API) chamada GeoIP, criada pela MaxMind¹⁶, que fornece uma base de dados, mensalmente atualizada, para a localização geográfica por país de um servidor dado o seu

¹⁶ Disponível em: <http://www.maxmind.com/app/ip-location>. Acessado em: 18 de abril de 2014

endereço IP. Segundo testes realizados pela própria empresa, a acurácia dessa base é de 99,8%.

Por último, as informações sobre o suporte dos *sites* ao IPv6 foram as mais detalhadas. Com a utilização de requisições HTTP, DNS e PING, os seguintes elementos puderam ser analisados:

- Endereço IPv6 do *site*;
- Endereço IPv6 de nomes de *sites* alternativos;
- Se o endereço respondia ou não à requisições HTTP GET (possui servidor Web);
- Se o endereço respondia à requisições PING6;
- Se seus servidores DNS possuíam suporte à IPv6;
- Se os servidores DNS dos *sites* alternativos possuíam suporte à IPv6.

Com todos esses indicadores em mãos, consegue-se ter uma melhor ideia do progresso desse novo protocolo nos domínios estudados. Entretanto, eles não são suficientes para que se tenha garantia de que um usuário com acesso exclusivamente em IPv6 teria uma experiência de uso semelhante ao usuário com IPv4, dado que outros elementos das páginas não foram testados. Um caso típico é um *site* prover suporte à esse protocolo em seu servidor principal e não em seu servidor de imagens. Assim, quando um usuário tenta acessar tal *site* exclusivamente via IPv6, as imagens não são carregadas.