LUIZ CLÁUDIO GOMES MAIA

USO DE SINTAGMAS NOMINAIS NA CLASSIFICAÇÃO AUTOMÁTICA DE DOCUMENTOS ELETRÔNICOS

Belo Horizonte
2008

LUIZ CLÁUDIO GOMES MAIA

USO DE SINTAGMAS NOMINAIS NA CLASSIFICAÇÃO AUTOMÁTICA DE DOCUMENTOS ELETRÔNICOS

Tese apresentada ao Programa de Pós Graduação em Ciência da Informação da Escola de Ciência da Informação da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Doutor em Ciência da Informação.

Linha de pesquisa:
Organização e Uso da Informação.

Orientador:
Prof. Dr. Renato Rocha Souza.

Belo Horizonte
2008

M28u

Maia, Luiz Cláudio Gomes

Uso de sintagmas nominais na classificação automática de documentos eletrônicos [manuscritos] / Luiz Cláudio Gomes Maia. – 2008.

Orientador**:** Dr. Renato Rocha Souza.

Tese (Doutorado) – Universidade Federal de Minas Gerais. Escola de Ciência da Informação. Departamento Organização e Uso da Informação.

Bibliografia: f.

1.Indexação automática – Sintagmas Nominais. 2. Processamento Linguagem Natural 3. Ciência da informação - Teses. 4. Recuperação da Informação. I. Titulo. II. Souza, Renato Rocha. III. Unidade Federal de Minas Gerais, Escola de Ciências da Informação.

CDU : 025.4.036

UFMG

Universidade Federal de Minas Gerais
Escola de Ciência da Informação
Programa de Pós-Graduação em Ciência da Informação

FOLHA DE APROVAÇÃO

"USO DE SINTAGMAS NOMINAIS NA CLASSIFICAÇÃO AUTOMÁTICA DE DOCUMENTOS ELETRÔNICOS"

Luiz Cláudio Gomes Maia

Tese submetida à Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação em Ciência da Informação da Universidade Federal de Minas Gerais, como parte dos requisitos à obtenção do título de **"doutor em Ciência da Informação"**, linha de pesquisa **"Organização e Uso da Informação"**.

Tese aprovada em: 12 de dezembro de 2008.

Por:

Prof. Dr. Renato Rocha Souza - ECI/UFMG (Orientador)

Prof. Dr. Manoel Palhares Moreira - PUC/MG

Prof. Dr. George Leal Jamil - FUMEC

Prof. Dr. Marcello Peixoto Bax - ECI/UFMG

Profa. Dra. Beatriz Valadares Cendón - ECI/UFMG

Aprovada pelo Colegiado do PPGCI

Profa. Maria Aparecida Moura
Coordenadora

Versão final Aprovada por

Prof. Renato Rocha Souza
Orientador

Av. Antônio Carlos, 6627 – Sala 2003 – Campus Pampulha – Cx. Postal 1606 – CEP: 30161-970 – Belo Horizonte – MG
Telefone: (31)3409-6103 – Fax: (31)3409-5207 – www.eci.ufmg.br/ppgci – e-mail: ppgci@eci.ufmg.br

UFMG

Universidade Federal de Minas Gerais
Escola de Ciência da Informação
Programa de Pós-Graduação em Ciência da Informação

ATA DA DEFESA DE TESE DE **LUIZ CLÁUDIO GOMES MAIA**, matrícula: 2006203058

Às 14:00 horas do dia 12 de dezembro de 2008, reuniu-se na Escola de Ciência da Informação da UFMG a Comissão Examinadora aprovada pelo Colegiado do Programa de Pós-Graduação em Ciência da Informação em 04/12/2008, para julgar, em exame final, o trabalho intitulado ***Uso de sintagmas nominais na classificação automática de documentos eletrônicos***, requisito final para obtenção do Grau de DOUTOR em CIÊNCIA DA INFORMAÇÃO, Área de Concentração: Produção, Organização e Utilização da Informação, Linha de Pesquisa: Organização e Uso da Informação (OUI). Abrindo a sessão, o Presidente da Comissão, Prof. Dr. Renato Rocha Souza, após dar conhecimento aos presentes do teor das Normas Regulamentares do Trabalho Final, passou a palavra ao candidato para apresentação de seu trabalho. Seguiu-se a argüição pelos examinadores com a respectiva defesa do candidato. Logo após, a Comissão se reuniu sem a presença do candidato e do público, para julgamento e expedição do resultado final. Foram atribuídas as seguintes indicações:

Prof. Dr. Renato Rocha Souza - Orientador	APROVADO
Prof. Dr. Manoel Palhares Moreira	APROVADO
Prof. Dr. George Leal Jamil	APROVADO
Prof. Dr. Marcello Peixoto Bax	APROVADO
Profa. Dra. Beatriz Valadares Cendón	APROVADO

Pelas indicações, o candidato foi considerado APROVADO.

O resultado final foi comunicado publicamente ao candidato pelo Presidente da Comissão. Nada mais havendo a tratar, o Presidente encerrou a sessão, da qual foi lavrada a presente ATA que será assinada por todos os membros participantes da Comissão Examinadora.

Belo Horizonte, 12 de dezembro de 2008.

Prof. Dr. Renato Rocha Souza
Orientador - ECI/UFMG

Prof. Dr. Manoel Palhares Moreira
PUC/MG

Prof. Dr. George Leal Jamil
FUMEC

Prof. Dr. Marcello Peixoto Bax
ECI/UFMG

Profa. Dra. Beatriz Valadares Cendón
ECI/UFMG

Obs: Este documento não terá validade sem a assinatura e carimbo da Coordenadora.

Prof°. Maria Aparecida Moura
Coordenadora do Programa de Pós-Graduação em Ciência da Informação - ECI/UFMG

AGRADECIMENTOS

Na vida não conquistamos nada sozinhos. Precisamos de outras pessoas para alcançar os nossos objetivos e muitas vezes um simples gesto pode mudar a nossa vida e contribuir para o sucesso.

Gostaria de agradecer, inicialmente, à toda minha família, principalmente minha mãe Ivanilde e irmã Letícia, pelo carinho, meu irmão Marcus, pela força e amizade, e a meu pai Wellington (*in memorian*) que se foi durante o tempo de doutorado e que dedico esta conquista.

Meu agradecimento mais que especial ao professor Dr. Renato Rocha Souza, que através da confiança e incentivo transmitidos sempre com muita paciência e atenção, em cada orientação, tornou possível a realização desse trabalho.

Agradecer à professora Dra. Marlene Oliveira por ter me acolhido no doutorado de Ciência da Informação, e pelas orientações durante o percurso.

Agradecer à professora Dra. Isis Paim, paraninfa honorária desta conquista, pela atenção e gentileza na revisão minuciosa deste texto.

Agradecer aos professores Dr. Manoel Palhares Moreira, Dr. George Leal Jamil, Dr. Marcello Peixoto Bax, Dra. Beatriz Valadares Cendón e Dra. Lídia Alvarenga que participaram da banca de qualificação e defesa final deste trabalho, e muito contribuíram com opiniões e sugestões para o mesmo.

Aos colegas do PPGCI pelo companheirismo durante o curso, e também aos funcionários do PPGCI pela competência demonstrada desde a entrada do mestrado em 2003, em especial para a secretária Nely Oliveira.

Aos colegas e professores do curso de Pedagogia na Universidade do Estado de Minas Gerais pela compreensão e ajuda em minhas ausências enquanto elaborava este trabalho. Em especial, agradecer à Ester Castro, pela amizade e ajuda nos trabalhos do curso da graduação, além da revisão deste trabalho.

Algumas pessoas auxiliaram a driblar a ansiedade nesta etapa final: agradecer à Cínthia Xavier pelo carinho; e a Paulo Estevão pela amizade de sempre.

Aos alunos, professores e colegas de trabalho da Faculdade de Tecnologia INED pelo apoio. Um agradecimento especial aos professores Marlene Gomes, José Otavio, Mauro Câmara e Marlon Paolo que presenciaram o momento da defesa.

À Universidade Federal de Minas Gerais pela oportunidade de estudar de 1994, ainda no Colégio Técnico, até agora.

Às demais pessoas que direta ou indiretamente contribuíram na elaboração desta tese.

À Deus por ter me dado a vida e estar sempre me ajudando na escolha de meus caminhos.

MAIA, Luiz Cláudio Gomes. **USO DE SINTAGMAS NOMINAIS NA CLASSIFICAÇÃO AUTOMÁTICA DE DOCUMENTOS ELETRÔNICOS. 2008.** Tese (Doutorado em Ciência da Informação) – Universidade Federal de Minas Gerais – UFMG. Belo Horizonte, 2008.

Resumo

Esta pesquisa verificou se ocorre aprimoramento na classificação de documentos eletrônicos com o uso de técnicas e algoritmos de mineração de texto (análise de texto) utilizando além das palavras, sintagmas nominais como indexadores. Utilizaram-se duas ferramentas nos experimentos propostos desta pesquisa o OGMA e a WEKA. O OGMA foi desenvolvido pelo autor para automatizar a extração dos sintagmas nominas e o cálculo do peso de cada termo na indexação dos documentos para cada um dos seis métodos propostos. A WEKA foi utilizada analisar os resultados encontrados pelo OGMA utilizando aos algoritmos de agrupamento e classificação, simplekmeans e NaiveBayes, respectivamente, obtendo um valor percentual indicando quantos documentos foram classificados corretamente. Os métodos com melhores resultados foram o de termos sem *stopwords* e o de sintagmas nominais classificados e pontuados como descritores.

Palavras-chave: análise de texto, agrupamento automático de documentos, indexação automática, sintagmas nominas.

MAIA, Luiz Claudio Gomes. **USE OF NOUN PHRASES IN AUTOMATIC CLASSIFICATION OF ELECTRONIC DOCUMENTS.** 2008. Dissertation (Ph.D. in Information Science) - Federal University of Minas Gerais - UFMG. Belo Horizonte, 2008.

Abstract

This research work presents a proposal for the classification of electronic documents using techniques and algorithms based on natural language processing and noun phrases indexing along with plain keywords. Two tools, OGMA and Weka, were used for the experiments proposed. OGMA was developed by the author to automate the extraction of noun phrases and to perform the calculation of the weight of each term in the process of document indexing for each of the six proposed methods. The WEKA was used to analyze the OGMA results using the algorithms of clustering and classification "Simplekmeans" and "NaiveBayes", respectively. This process resulted in a percentage value indicating how many documents were classified correctly. The best performing methods were those with the terms without stopwords and the classified and scored noun phrases.

Keywords: text analysis, clustering, automatic indexing, noun phrases, natural language processing.

Lista de Figuras

FIGURA 1 – INFORMAÇÃO DIGITAL CRIADA, CAPTURADA E REPLICADA DE 2006 A 2010. 19
FIGURA 2 – FASES DO PROCESSAMENTO DO DOCUMENTO PARA SUBMISSÃO À INDEXAÇÃO. 33
FIGURA 3 – EXEMPLO DE UTILIZAÇÃO DO VISL-PALAVRAS. 41
FIGURA 4 – EXEMPLO DE UTILIZAÇÃO DO GRAMMARPLAY. FONTE: TELA DO PROGRAMA. 42
FIGURA 5 – DIAGRAMA EM ÁRVORE FORNECIDO PELO GRAMMAR PLAY. 43
FIGURA 6 – EXEMPLO DE UTILIZAÇÃO DO LX-TAGGER. 44
FIGURA 7 – ANALISADOR CURUPIRA. 45
FIGURA 8 – REPRESENTAÇÃO DO MÉTODO ED-CER. 47
FIGURA 9 – PROCESSO DE AGRUPAMENTO DE DOCUMENTOS POR ASSUNTO. 54
FIGURA 10 – FIGURA DE OGMA. 61
FIGURA 11 – TELA DO OGMA. 62
FIGURA 12 – DICIONÁRIO DO OGMA: TABELA DE NOMES E ADJETIVOS. FONTE: ELABORADA PELO AUTOR DESTA TESE. 62
FIGURA 13 – DICIONÁRIO DO OGMA: TABELA DE VERBOS. 63
FIGURA 14 – DICIONÁRIO DO OGMA: TABELA GRAMATICAL. 64
FIGURA 15 – RESULTADO DO CÁLCULO DA SIMILARIDADE ENTRE DOIS DOCUMENTOS PELO OGMA. 70
FIGURA 16 – TELA DO MÓDULO *WEKA EXPLORER*. FONTE: *WEKA EXPLORER*. 72
FIGURA 17 – ETAPAS DO EXPERIMENTO PROSPECTIVO. 75
FIGURA 18 – ETAPAS DO EXPERIMENTO CONSOLIDADO. 77
FIGURA 19 – ARQUIVO DE LOTE (BAT) PARA REALIZAR A MAIORIA DAS ETAPAS DO EXPERIMENTO AUTOMATICAMENTE. 80
FIGURA 20 – RELAÇÃO DE ATRIBUTOS E TIPOS ARQUIVO ARFF. 81
FIGURA 21 – DADOS DO ARQUIVO ARFF. 81
FIGURA 22 – RESULTADOS WEKA PARA O CORPUS ENANCIB05. 91
FIGURA 23 – NÚMERO MÉDIO DE DESCRITORES POR DOCUMENTO, CORPUS JORNAIS04. FONTE: EXPERIMENTO. 93
FIGURA 24 – RESULTADOS WEKA PARA O CORPUS JORNAIS04. 95

Lista de Quadros

QUADRO 1 – EVOLUÇÃO ATRAVÉS DO TEMPO DO SUPORTE À INFORMAÇÃO. 21
QUADRO 2 – PADRÕES DE METADADOS ORIGINADOS DA BIBLIOTECONOMIA. 27
QUADRO 3 – FCS NA GESTÃO DA PRECISÃO NO PROCESSO DE BUSCA E RECUPERAÇÃO DA INFORMAÇÃO. 30
QUADRO 4 – EXEMPLO DE REPRESENTAÇÃO VETORIAL: TERMOS X DOCUMENTOS. 32
QUADRO 5 – CATEGORIAS GRAMATICAIS DAS FUNÇÕES DE LIBERATO. 39
QUADRO 6 – ANALISADORES LÍNGUA PORTUGUESA. 46
QUADRO 7 – ETIQUETAS (CLASSES GRAMATICAIS) DO MÉTODO ED-CER. 47
QUADRO 8 – SÍMBOLOS TERMINAIS DO MÉTODO ED-CER. 48
QUADRO 9 – SÍMBOLOS NÃO TERMINAIS. 49
QUADRO 10 – GRAMÁTICA DO MÉTODO ED-CER. 49
QUADRO 11 – PONTUAÇÃO DA AVALIAÇÃO DOS SNS COMO DESCRITORES. 50
QUADRO 12 – CLASSIFICAÇÃO DOS SINTAGMAS NOMINAIS (CSN). 51
QUADRO 13 – MELHORES VALORES ENCONTRADOS COMO PESO PARA CADA CSN. 51
QUADRO 14 – PRINCIPAIS MÉTRICAS DE SIMILARIDADE. 59
QUADRO 15 – ETIQUETAS UTILIZADAS NO OGMA E NO ED-CER. 64
QUADRO 16 – REGRAS DE EXTRAÇÃO DE SN DO MÉTODO OGMA. 66
QUADRO 17 – COMPARAÇÃO ENTRE EXTRAÇÃO DE SN PELO ED-CER, OGMA E VISL. 71
QUADRO 18 – COMPOSIÇÃO DO CORPUS ENANCIB05. 73
QUADRO 19 – COMPOSIÇÃO DO CORPUS JORNAIS04. 74
QUADRO 20 – SIGLAS DOS MÉTODOS DE ANÁLISE UTILIZADOS NOS EXPERIMENTOS. 83

Lista de Tabelas

TABELA 1 – NÚMERO MÉDIO DE DESCRITORES POR DOCUMENTO E MÉTODO, *CORPUS* ENANCIB05 84
TABELA 2 – SIMILARIDADE ENTRE DOCUMENTOS *CORPUS* ENANCIB05 84
TABELA 3 – COMPARAÇÕES ENTRE O VALOR DE SIMILARIDADE *CORPUS* ENANCIB05 85
TABELA 4 – RESULTADO AGRUPAMENTO MÉTODO TT, *CORPUS* ENANCIB05 86
TABELA 5 – RESULTADO AGRUPAMENTO MÉTODO TTS, *CORPUS* ENANCIB05 87
TABELA 6 – RESULTADO AGRUPAMENTO MÉTODO TC, *CORPUS* ENANCIB05 88
TABELA 7 – RESULTADO AGRUPAMENTO MÉTODO TR, *CORPUS* ENANCIB05 89
TABELA 8 – SOMA DA DIFERENÇA ENTRE O GT CORRETO 90
TABELA 9 – RESULTADO WEKA PARA O *CORPUS* ENANCIB05 91
TABELA 10 – COMPARAÇÃO EXPERIMENTO PROSPECTIVO X EXPERIMENTO CONSOLIDADO 92
TABELA 11 – NÚMERO MÉDIO DE DESCRITORES POR DOCUMENTO E MÉTODO, CORPUS JORNAIS04 93
TABELA 12 – RESULTADO WEKA/*NAIVE BAYES*, CORPUS JORNAIS04 94
TABELA 13 – RESULTADO WEKA/*SIMPLEKMEANS*, CORPUS JORNAIS04 94

Lista de símbolos e abreviaturas

ARFF - *Attribute-Relation File Format*
CG - *Constraint Grammar*
CORA - *Computer Science Research Paper Search Engine*
CPU - Unidade Central de Processamento
CSN – Classe do Sintagma Nominal
ENANCIB - Encontro Nacional de Pesquisa em Ciência da Informação
FAQ - *Frequently Asked Questions*
GPL - *General Public License*
GT - Gramática transformacional
HTML - *Hyperlink Text Markup Language*
IDF - *Inverse Document Frequency*
LSA - Análise Semântica Latente
LSI - *Latent Semantic Indexing*
KDT - *Knowledge Discovery from Text*
MARC - *Machine Readable Cataloging*
MIT - *Massachusetts Institute of Technology*
NDLTD - *Networked Digital Library of Theses and Dissertations*
NIST - *National Institute of Standards and Technology*
PDF – *Portable Document Format*
PLN – Processamento de Linguagem Natural
PPGCI – Programa de Pós Graduação em Ciência da Informação
RAM - Memória de acesso aleatório
RDF - *Resource Description Framework*
RI - Recuperação da Informação
SAdj - Sintagma Adjetival
SAdv - Sintagma Adverbial
SN - Sintagma Nominal
SOIF - *Summary Object Interchange Format*
SP - Sintagma Preposicional
SRI - Sistema de Recuperação da Informação
SRPTV - Sistema de Recomendação Personalizada de Programas de TV
SV - Sintagma Verbal
SVD - *Simple Value Decomposition*
TDE – Teses e Dissertações Eletrônicas
TEI - *Text encoding for Information Interchange*
TF - *term frequency*
TF-IDF - *term frequency–inverse document frequency*
TI - Tecnologias de Informação
VISL - *Visual Interactive Syntax Learning*
VSM - *Vector Space Model*
WEKA - Waikato Environment for Knowledge Analysis
W3C - *Wold Wide Web Consortium*

Sumário

Resumo ..... **vii**
Lista de Quadros ..... **x**
Lista de Tabelas ..... **xi**
Lista de símbolos e abreviaturas ..... **xii**
1 INTRODUÇÃO **15**
1.1 Organização deste trabalho **16**
1.2 Motivação **17**
1.3 Objetivos da pesquisa **18**
2 REFERENCIAL TEÓRICO **19**
2.1 Recuperação da Informação **19**
2.1.1 Sistemas de recuperação da informação 19
2.1.2 Representação e armazenamento da informação 20
2.1.3 Recuperação da informação e a necessidade de informação do usuário 23
2.1.4 Metadado 25
2.1.5 Descritores 27
2.2 Conceitos básicos sobre análise de texto **29**
2.2.1 Construção e armazenamento do índice 30
2.2.2 Modelo booleano 31
2.2.3 Modelo vetorial 32
2.2.4 Índice por peso (TF-IDF) 33
2.3 Processamento de linguagem natural **34**
2.3.1 Gramática gerativa 36
2.3.2 Analisadores do português e a extração de sintagmas nominais 40
2.3.3 O método ED-CER 46
2.3.4 Escolha de descritores utilizando sintagmas nominais 50
2.4 Similaridade de documentos eletrônicos **52**
2.4.1 Classificação de documentos 53
2.4.2 Conglomerados 54
2.4.3 Medidas de similaridade em documentos eletrônicos 55
3 METODOLOGIA E REALIZAÇÃO DA PESQUISA **61**
3.1 Ferramentas **61**
3.1.1 OGMA: ferramenta para análise de texto 61
3.1.2 WEKA - Waikato environment for knowledge analysis 71
3.2 *Corpora* – coleção de teste **72**

3.3 Experimento prospectivo **75**
3.4 Experimento consolidado **77**
3.5 Etapas da pesquisa **78**
3.5.1 Extração dos sintagmas nominais 78
3.5.2 Construção das tabelas de descritores e pesos 79
3.5.3 Cálculo da similaridade entre as tabelas 79
3.5.4 Automação das etapas pelo OGMA 80
3.5.5 Arquivo ARFF 80
3.5.6 Geração da matriz e conversão para utilização no software WEKA 81
3.5.7 Aplicação dos algoritmos de classificação/Naive Bayes e agrupamento/simplekmeans 82
4 ANÁLISE DOS RESULTADOS **83**
4.1 Experimento prospectivo **83**
4.2 Experimento consolidado **90**
5 CONSIDERAÇÕES FINAIS **97**
REFERÊNCIAS **99**
ANEXOS **104**
I – Comparativo: Extração de SN Ogma x VISL 104
II – Enumeração dos artigos que formam o corpus ENANCIB 2005. 110
III – Lista de *stopwords* utilizada pelo OGMA. 115
IV – Exemplos de utilização do OGMA 118
V – Resultados do programa WEKA – Classificação/Naive Bayes 121
VI – Resultados do programa WEKA – Agrupamento/SimpleKMeans 131
VII – Número descritores extraídos de cada documento por método 141
VIII – Algoritmos do OGMA 146

1 INTRODUÇÃO

Há muitos anos o homem tem armazenado, catalogado e organizado a informação, com o principal objetivo de recuperá-la para uso posterior, sendo recentemente a área da biblioteconomia a responsável por ajudar o homem nesse processo.

Entretanto, vivencia-se com o advento das redes de comunicação uma crescente preocupação com essas formas de tratamento e organização da informação, pois cresce de maneira cada vez mais rápida a quantidade dos textos armazenados em formato digital. A maioria deles é esquecida, pois nenhum ser humano pode ler, entender e sintetizar todo esse volume informacional.

Estudos (JANSSENS, 2007 e IDC, 2007) apontam que no ano de 2007 existiam aproximadamente 550 bilhões de documentos somente on-line, com aproximadamente 7,5 petabytes[1] entre *websites* e base de dados. Para armazenar 7,5 petabytes, em uma pilha de páginas de papel, onde cada página conteria 2500 caracteres, sendo que um byte equivale a 1 caractere, teríamos uma pilha de 300.000km (1 cm para 100 páginas) o que daria para alcançar a lua ou dar a volta na terra 7,5 vezes. Uma pessoa lendo uma página por minuto gastaria 5.7 bilhões de anos para ler tudo.

Isso tem incentivado os pesquisadores a explorar estratégias para tornar acessível ao usuário a informação relevante (BAEZA-YATES e RIBEIRO-NETO, 1999; FRANTZ, SHAPIRO e VOISKUNSKII, 1997; MEADOW, BOYCE e KRAFT, 2000; CROFT, 2000).

Um dos objetivos da linha de pesquisa denominada Recuperação da Informação (RI) do Programa de pós-graduação em ciência da informação é o de estudar e propor modelos que possibilitem às pessoas uma seleção mais rápida da informação necessária. Os problemas tratados pelos sistemas de recuperação da informação (SRI) estão relacionados a coletar, representar, organizar e recuperar documentos entre outras formas de representação da informação.

Com todo esse volume informacional somente o tratamento do texto dos documentos já não constitui garantias de uma recuperação eficiente. O aprimoramento e também novos estudos envolvendo a análise pelo computador da linguagem humana são necessários para melhorar a recuperação desses documentos. A análise realizada pelo computador da linguagem humana é conhecida como processamento da linguagem natural (PLN).

O processamento da linguagem natural (PLN) "*trata computacionalmente os diversos aspectos da comunicação humana*" (GONZALEZ e LIMA, 2003) e, esta pesquisa objetivou

[1] Um petabyte equivale a $10x^{15}$ bytes.

verificar se ocorre aprimoramento em medidas de similaridade entre documentos eletrônicos com o uso de técnicas de processamento de linguagens naturais.

Para alcançar com êxito o objetivo desta tese foi necessária a criação de uma ferramenta computacional que adaptasse e complementasse os modelos propostos por três pesquisas importantes em ciência da informação, computação e lingüística (SOUZA, 2005; MIORELLI, 2001; PERINI, 1996). Essa ferramenta computacional foi denominada Ogma[2] e trata-se de um programa que permitiu realizar todos os cálculos e análises textuais necessárias para a execução do experimento proposto.

1.1 Organização deste trabalho

O trabalho estrutura-se da seguinte forma:

Neste primeiro capitulo, após a introdução, temos subseções que são: (a) a motivação, que mostrou a importância e a relevância do estudo; e (b) o objetivo geral e os objetivos específicos, que sintetizava o que se pretendia alcançar com a pesquisa.

No segundo capítulo, a revisão de literatura apresenta uma discussão sobre a evolução dos sistemas de recuperação da informação, uma introdução às técnicas tradicionais das áreas de recuperação da informação, processamento de linguagem natural e análise da similaridade e classificação de documentos eletrônicos, além de conceitos necessários para melhor entendimento desta pesquisa.

No terceiro capítulo, a metodologia de pesquisa utilizada neste estudo é explicada, caracterizando as etapas da pesquisa, o experimento realizado bem como o corpus utilizado.

Também no terceiro capítulo foi apresentada em detalhes a descrição do desenvolvimento da ferramenta Ogma, elemento importante desta pesquisa que possibilitou a automatização do experimento proposto.

A apresentação e a análise dos resultados foram descritas no quarto capítulo. Seguem-se as considerações finais no quinto capítulo.

Por fim, encontram-se os anexos, que foram compostos pelos resultados do experimento e outros itens resultantes da pesquisa.

[2] O nome Ogma foi dado em homenagem ao deus celta Ogma (nome reduzido de Ogmious). Esse Deus criou mecanismos de linguagem e engrandeceu a comunicação do povo celta.

1.2 Motivação

A criação e o desenvolvimento de mecanismos que auxiliem as pessoas na busca por uma informação precisa são contínuos e devem ser cada vez mais incentivados pela ciência.

As ciências humanas e sociais consideram a escrita como condição básica e necessária para o surgimento e o desenvolvimento do pensamento lógico e racional; A ciência da informação em seus primórdios relacionava-se com o registro físico da informação, e já se percebia a sua importância na evolução da humanidade que passou a registrar, estocar e recuperar as informações que potencializam o conhecimento (BARRETO, 2008).

A partir do desenvolvimento tecnológico acontecido na década de 1960, o tratamento da informação passou a ser visto com olhar diferenciado pela comunidade científica. No período compreendido entre 1961 e 1962, surgiu a primeira formulação do conceito da ciência da informação, resultado das conferências do Instituto Tecnológico da Geórgia, nos Estados Unidos, mais conhecido como "Georgia Tech":

> "Ciência da informação é a ciência que investiga as propriedades e comportamento da informação, as forças que regem o fluxo da informação e os meios de processamento da informação para uma acessibilidade e usabilidade ótimas. Os processos incluem a origem, disseminação, coleta, organização, recuperação, interpretação e uso da informação. O campo deriva de ou relaciona-se com a matemática, a lógica, a lingüística, a psicologia, a tecnologia da computação, a pesquisa operacional, as artes gráficas, as comunicações, a biblioteconomia, a administração e alguns outros campos." (SHERA & CLEVELAND, 1977)

Saracevic (1996) apresentou o seguinte conceito sobre ciência da informação, no qual ressalta suas características, tanto científica quanto aplicada:

> "Ciência da Informação é um campo dedicado às questões científicas e à prática profissional voltadas para os problemas da efetiva comunicação do conhecimento e de seus registros entre os seres humanos, no contexto social institucional ou individual do uso e das necessidades de informação. No tratamento destas questões são consideradas de particular interesse as vantagens das modernas tecnologias informacionais." (SARACEVIC, 1996)

O desenvolvimento de um modelo que utilize conceitos elaborados através da reunião de diversos estudos recentes sobre recuperação de documentos – que, na denominação de SARACEVIC, podem ser chamados de "modernas tecnologias informacionais" – é de grande necessidade para a evolução da área ciência da informação.

De forma geral, pode-se justificar a presente pesquisa nos seguintes termos:

- Na necessidade de ferramentas mais eficazes para organização de informação, devido ao crescimento contínuo do volume dessas.
- Nas novas possibilidades que se afiguram para a busca de conteúdo em documentos eletrônicos similares, através de critérios que privilegiem a relevância.

1.3 Objetivos da pesquisa

O objetivo geral desta pesquisa foi:

- Investigar a utilização de sintagmas nominais pontuados como elementos de classificação por similaridade e aglomerados de documentos eletrônicos.

Para a realização desse objetivo pudemos relacionar os seguintes objetivos específicos:

a) Realizar a extração automatizada de SN de documentos eletrônicos através do desenvolvimento de uma ferramenta específica utilizando uma adaptação do método ED-CER (MIORELLI, 2001) e do modelo proposto por PERINI (1996).
b) Automatizar o cálculo da pontuação de cada sintagma nominal extraído como possível descritor do documento, utilizando a metodologia proposta por SOUZA (2005).
c) Analisar o resultado da aplicação de medidas de similaridade, utilizando técnicas de mineração de texto em comparação com a técnica de extração de SN pontuados, em determinado *corpus*.

2 REFERENCIAL TEÓRICO

Neste capítulo, são apresentados os marcos teóricos necessários para o entendimento dos experimentos propostos nesta tese.

2.1 Recuperação da Informação

Durante os últimos anos, um volume crescente de informações tem sido registrado em várias bases de dados, nos mais diversos domínios do conhecimento e sob diversas formas (numéricas, textuais, imagens etc.). É uma verdadeira coleção globalizada e interligada de informações representadas através de diversos modelos.

Um relatório elaborado pela empresa IDC (2007) estima que o tamanho do volume de informação digital – informação que é criada ou capturada em meio digital e replicada – foi em 2006 de 161 exabytes, e crescerá em 2010 para 988 exabytes, numa proporção de crescimento anual de 57%. Conforme ilustrado na figura abaixo:

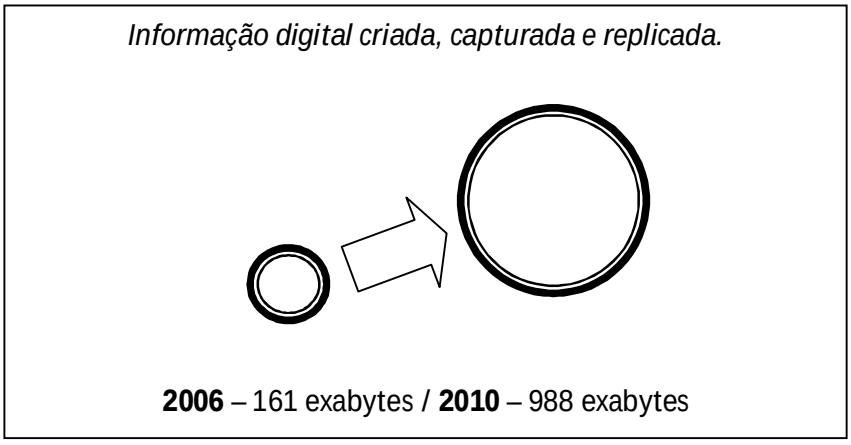


FIGURA 1 – Informação digital criada, capturada e replicada de 2006 a 2010.
Fonte: adaptado de IDC, 2007.

2.1.1 Sistemas de recuperação da informação

Não existe um consenso relativo à conceituação do termo sistema de recuperação da informação (ARAUJO, 1995). O termo pode assumir desde conceitos mais abrangentes até conceitos mais específicos e práticos. Nesta pesquisa tomaram-se os que são trabalhados nos marcos teóricos da recuperação da informação: as definições de Harter (1996), Salton (1968) e de Baeza-Yates e Ribeiro-Neto (1999).

Harter define um sistema de recuperação de informação (SRI) como "*um dispositivo que intermedia a comunicação entre os usuários e a coleção de informação*". Já Salton e McGill (1983) conceituam tecnicamente um SRI como *"um sistema que trata da*

representação, do armazenamento, da organização e do acesso aos itens de informação". E Baeza-Yates e Ribeiro-Neto (1999) definem SRIs como "*sistemas que lidam com as tarefas de representação, armazenamento, organização e acesso aos itens de informação*".

De acordo com Souza (2005, p.29) os SRI organizam e viabilizam o acesso aos itens de informação, desempenhando as atividades de:

- **representação** das informações contidas nos documentos, usualmente através dos processos de indexação e descrição dos documentos;
- **armazenamento** e gestão física e/ou lógica desses documentos e de suas representações;
- **recuperação** das informações representadas e dos próprios documentos armazenados, de forma a satisfazer as necessidades de informação dos usuários. Para isso é necessário que haja uma *interface* na qual os usuários possam descrever suas necessidades e questões, e também possam analisar os resultados recuperados pelo SRI.

2.1.2 Representação e armazenamento da informação

Ao longo da história, os SRI desenvolvidos passaram por várias alterações. Na medida em que foram surgindo novas formas de saberes e de conhecimento, na sociedade e no pensamento humano, aflorou a necessidade de novas expressões e formas de representação, o que conseqüentemente implicou diretamente a alteração das estruturas de tais modelos de recuperação e análise.

Os SRIs dependem, então, não só dos saberes e conhecimentos de cada época mas também das formas de representação da informação e suas superfícies, sendo esta última possibilitada pelas tecnologias de informação (TI). A TI tem intervalos de evolução cada vez menores e possibilita novas superfícies de registro da informação com maior capacidade de armazenamento e distribuição. No QUADRO 1, a seguir, pode-se observar a evolução dos diversos meios de armazenamento e distribuição da informação em relação à data de invenção.

QUADRO 1 – Evolução através do tempo do suporte à informação.

SUPORTE	DATA	INVENTOR	PAÍS
Alfabeto	1500a.C.	Fenícios	Fenícia
Correio	VI a.C		
Papel	105 a.C.	Tsai Lun	China
Códex	400		
Impressora	1440	Johann Gutenberg	Alemanha
Máquina de escrever	1714	Henry Hill	Grã-Bretanha
Litografia	1796	Aloys Senenfelder	Alemanha
Calculadora digital	1823	Charles Babbage	Grã-Bretanha
Fotografia	1827	Joseph Nièpse	França
Telefone	1876	Graham Bell	EUA
Fonógrafo	1877	Thoma Edison	EUA
Microfone	1878	David Hughes	EUA
Linotipo	1884	O. Mergenthaler	EUA
Gramofone	1887	Emile Berliner	EUA
Antena	1895	Aleksander Pópov	Rússia
Cinema	1895	Irmãos Lumiére	França
Telégrafo	1895	Guglielmo Marconi	Itália
Televisão	1923	V.K. Zworykin	EUA
Fax	1929	Rudolf Hell	Alemanha
Radar	1935	R. Watson-Watt	Grã-Bretanha
Telefone móvel	1940	Hedwig Kiesler	Áustria
Computador (Eniac)	1946	Eckert e Manchly	EUA
Foto copiadora	1948	Chester Carison	EUA
Holografia	1949	Denis Gabor	Grã-Bretanha
Arpanet/Internet	1969/1972		EUA
Microcomputador (Apple II)	1973/1977	Soc. RE/Steven Jobs	França/EUA
www	1994	Tim Berners Lee	Suíça
CD	1979	Joop Sinjou	Holanda
DVD	1996	WB	Japão e EUA
HTDV	1996	Zenith	EUA

Fonte: adaptado de SIMEÃO, 2003, pelo autor desta tese.

Realizando uma retrospectiva percebe-se o quanto o homem, ao descobrir uma nova tecnologia para transmissão e armazenagem da informação, demora a se adaptar e a explorar todas as potencialidades do novo meio.

Em 1440 Gutenberg, ao aperfeiçoar a imprensa de tipos móveis, não causou, no primeiro momento, grande revolução da imprensa na Europa, e o que foi produzido nos 50 anos posteriores foram os mesmos textos que os monges costumavam copiar à mão (VILLAÇA, 2002, p. 47). O intervalo entre a imprensa de tipos móveis de Gutenberg e a linotipo de Mergenthaler foi de mais de 400 anos. Conforme McGarry:

> "A tecnologia da imprensa se manteve inalterada durante 360 anos até que, em 1814 o jornal Times (de Londres), utilizando a prensa Koening, avançou na reprodutibilidade e disseminação e passou a copiar 1100 exemplares por hora, diminuindo os custos e aumentando a distribuição. Isso era apenas o começo: o princípio da rotativa aliado aos rolos de papel, tornados possíveis com a máquina de fazer papel de Foutdriner, que iria lançar as bases da impressão de jornais." (McGARRY[3] citado por SIMEÃO, 2003, p.37)

Em contrapartida, para exemplificar a diminuição dos intervalos, em maio de 1994 as empresas Sony e Philips anunciaram que iriam trabalhar cooperativamente no desenvolvimento de uma nova mídia de alta densidade, conhecida hoje em dia como DVD. Essa tecnologia foi lançada em 1996 com o objetivo de substituir os CDROMs para computadores de 1979, e o VHS de 1980. O intervalo entre a tecnologia anterior e a atual caiu para 16 anos. Além disso, a capacidade de armazenamento aumentou bastante; o DVD é um disco do mesmo tamanho e formato que um CDROM, só que ele armazena 10 vezes mais dados.

As novas tecnologias possibilitaram muitas vantagens à recuperação da informação. Nota-se que, quando os SRIs oferecem novos canais de distribuição, seleção e armazenagem, o custo para utilização também se torna menor. Atualmente, graças à codificação digital e às redes de telecomunicação, uma informação pode cruzar de um canto a outro do planeta de forma instantânea, sendo que esta mesma velocidade não seria possível com a informação codificada em papel e tendo como rede os correios.

Mesmo com todas as possibilidades abertas à publicação e à divulgação da informação por novas tecnologias de comunicação, criou-se, ou mesmo se mantém uma dificuldade na codificação dessas informações. Villaça (2002, p.51) aponta um dos motivos: "o pensar e escrever humano não é binário". E ainda não existem tecnologias que permitam a codificação do pensar. Conforme Villaça o pensamento humano:

---

[3] MCGARRY, K.J. **Da documentação à informação:** um contexto em evolução. Lisboa: Presença, 1984.

> “ não trabalha com unidades de informação apenas, mas por figurações intuitivas e hipotéticas. A decepção com o “pensamento sem corpo” das inteligências artificiais provém do fato de as operações serem efetivadas em lógica binária, aquela que se impôs com a lógica matemática de Russell e Whitehead, a máquina de Turing, o modelo neuronal de McCulloc e Pitts, a cibernética de Wiener e Von Neumann, a álgebra de Boole ou a informática de Shannon.” (VILLAÇA, 2002, p. 51)

Apesar de ainda não ser possível representar o pensamento humano em toda sua amplitude, tecnologias como a Internet, mas especificamente a *Web*, possibilitam novas formas de representação da informação. Um texto disponibilizado na Web não necessita seguir uma ordem tradicional de início e fim. Através dos *hyperlinks* o texto pode ganhar um novo fluxo de acordo com o interesse ou necessidade do usuário.

Assim, as possibilidades de representação da informação, antes limitadas a signos fixos em uma folha de papel, agora se rendem às possibilidades de uma nova interface. Uma superfície volátil onde seja possível construir um conteúdo não linear, o hipertexto, passa, por formas de exploração de recursos áudio-visuais não possíveis no papel. Levy cita o hipertexto como facilitador de aprendizagem:

> “A multimídia interativa ajusta-se particularmente aos usos educativos. [...] quanto mais ativamente uma pessoa participar da aquisição de um conhecimento, mais ela irá integrar e reter aquilo que aprender. Ora, a multimídia interativa, graças à sua dimensão reticular ou não-linear, favorece uma atitude exploratória.”. (LEVY, 1999, p. 40)

Gutemberg proporcionou um meio de criar exemplares idênticos. Em contrapartida, com o hipertexto existe a possibilidade de durante a leitura se construirem versões diferentes de um mesmo exemplar. Mueller e Passos fazem uma comparação do fluxo do texto impresso com o fluxo do texto eletrônico através do hipertexto:

> “Um livro contém uma quantidade limitada de informação que só pode ser extraída por um número limitado de maneiras. A informação eletrônica equivalente pode ser ligada a uma quantidade virtualmente ilimitada de informação adicional, à qual o usuário pode ter acesso de várias maneiras, de acordo com o seu desejo.” (MUELLER e PASSOS, 2000, p.25)

Diante dessas novas possibilidades proporcionadas pelas transformações tecnológicas pelas quais passamos, uma coisa é certa: vivemos em *uma* época limite na qual toda a antiga ordem das representações e dos saberes oscila para dar lugar a imaginários modos de representação do conhecimento ainda pouco estabilizados.

2.1.3 Recuperação da informação e a necessidade de informação do usuário

Os primeiros estudos sobre avaliação de SRIs relacionavam que o nível de satisfação do usuário com o resultado apresentado pelo sistema estava interligado com a porcentagem de

resultados relevantes obtidos por sua pesquisa. Como exemplo, tem-se em 1950 o modelo de Cranfield, como um dos primeiros métodos de avaliação de SRI.

Ainda na década de 50 a relevância e a revocação (*recall*) eram as medidas responsáveis pela avaliação dos SRIs. Posteriormente verificou-se a confusão com o uso do termo "relevância" e passou-se a utilizar "precisão" no lugar de "relevância" (SARACEVIC, 2007). Ainda na década de 60, Salton (1968) afirma que o princípio de todo e qualquer SRI é aumentar a precisão (*precision*) e a revocação (*recall*); essas medidas são representadas da seguinte maneira:

$$precisão = \frac{número\ de\ doc.\ pertinentes\ recuperados}{número\ de\ documentos\ recuperados}$$

$$revocação = \frac{número\ de\ doc.\ pertinentes\ recuperados}{número\ total\ de\ documentos\ pertinentes}$$

Mas quais seriam as medidas e os componentes que deveriam ser considerados na avaliação de um SRI? Os usuários, suas cognições e seu ambiente de trabalho devem ser incluídos? (HARTER e HERT, 1997)

O significado de "avaliação" toma diferente forma em muitos contextos. Hernon e Maclure (citado por HARTER e HERT, 1997) definem a avaliação como o processo de identificar e coletar dados sobre serviços e atividades específicas, estabelecer critérios, possibilitando que o sucesso possa ser analisado, e determinar a qualidade do serviço, e o grau com que o serviço atingiu o objetivo e metas inicialmente traçadas.

Um usuário reconhece sua necessidade de informação. O usuário se dirige a um SRI com uma consulta, com base no que necessita. O sistema então compara a consulta com representações nos documentos do sistema. A intenção é que alguns ou todos os documentos apresentados satisfaçam, parcialmente ou totalmente, a necessidade de informação do usuário. (ELLIS citado por HARTER e HERT, 1997)

Saracevic (2007) discorre sobre as diversas interpretações para relevância e as dificuldades de aplicá-la na prática à recuperação da informação. Saracevic (2007) afirma que a "*relevância é uma noção humana e não de um sistema, e que noções humanas são extremamente complexas*". O autor realiza um panorama dos caminhos percorridos pelo conceito de relevância na ciência da informação e aponta o conceito "*como um conceito chave para a Ciência da Informação no geral, e para a Recuperação da Informação em particular*."

Nesse artigo o autor destaca as seguintes manifestações de relevância: (SARACEVIC, 2007)

- Relevância de sistema ou algoritmos: relação entre uma consulta e a informação ou os objetos. Cada sistema tem as formas e os meios pelos quais os objetos fornecidos sejam representados, organizados, e associados a uma consulta.
- Relevância de tópico ou tema: relação entre os assuntos ou temas expressa em uma consulta por tema ou assunto abrangido, a informação ou os objetos. Assumiu-se que ambas as consultas e objetos podem ser identificados como sendo necessário sobre um tema ou assunto.
- Relevância cognitiva ou pertinência: relação entre o estado cognitivo dos conhecimentos de um usuário e informações ou objetos.
- Relevância situacional ou utilidade: relação entre a situação, tarefa ou problema e as informações e objetos.
- Relevância afetiva: relação entre as intenções, metas, emoções, e as motivações de um usuário e informações.

A manifestação de relevância, dentro dos critérios estabelecidos por Saracevic (2007), utilizada nesta pesquisa é a relevância por tópico ou tema. A justificativa é que o presente estudo tem a preocupação com a tematicidade (*aboutness*) de documentos eletrônicos.

O autor também destaca que, por muito tempo, a biblioteconomia se preocupou em trabalhar a questão da tematicidade de um documento e se preocupava pouquíssimo com a relevância do mesmo. Em contrapartida a ciência da informação possui extensa literatura sobre relevância e pouquíssima sobre tematicidade.

2.1.4 Metadado

O metadado corresponde a um conjunto de informações que descreve o documento. Trata-se, em outras palavras, de dados estruturados que descrevem as características de um recurso de informação. Sua correta criação e utilização são de extrema importância para a qualidade da informação e para o SRI.

Milstead e Feldman (1999) complementam esse conceito, definindo os principais propósitos: *"metadados são dados que descrevem propriedades de um recurso para diversos propósitos, como o contexto em que recurso se insere, sua qualidade, suas condições de uso, sua identificação, suas estratégias de preservação etc"*.

Apesar de o termo ter-se tornado mais popular recentemente, seu conceito já era utilizado há muito tempo em áreas como a biblioteconomia; e o padrão *machine readable cataloging* (*MARC*)[4] é um dos exemplos. O principal objetivo do MARC é servir como formato padrão para o intercâmbio de registros bibliográficos e catalográficos além de servir de base para a definição de formatos de entrada entre as instituições que o utilize. Desde a década de 60 o padrão é atualizado em versões e tem conseguido além de manter compatibilidade com os padrões anteriores, manter-se funcional.

Em relação à *web*, um dos problemas é que certos autores inflam os metadados de seus documentos para se ter uma visibilidade maior. Além disso, a maioria das páginas *html* quase não apresentam metadados, ou constituem apenas palavras chaves e descrição. (IRVIN, 2003)

Tem-se trabalhado em padrões para realizar a correta formulação dos metadados, como por exemplo, o *Schema for the electronic theses and dissertations metadata set* - ETD-MS e o *Dublin core*[5] (elaborado a partir do MARC) utilizados pela *Networked Digital Library of Theses and Dissertations (NDLTD).* O *Dublin core* também foi adotado como padrão pelo *Open Archives Initiative*[6] e D-Space[7] (GREENBERG, 2004). Porém quando se amplia o universo das bibliotecas e bases digitais para toda a *web*, obtém-se que apenas 0,3% das páginas *html* contêm metadados no padrão *Dublin core*.[8] (LAWRENCE e GILES, 1999)

Ferramentas de extração e geração automática de metadados têm sido criadas e aprimoradas, podendo-se citar como exemplos a DC-Dot e a *Klarity*, que criam metadados no padrão Dublin core. (IRVIN, 2003, p. 9; GREENBERG, 2004) Isso tem gerado uma discussão sobre qual seria o metadado com uma qualidade melhor: o gerado através da análise manual (humana) ou através da análise, usando uma dessas ferramentas (computador). (ANDERSON e PEREZ-CARBALLO, 2001)

O padrão que representa uma convergência de conhecimentos provindos da computação e da ciência da informação é o padrão de metadados *resource description framework* (*RDF*) definido pelo *World Wide Web Consortium* (*W3C*) que vem trabalhando em definições baseadas na linguagem de marcação XML, também um produto da W3C. O *RDF* tem o objetivo de proporcionar o aperfeiçoamento da semântica na web. Originalmente,

---

[4] Ao final da década de 50 do século passado, a Library of Congress (LC) iniciou suas investigações sobre a possibilidade de automatizar suas operações, e durante as conferências sobre catálogos mecanizados realizadas a partir de 1965, a LC apresentou trabalhos discutindo o formato do MARC. Em 1968, a LC encerrando suas atividades publicou um relatório sobre o conjunto dos caracteres gráficos, o formato MARC II, e sobre suas experiências. Para mais informações site: http://www.loc.gov/marc

[5] Para mais informações site: http://www.dublincore.org

[6] Para mais informações site: http://www.openarchives.org

[7] Para mais informações site: http://www.dspace.org

[8] O estudo envolveu cerca de 2500 servidores web.

o *RDF* foi definido como um padrão de metadados; entretanto hoje é mais utilizado para modelar informações em diversas sintaxes. O RDF teve sua origem a partir de idéias de um grupo de discussão sobre Dublin core. No quadro a seguir estão relacionados os padrões de metadados com origem na biblioteconomia.

Lourenço (2005) realizou uma compilação desses padrões originados da biblioteconomia. Os padrões estão relacionados no QUADRO 2:

QUADRO 2 – Padrões de metadados originados da biblioteconomia.

PADRÃO	RESPONSÁVEL	ANO
MARC - *Machine Readable Cataloging Record*	*Library of Congress (LC)*	1960
Dublin Core	OCLC	1968
GILS - *Government Information Location Service*	*National Archives dos EUA*	1992
EAD - *Encoded Archival Description*	Universidade da Califórnia	1993
RDF - *Resource Description Framework Schema*	W3C - *World Wide Web Consortium*	1998

Fonte: Adaptado de LOURENÇO, 2005.

Existem diversos outros padrões de metadados não originados da biblioteconomia. Lourenço (2005) também cita alguns exemplos: *Text encoding for information interchange (TEI) da* Associação de Computadores e Humanidades; Internet anonymous Ftp Archive (AIFA); *Summary object interchange format* (SOIF) da Universidade do Colorado e mesmo a *tag* de metadados contidas em documentos HTML.

2.1.5 Descritores

Para que um sistema de recuperação de informação possa responder às demandas dos usuários com tempos de respostas aceitáveis, é preciso que os documentos constantes na base de dados sejam submetidos a um tratamento prévio. Esse procedimento permite a extração dos descritores e sua estruturação com vistas a um acesso rápido às informações.

O processo de indexação produzindo uma lista de descritores visa à representação dos conteúdos dos documentos. Ou seja, esse processo tem como objetivo extrair as informações contidas nos documentos, organizando-as para permitir a recuperação destes últimos. Contudo, na maioria dos sistemas convencionais de recuperação de informação, os descritores não passam de uma simples lista de palavras extraídas dos documentos, que constituem a coleção.

De acordo com Ramsden (1974) o termo ‘linguagens naturais’ é comumente utilizado para denominar a linguagem falada e a linguagem escrita. É possível em indexação empregar

a linguagem natural simplesmente como é falada ou usada nos documentos sem tentar, por exemplo, controlar sinônimos ou indicar os relacionamentos entre os termos. Um índice feito dessa maneira chama-se índice de linguagem natural. Como alternativa ao índice de linguagem natural, pode-se usar uma linguagem artificial adaptada às necessidades do sistema de classificação, ou seja, uma linguagem de indexação. "*Esta linguagem refletirá em um vocabulário controlado para o qual foram tomadas decisões cuidadosas sobre os termos a serem usados, o significado de cada um e os relacionamentos que apresentam*." (RAMSDEN, 1974, pág. 3)

Existem contextos nos quais se pode utilizar uma linguagem de indexação: sistemas de classificação, listas de cabeçalhos de assunto, tesauros, etc; sendo que as linguagens consistem de um vocabulário controlado e uma sintaxe a ser seguida.

O processo de indexação visa à representação dos conteúdos dos documentos, produzindo uma lista de descritores. Ou seja, esse processo tem como objetivo extrair as informações contidas nos documentos, organizando-as para permitir a recuperação destes últimos. Assim, os descritores devem, na maior extensão possível, ser portadores de informação, de maneira a relacionar um objeto da realidade extralingüística com o documento que traz informações sobre esse objeto. Contudo, na maioria dos SRI convencionais, os descritores representam com muita limitação as informações presentes no documento.

Os termos isolados decorrem da análise do vocabulário e da sintaxe dos documentos a serem classificados e conseqüente extração e agrupamento dos termos que apresentam uma unidade semântica.

A utilização das palavras como representação temática de um documento, segundo Kuramoto (2002), não é o ideal, devido aos vários problemas encontrados nas propriedades lingüísticas das mesmas. Exemplificando, temos:

a) polissemia: a palavra pode ter vários significados. Exemplo: chave (solução de um problema; ferramenta para abertura de portas; e também ferramenta para apertar parafusos);

b) sinonímia: duas palavras podem designar o mesmo significado. Exemplo: abóbora e jerimum;

c) duas ou mais palavras podem combinar-se em ordem diferente designando idéias completamente diversas. Exemplo: crimes, juvenis, vítimas (vítimas de crimes juvenis; vítimas juvenis de crimes).

A partir dessas três propriedades, Kuramoto conclui que a polissemia e a combinação de palavras podem atuar no resultado de uma busca em um SRI aumentando a taxa de ruído. No caso de ocorrência de sinonímia, pode ocorrer o incremento da taxa de silêncio. A taxa de ruído e a taxa de silêncio correspondem a uma negação da taxa de precisão e taxa revocação já apresentadas. Temos:

$$taxa\ de\ ruído = \frac{número\ de\ doc.não\ pertinentes\ recuperados}{número\ total\ de\ documentos}$$

$$taxa\ de\ silêncio = \frac{número\ de\ documentos\ pertinentes\ não\ recuperados}{número\ total\ de\ documentos\ pertinentes}$$

2.2 Conceitos básicos sobre análise de texto

A análise de texto (*text analysis*) corresponde a uma área que envolve outras subáreas como, por exemplo, a mineração de texto (*text mining)* e a área de processamento de linguagem natural (PLN). A PLN também é uma subárea da inteligência artificial e da lingüística que estuda os problemas da geração e tratamento automático de línguas naturais.

A mineração de texto constitui na extração de informações sobre tendências ou padrões em grandes volumes de documentos textuais, em que uma amostra significativa de informações seja avaliada em textos contidos em bases textuais e em fontes de informação em linha. (POLANCO e FRANÇOIS, 2000)

A mineração de dados consiste em extrair informação de bancos de dados estruturados; já a mineração de texto, extrai informação de dados não estruturados ou semi-estruturados.

Existem ferramentas que realizam esse tratamento da informação não estruturada como o BR/Search. A aplicabilidade de ferramenta do tipo é enorme. O fabricante, Policentro TI S.A., cita exemplos de utilização: tribunais, câmaras legislativas, ministérios, assessorias de comunicação social, bibliotecas, centros de documentação, departamentos jurídicos, recursos humanos, etc., em particular, como soluções de armazenamento e recuperação de normas, procedimentos, legislação, jurisprudência, fiscalizações, FAQ, acervos, currículos, pareceres, acórdãos, correspondências, dossiês, prontuários, artigos, clipping de jornais, estatutos, regimentos e todo tipo de informação textual que necessite de um instrumento eficaz em termos de recuperação.

A ferramenta foi utilizada no estudo de Araújo Jr. (2006)[9] que trata da comparação entre indexação manual e ferramenta de mineração de textos, por meio da análise do índice de precisão de resposta nos processos de busca e recuperação da informação.

Após a aplicação do experimento, Araújo Jr. (2006) destaca a importância de como ferramentas do tipo podem auxiliar o processo de indexação:

> "Os dados analisados comprovam que o uso da ferramenta de mineração de textos na busca e recuperação da informação trará sempre como resposta maior quantidade de itens bibliográficos do que a lista de palavras-chave utilizadas na indexação manual." (ARÁUJO JR., 2006)

Aráujo Jr. (2006) aponta os fatores críticos de sucesso e os objetivos possibilitados pela aplicação da mineração de texto, bem como sua integração com a indexação. Esses objetivos estão relacionados no quadro abaixo:

QUADRO 3 – FCS na gestão da precisão no processo de busca e recuperação da informação.

FATORES CRÍTICOS DE SUCESSO (FCS)	OBJETIVOS-CHAVE
Aplicação da mineração de textos	Auxiliar o processo de indexação
Integração dos resultados obtidos com a mineração de textos à indexação	Montagem de uma sistemática de uso da mineração de textos no processo de indexação, com vistas ao aumento do índice de precisão nos processos de busca e recuperação da informação.

Fonte: ARAÚJO, 2006.

O autor também destaca a funcionalidade do software BR/Search que "*possibilita a geração e utilização da lista de palavras mais freqüentes*[10] *no resultado total [...] com uso de mineração de textos.*" (ARAÚJO JR., 2006)

2.2.1 Construção e armazenamento do índice

O índice tem como objetivo a recuperação rápida da informação. A forma como se constrói, armazena e manipula o índice muda de acordo com a tecnologia empregada e conseqüentemente sua evolução. Tradicionalmente, a unidade central de processamento (*CPU)* era lenta e a utilização de técnicas de compactação não seria interessante. Hoje as *CPUs* já são mais rápidas; entretanto temos um armazenamento em disco rígido lento, que para contornar necessitamos diminuir o espaço de armazenamento ou mesmo utilizar

[9] O estudo corresponde a uma tese de doutoramento defendida pelo autor no departamento de ciência da informação e documentação da Universidade de Brasília. A importância do tema na ciência da informação ficou destacada pela premiação de melhor tese de 2006 pela Ancib durante o Enancib 2006.

[10] A ferramenta desenvolvida para a realização do experimento proposto nesta pesquisa, OGMA, também possui essa funcionalidade.

memórias mais rápidas (subindo na hierarquia de memória) como a memória de acesso aleatório (*RAM)*.

Basicamente, a criação do índice significa criar um dicionário de palavras utilizadas em todos os documentos da coleção e criar um índice invertido indicando em que documento cada palavra aparece.

Com a criação desse índice torna-se extremamente mais rápida a busca de informações do que a varredura de todos os textos palavra por palavra.

2.2.2 Modelo booleano

Entre os modelos utilizados na recuperação da informação, a maior parte dos SRI tem como base o modelo clássico ou o modelo estruturado:

Nos modelos clássicos, cada documento é descrito por um conjunto de palavras-chave representativas, também chamadas de termos de indexação, que buscam representar o assunto do documento e resumir seu conteúdo de forma significativa. (BAEZA-YATES e RIBEIRO-NETO, 1999)

Nos modelos estruturados, podem-se especificar, além das palavras-chave, algumas informações acerca da estrutura do texto. Essas informações podem ser as seções a serem pesquisadas, fontes de letras, proximidade das palavras, entre outras características.

Dentre os modelos clássicos, temos o booleano, o vetorial e o probabilístico. O modelo booleano é baseado na teoria dos conjuntos e possui consultas especificadas com termos e expressões booleanas. Nas consultas são utilizados operadores lógicos como E, OU, NÃO para filtragem do resultado.

Apesar de ser um modelo bastante simples e muito utilizado ele apresenta as seguintes desvantagens:

- A recuperação é baseada numa decisão binária sem noção de combinação (*matching*) parcial;
- Nenhuma ordenação de documentos é fornecida;
- A passagem da necessidade de informação do usuário à expressão booleana é considerada complicada;
- As consultas booleanas formuladas pelos usuários são freqüentemente simplistas;
- Em conseqüência, o modelo booleano permite retorno de poucos ou muitos documentos em resposta às consultas;

- O uso de pesos binários é limitante; (BAEZA-YATES e RIBEIRO-NETO, 1999)

Para contornar estas limitações novos modelos são desenvolvidos tendo como base algum destes modelos clássicos.

2.2.3 Modelo vetorial

Um dos modelos que permite localizar similaridade entre documentos é o vetorial; esse modelo é também o mais comum. O vetor é definido através do conjunto de documentos que formam o corpus. (SALTON e MCGILL, 1983)

Todo o texto dos documentos é extraído e convertido em um formato que permita a fácil manipulação. A ordem das palavras é ignorada, o que pode ser interpretado como colocar todas as palavras de cada documento em um 'saco' separado (a expressão *bag of words*). Todas as palavras contidas em cada documento (saco) são contadas (processo de indexação) e o número de vezes que cada palavra aparece (forma mais simplista de dar valor ao peso da palavra) é armazenado em um vetor chamado termo-por-documento.

Ele é arranjado de forma que cada linha represente uma palavra (termo) e cada coluna represente um documento. Os valores contêm o peso dos termos para cada documento. Em geral esse tipo de vetor é extenso e a maioria dos pesos dos termos é zero.

Nas colunas estão representados os pesos de cada termo no documento. No exemplo do QUADRO 4, na próxima página, o termo 'rede' tem o peso de 0,75 no documento 5 enquanto que o termo 'pesquisa' não aparece no documento 3, sendo, portanto, zero o seu peso.

QUADRO 4 – Exemplo de representação vetorial: termos x documentos.

TERMO/DOCUMENTO	d1	d2	d3	d4	d5	d6	d7	d8	d9
Rede	0	0,60	0	0,20	0,75	0,02	0	0,15	0,80
Social	0,20	0	0,05	0,30	0,75	0	0,02	0	0
Pesquisa	0	0,40	0	0,50	0	0	0	0	0,20
Vetor	0,20	0	0	0	0	0	0,10	0,10	0

Fonte: exemplo do autor desta tese.

Sobre o uso de pesos no modelo vetorial, Baeza-Yates e Ribeiro-Neto (1999) apresentam algumas considerações:

- pesos não binários podem considerar mais adequadamente combinações parciais;

- estes pesos são utilizados para calcular um grau de similaridade entre a consulta e o documento;
- a fórmula com que são calculados os pesos varia dentre as implementações.

Cada documento (coluna) pode ser considerado um vetor ou uma coordenada em um espaço do vetor multidimensional em que cada dimensão represente um termo.

Alguns termos que podem prejudicar a recuperação, conhecidos como *stopwords*[11], são extraídos do texto através de um processo de tratamento do documento conforme ilustrado na FIGURA 2:

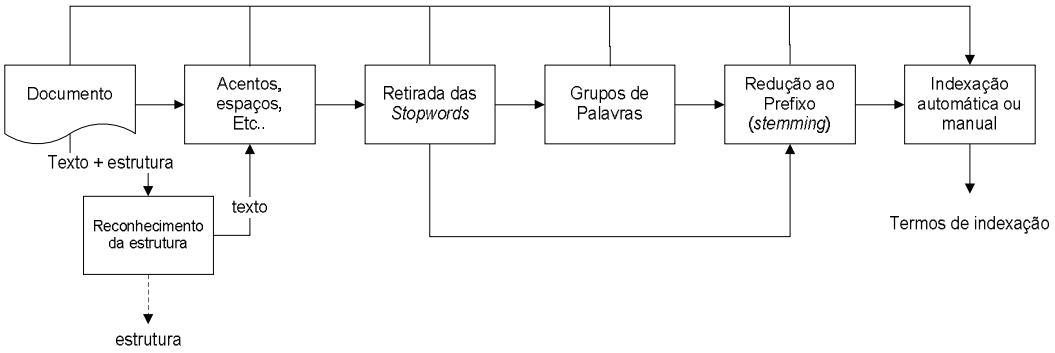


FIGURA 2 – Fases do processamento do documento para submissão à indexação.
Fonte: Adaptado de BAEZA-YATES & RIBEIRO-NETO, 1999, p. 166.

Ao final do processamento têm-se, através de um processo de indexação automática ou manual os termos de maior relevância para indexação. Técnicas como a de *stemming*[12] podem ser utilizadas para reduzir a redundância semântica entre os termos.

2.2.4 Índice por peso (TF-IDF)

A medida *term frequency–inverse document frequency* (TF-IDF) corresponde a uma medida estatística utilizada para avaliar o quanto uma palavra é importante para um documento em relação à uma coleção (*corpus*). Essa importância aumenta proporcionalmente com o número de vezes em que a palavra apareça no documento e diminui de acordo com a freqüência da palavra na coleção.

O *term frequency* (TF) corresponde ao número de vezes em que o termo apareça no documento. Uma normalização evita que documentos grandes sobressaiam aos documentos pequenos no conjunto. A equação é dada por:

[11] Palavras que não são úteis para a recuperação de informações (eexmplo palavras comuns, preposição, artigos, etc.)
[12] Processo de remover prefixos e sufixos das palavras do documento.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Na qual:

$n_{i,j}$ é o numero de ocorrências do termo t_i no documento d_j e o denominador corresponde ao número de ocorrências de todos os termos no documento d_j.

Já a IDF é uma medida de grande importância para complementar a equação acima já que avalia a importância do termo na coleção. É obtida dividindo-se a quantidade de documentos pelo número de documentos, contendo o termo e então obtendo o logaritmo do resultado:

$$idf_i = log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

Na qual:

|D | é o total de documentos no corpus

$|\{d_j : t_i \in d_j\}|$ número de documentos nos quais o termo t_i aparece.

Através da união das duas têm-se TF-IDF:

$$tfidf_{i,j} = tf_{i,j} . idf_i$$

Dependendo da aplicação e do experimento, a partir do modelo TF-IDF podem surgir outros modelos que modifiquem a sistemática de atribuição de pesos.

2.3 Processamento de linguagem natural

O processamento da linguagem natural (PLN) tem como objetivo realizar o tratamento através do computador de aspectos da comunicação humana, como sons, palavras, sentenças e discursos, levando em consideração formatos e referências, significados e estruturas, usos e contextos. De forma simplista, pode-se dizer que o PLN visa fazer o computador comunicar-se em linguagem humana, nem sempre necessariamente em todos os níveis de entendimento e/ou geração de sons, palavras, sentenças e discursos. Esses níveis são:

- **fonológico e fonético**: trata do relacionamento das palavras com os sons que produzem;
- **morfológico**: trata da construção das palavras a partir unidades de significado primitivas e de como classificá-las em categorias morfológicas;

- **sintático**: trata do relacionamento das palavras entre si, cada uma assumindo seu papel estrutural nas frases, e de como as frases podem ser partes de outras, constituindo sentenças;
- **semântico**: trata do relacionamento das palavras com seus significados e de como eles são combinados para formar os significados das sentenças; e
- **pragmático**: trata do uso de frases e sentenças em diferentes contextos, afetando o significado.

Um dos exemplos das inúmeras possibilidades do PLN é o concurso do prêmio Loebner, realizado na Universidade de Reading, no qual programas de computador (*chatbots*) tentam manter uma conversação (*chat*) com um jurado humano. O jurado fica diante de um terminal de conversação duplo e deve identificar que terminal é a máquina (programa) e que terminal é a pessoa.

O concurso utiliza as idéias do matemático britânico Alan Turing, que propôs uma regra subjetiva, mas simples, para determinar se máquinas podiam ser capazes de realizar o pensamento. Turing (1950) argumentou que a conversação seja prova de inteligência. Se um computador fala como uma pessoa, então para todos os fins práticos ele pensa como uma pessoa.

Outro exemplo de PLN é a análise semântica latente (*LSA*), ou também *latent semantic indexing* (*LSI*), relacionada à manipulação de vetores de índice. Ela está relacionada à aplicação da matemática para analisar a relação entre termos e documentos e decompor o vetor de índice. O processo matemático utilizado é o *simple value decomposition* (*SVD*).

A LSA trabalha com a sinonímia e polissemia. Por exemplo, para a consulta ‘extravio de bagagem’ aplicada a uma ferramenta de busca que usa LSA, o sistema recuperará documentos que contenham as frases ‘extravio de bagagem’ e ‘extravio de mala’ já que ‘bagagem’ e ‘mala’ têm o mesmo significado no contexto. Da mesma forma, em uma consulta por "banco de dados", o resultado incluirá somente documentos que contenham uma relação com ‘banco de dados’, excluindo documentos que se refiram a banco como objeto de descanso e a banco como entidade financeira.

2.3.1 Gramática gerativa

A lingüística é o estudo científico da linguagem verbal humana. A gramática gerativa é uma teoria lingüística elaborada por Noam Chomsky e pelos lingüistas do *Massachusetts Institute of Technology* (*MIT*) entre 1960 e 1965[13]. O estudo da gramática generativa revolucionou os estudas da lingüística.

Os modelos tradicionais descrevem somente as frases realizadas e, portanto, não relacionam um grande número de dados lingüísticos (como a ambigüidade, os constituintes descontínuos, etc.). Nessa perspectiva, esses modelos tradicionais correspondem a um mecanismo finito que permite analisar[14] um conjunto de frases (bem formadas, corretas) de uma língua, e somente elas.

Chomsky (1969) propõe então uma teoria capaz de dar conta da criatividade do falante, de sua capacidade de emitir e de compreender frases inéditas.

A gramática gerativa compreende na realidade um conjunto de modelos teóricos que tem em comum a sua intenção de estudar o dispositivo mental inato, responsável pela produção lingüística.

Na gramática gerativa aplicam-se três modelos mais conhecidos: (CHOMSKY, 1969; LOPES, 1999)

a) modelo dos estados finitos
b) modelo sintagmático – gramática sintagmática
c) modelo transformacional – gramática transformacional (GT)

O modelo dos estados finitos baseia-se na aplicação de regras recursivas sobre um vocabulário finito. Segundo Chomsky, esse tipo de descrição gramática concebe as frases como tendo sido engendradas por uma série de escolhas executadas pelo formulador da frase.

Na mesma linha, segundo Trask (2004), a gramática sintagmática é

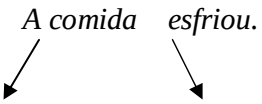
> "um tipo de gramática gerativa, que representa diretamente a estrutura dos constituintes. Normalmente, consideramos a estrutura de qualquer sentença (frase) como um caso de estrutura de constituintes, em que as unidades menores se combinam para formar unidades maiores, que são, por sua vez, combinadas formando unidades ainda maiores, e assim sucessivamente".

Assim, por exemplo, na gramática sintagmática, o sujeito da frase é identificado por propriedades formais. Nesse modelo toda a oração é formada por unidades de significado (os

[13] O principal trabalho inicial é a publicação em 1957 de uma parte da tese de Noam Chomsky , sob o título de *Estruturas Sintácticas.*
[14] Utiliza-se o termo "engendrar"

sintagmas) que se organizam de acordo com leis determinadas. As leis que organizam os sintagmas são chamadas sintagmáticas.

No exemplo:



(primeiro sintagma) (segundo sintagma)

Na frase do exemplo, aparecem dois sintagmas. Se no primeiro grupo, no lugar de ‘a comida’ escreve-se ‘comida a’, não se formaria um sintagma, pois a seqüência não existe nas leis sintagmáticas.

Observa-se também que, na frase, o primeiro sintagma se organiza em torno de um substantivo (‘comida’), portanto o primeiro grupo é um sintagma nominal (SN). Já o segundo sintagma tem como base um verbo. Portanto é um sintagma verbal (SV).

O sintagma verbal é caracterizado pela presença do verbo. Além do verbo, outros termos podem fazer parte do SV, dependendo do verbo que funcione como núcleo. Esses outros elementos são, por sua vez, sintagmas - nominais ou preposicionados.

Os sintagmas nominais e verbais são básicos numa oração. Além deles existem:

a) sintagma adjetival (SA)

b) sintagma adverbial (SAdv)

c) sintagma preposicional (SP)

Conforme Perini (1996), o sintagma nominal possui uma estrutura bastante complexa, pois é possível distinguir em sua composição várias funções sintáticas. Seu núcleo pode ser um nome (comum ou próprio) ou um pronome (pessoal, demonstrativo, indefinido, interrogativo ou possessivo). O sintagma nominal pode também ser constituído por determinantes e/ou modificadores, sendo que os modificadores antecedem ou sucedem o núcleo, enquanto os determinantes apenas o antecedem. (MIORELLI, 2001)

Alguns exemplos de composição de sintagmas nominais:

a) substantivo / nome próprio:

<u>Ester</u> é uma profissional excelente.

b) determinante + substantivo:

<u>Aquela professora</u> é excelente.

c) determinante + substantivo + adjetivo:

<u>Aquela professora bonita</u> é excelente.

d) determinante + substantivo + sintagma preposicional:

<u>Aquela professora de cultura</u> é excelente.

d) determinante + substantivo + oração adjetiva:

A professora que veio da China.

e) pronome substantivo:

Ela é uma excelente profissional.

f) qualquer palavra substantivada:

O responsável pela chave foi embora.

A gramática sintagmática é mais forte do que uma gramática de estados finitos, pois consegue realizar uma análise (engendrar) em frases nas quais os modelos de estados finitos não consigam realizar. (LOPES, 1999)

Devido a dificuldades encontradas na análise sintagmática, em explicar frases com constituintes descontínuos (separados por morfemas) em uma oração, Chomsky (1969) propõe o uso do modelo transformacional ou gramática transformacional (GT). Esse modelo corresponde a "*uma gramática gerativa que inclui também o conceito de transformação, ou seja, a aplicação de um conjunto de regras que convertem uma estrutura profunda de uma língua em estrutura superficial*". (HOUAISS, 2001)

As estruturas superficiais "correspondem mais de perto à forma física de realização concreta da oração e determinam sua interpretação fonológica, enquanto que a *estrutura profunda* corresponde" (SILVA e KOCH, 2007), à

> "representação da frase em nível abstrato, na qual se estabelecem as relações semânticas básicas entre os itens lexicais, cuja ordem linear pode ser modificada com a aplicação das transformações que forem necessárias para derivar a estrutura superficial, mantendo as relações semânticas iniciais na estrutura subjacente". (HOUAISS, 2001)

Alguns autores levam o conceito de sintagma nominal da lingüística para trabalhar as questões semântica e informacional presentes no sintagma nominal. Silva e Koch (2007) definem que o sintagma nominal consiste em um: "*conjunto de elementos que constituem uma unidade significativa dentro da oração e que mantêm entre si relações de dependência e de ordem. Organizam-se em torno de um elemento fundamental, denominado núcleo, que pode, por si só, constituir o sintagma*."

Também na definição de Kuramoto (1996), sintagma nominal "*é a menor parte do discurso portadora de informação*".

Como exemplo, o sintagma nominal:

"O estudo da economia da informação".

Possui três outros SN embutidos:

1. a economia da informação
2. a informação
3. a economia

Os SN “a informação” e “a economia” são sintagmas nominais aninhados dentro do SN “a economia da informação”.

Como se pode perceber, a descoberta dos SN evidencia a organização em um esquema de árvore e, assim, diferentemente das palavras, o SN quando extraído do texto mantém o seu significado, o seu conceito.

O sintagma nominal pode ser dividido de acordo com as funções semânticas dos componentes de um SN, definidas por Liberato (1997)[15], que são: classificador (CLA), sub-classificador (SUB), qualificador (QUAL), recortador (REC) e quantificador (QUAN). Essas funções determinam como os objetos são classificados no mundo (tipos de características de acordo com a classe gramatical) e compõem o QUADRO 5.

Por exemplo, no termo “livro de bolso” a palavra “livro” constitui o centro do SN, ou o núcleo do SN; e “de bolso” caracteriza a identificação (qualificador) de uma classe de livros.

QUADRO 5 – Categorias gramaticais das funções de Liberato.

FUNÇÕES SEMÂNTICAS	DESCRIÇÃO	PRINCIPAIS CATEGORIAS GRAMATICAIS
CLA	Classificador	Substantivos
SUB	Sub-classificador	Adjetivos, pronomes possessivos
QUAL	Qualificador	Adjetivos
RECUNI	Recortador universal	Artigos definidos
RECPAR	Recortador parcial	Artigos indefinidos
QUAN	Quantificador	Advérbios precedidos de artigo definido

Fonte: LIBERATO, 1997.

A utilização dos sintagmas nominais como estruturas de acesso à informação contida em uma base de dados textual se apresenta como uma alternativa aos sistemas tradicionais de

[15] Liberato em sua tese de doutorado, defendida na UFMG, tem como objetivo trabalhar a abordagem cognitiva do sintagma nominal em português.

recuperação de informação. Podendo aproximar um pouco mais a necessidade informacional do usuário da forma como os documentos potenciais (aqueles que talvez possam responder a essa demanda) possam representar essa necessidade.

Alguns algoritmos de recuperação da informação trabalham com a identificação e a indexação por lexemas para facilitar a recuperação de termos com sentido semelhante. Lexemas são palavras vinculadas através de uma relação denominada de flexão (PERINI, 1996). Por exemplo, as palavras cantora e cantar não compõem um lexema porque não pertencem à mesma classe morfológica. A primeira é substantivo e a segunda, verbo.

Já as palavras cantoria, cantilena e cantata não pertencem ao mesmo lexema porque embora apresentem um radical comum e pertençam à classe dos substantivos, diferem entre si por sufixos derivativos e não por sufixos flexivos.

As palavras cantor, cantora, cantores e cantoras compõem um lexema porque pertencem à mesma classe morfológica, a dos substantivos, e diferem entre si unicamente por sufixos flexivos (morfema zero, -a, -es, -as). Além disso, distribuem-se de forma complementar.

Em classes de palavra que não mudem como as das preposições e conjunções, têm-se lexemas formados de uma única palavra.

Entretanto para classificação e recuperação da informação somente o uso de lexemas não é interessante (PERINI, 1996). Para PERINI o uso de sintagma nominal é mais eficiente nesse processo.

2.3.2 Analisadores do português e a extração de sintagmas nominais

A extração automatizada de sintagmas nominais sempre foi um elemento de dificuldade em pesquisas de recuperação da informação envolvendo sintagmas nominais. Kuramoto (2002) ressalta essa dificuldade, principalmente em um grande volume de textos:

> "O processo de reconhecimento, extração e indexação não automatizada, além de ser inviável economicamente em se tratando de grandes volumes de documentos, pode prejudicar a uniformidade no processo de reconhecimento, extração e indexação dos sintagmas nominais." (KURAMOTO, 2002)

O pesquisador complementa no mesmo artigo que "*A inexistência dessas ferramentas impede uma avaliação mais consistente envolvendo amostras de dados com maior volume de documentos.*" (KURAMOTO, 2002)

Entretanto em 2002 já existiam algumas ferramentas que possibilitariam tal extração; essas ferramentas computacionais são conhecidas como analisadores. Sua função básica é identificar as classes gramaticais e os elementos sintáticos e semânticos que compõem uma sentença ou texto.

Essa ferramenta é o analisador PALAVRAS, desenvolvido por Bick (1996) em sua tese de doutorado na *Southern University of Denmark*. Hoje o programa faz parte de um conjunto de ferramentas multilíngües denominada *visual interactive syntax learning* (VISL), que disponibiliza uma interface na internet, na qual o usuário envia sentenças ou textos completos e recebe de volta os textos com a marcação.[16] O site da VISL permite a visualização de árvores sintáticas, marcação em cores entre outras opções.

O princípio de análise da ferramenta é a gramática de restrições (*constraint grammar – CG*) que faz a análise do texto morfologicamente (lexemas), de grupos de palavras e da composição da oração. Com isso o programa obtém uma análise nos níveis ortográfico, semântico e sintático.

Após a aplicação da identificação do léxico, o programa elimina as ambigüidades encontradas em cada palavra, através da aplicação de um conjunto de regras na sentença identificando e eliminando possibilidades de formas sintáticas inexistentes.

A FIGURA 3 demonstra o resultado da análise efetuada pelo VISL na sentença "*ele está na selva*".

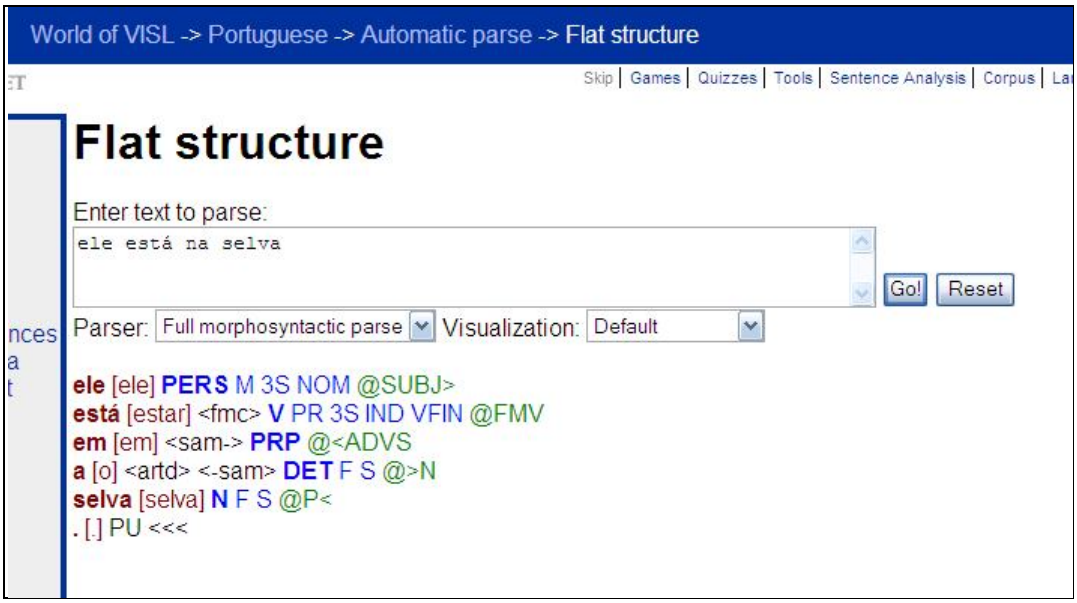


FIGURA 3 – Exemplo de utilização do VISL-Palavras.
Fonte: http://visl.sdu.dk.

[16] Um texto "marcado" é o mesmo que um texto "anotado", e significa que além do texto original existem marcações (ou anotações) indicando elementos lingüísticos diversos.

O princípio da gramática de restrições utilizado no VISL é também utilizado em outra ferramenta mais simples, o *Grammarplay* (OTHERO, 2004). O *Grammarplay* funciona apenas com sentenças e tem o objetivo de demonstrar a aplicação da gramática de restrições.

O programa realiza a análise da sentença com base em um léxico no qual identifica as possibilidades de cada palavra. Posteriormente identifica os sintagmas, aplicando as regras sintagmáticas de sua gramática. Com isso o programa consegue identificar os sintagmas nominais (SN), sintagmas adjetivais (SAdj), sintagmas preposicionais (SP), os sintagmas verbais (SV) e sintagmas adverbiais (SAdv). Em sua gramática também possui as regras lexicais que auxiliam o programa nas identificações acima.

A FIGURA 4, a seguir, demonstra o resultado da análise efetuada pelo *Grammarplay* na mesma sentença "*ele está na selva*".

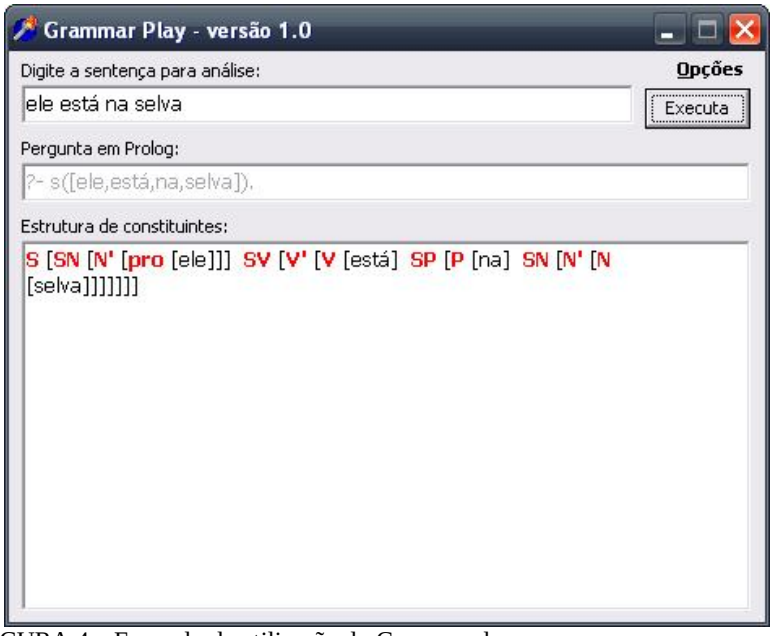


FIGURA 4 – Exemplo de utilização do Grammarplay.
Fonte: Tela do programa.

O programa também apresenta a possibilidade de visualização do resultado da análise em uma árvore. A FIGURA 5 demonstra esta funcionalidade:

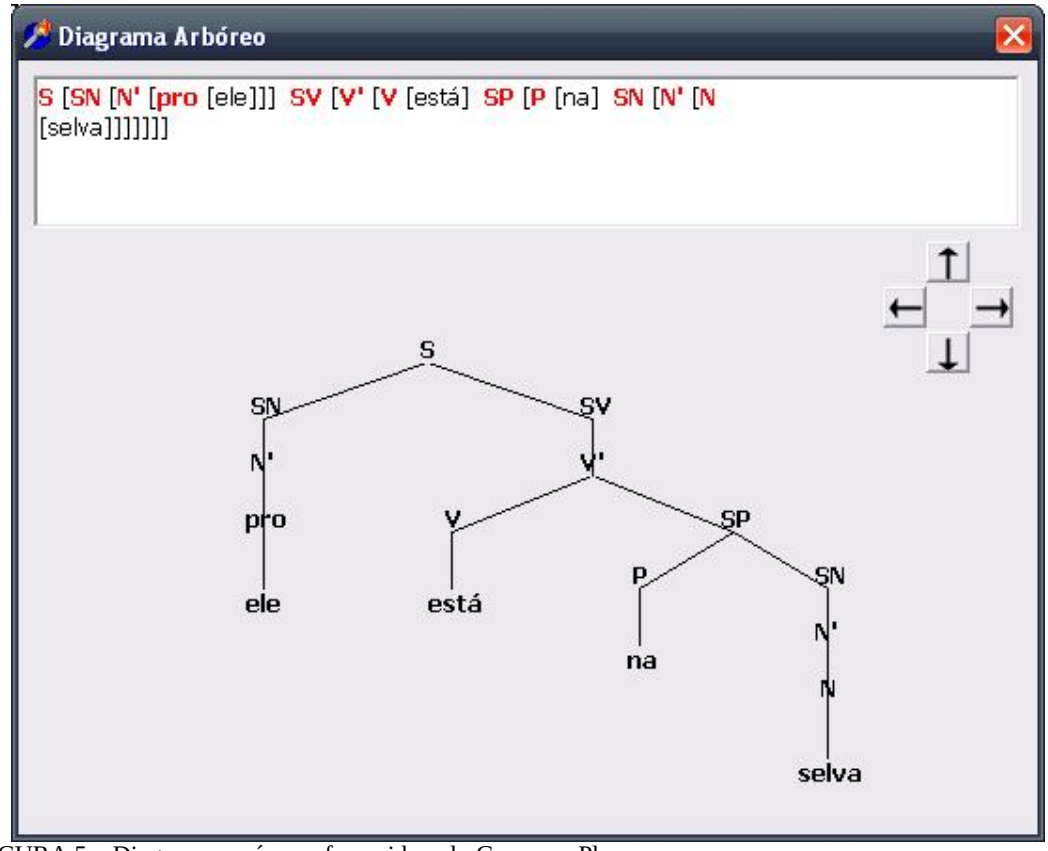


FIGURA 5 – Diagrama em árvore fornecido pelo Grammar Play.
Fonte: Tela do programa.

Outro analisador é o *lx-tagger* que faz parte de um conjunto de ferramentas de nome lx suite. O *lx-conjugator* realiza a conjugação de verbos, o *lx-infletor* fornece o termo em suas variantes de gênero (masculino/feminino) e número (singular/plural), o *lx-lemmatizer* (lematizador) identifica a conjugação do verbo fornecido e o lx-tagger realiza a marcação das categorias morfo-sintáticas do texto.

A FIGURA 6 demonstra o resultado da análise efetuada pelo *lx-tagger* na sentença "*ele está na selva*".

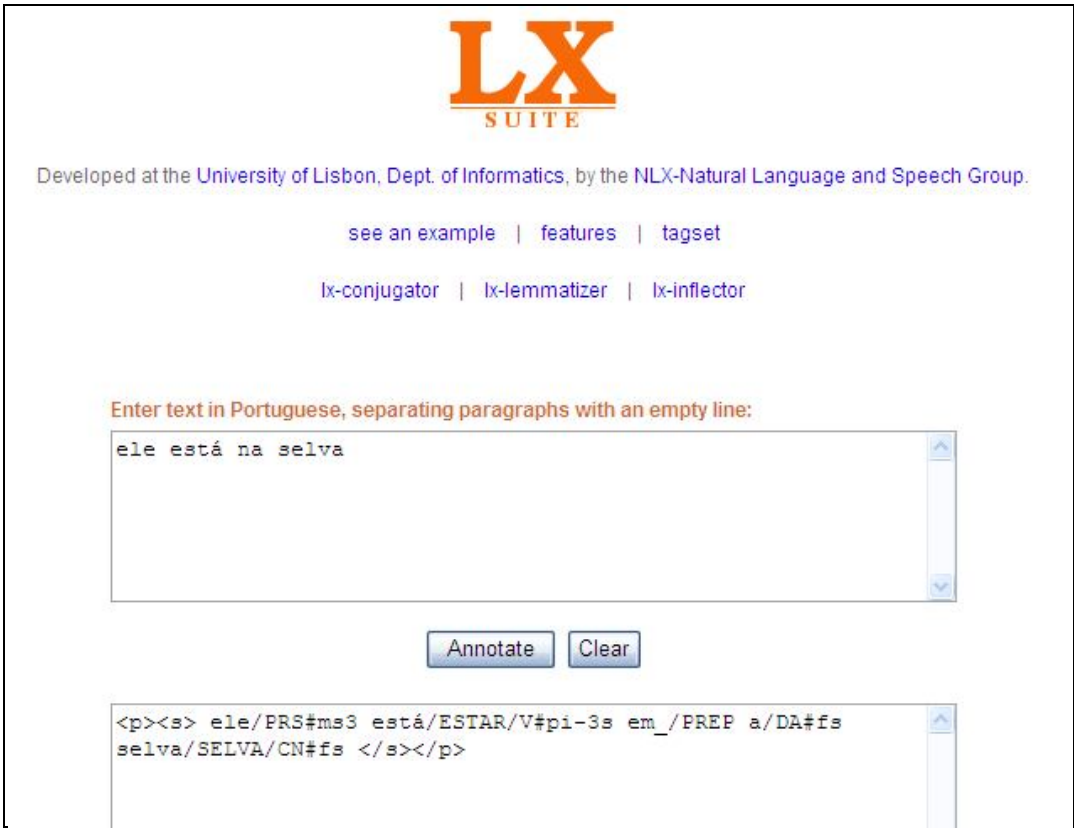


FIGURA 6 – Exemplo de utilização do lx-tagger.
Fonte: http://lxsuite.di.fc.ul.pt.

O programa lx-suite ganhou destaque na reportagem "No tempo em que as máquinas falavam", INFO (2006), na qual se discutem as diversas aplicações de ferramentas de análise do português.

Na reportagem é apresentado o projeto Gramaxing no qual os pesquisadores do grupo NLX têm como objetivo o desenvolvimento de uma gramática computacional do português que possa ser aplicado em um projeto maior conhecido como Delphin que abarca vários projetos de outras línguas.

O responsável pelo grupo NLX, Antônio Branco, também faz o seguinte relato à revista:

> "Defendemos que os computadores podem criar representações e significados da linguagem natural e manipulá-las para o apoio de sistemas de metadados, web semântica ou traduções. Se manipular um texto consiste em passá-lo de uma linguagem para outra, então, podemos concluir que o computador consegue interpretar. Mas se compreendermos que a interpretação corresponde à compreensão de um significado... aí prefiro não ter opinião. É uma questão para antropologia e que consiste em saber se a natureza dos humanos é replicável em máquina". (INFO, 2006)

Outro analisador é o projeto Curupira (FIG. 7) desenvolvido pelo núcleo interinstitucional de lingüística computacional desde 1997. O Curupira integra um revisor

gramatical automático (regra - o revisor gramatical automático para a língua portuguesa) utilizado em editores de texto como o Redator da Itautec. O Curupira toma como premissa, a hipótese de que as sentenças da língua portuguesa possam ser representadas por estruturas em árvores, e que essa representação seja não apenas útil, mas imprescindível para a determinação das relações de dependência que se observam entre os itens lexicais co-ocorrentes. (MARTINS, HASEGAWA E NUNES, 2002)

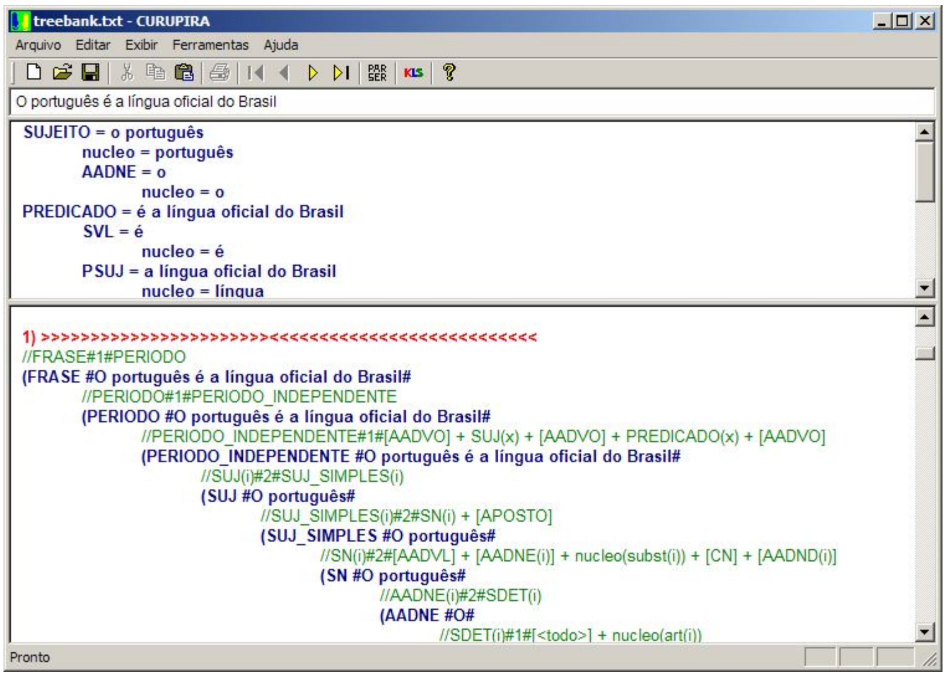


FIGURA 7 – Analisador Curupira.
Fonte: MARTINS, HASEGAWA E NUNES, 2002.

No QUADRO 6 estão relacionados os principais analisadores existentes na língua portuguesa. Na terceira coluna estão descritas as dificuldades encontradas em cada uma das ferramentas para serem aplicados em pesquisas como esta.

QUADRO 6 – Analisadores língua portuguesa.

FERRAMENTA	REFERÊNCIA	DIFICULDADES
VISL - Automatic Analysis of Portuguese	Bick, 1996[17]	(-) O envio do texto é feito pela web. (-) anotação manual: arquivo por arquivo. (-) Para anotações em um corpus maior é necessário adquirir licença. (-) Português europeu
Curupira - Parser para o português brasileiro	Martins, Hasegawa e Nunes, 2002[18]	(-) não está disponível ao público.
Grammar Play	Othero, 2004.	(-) léxico limitado. (-) apenas pequenas frases.
PoSiTagger	Aires, 2000.[19]	(-) dependente da ferramenta MXPost.
LX-Tagger / LX Suite	Natural Language and Speak group (NLX)[20]	(-) O envio do texto é feito pela web. (-) anotação manual: arquivo por arquivo. (-) Português europeu

Fonte: MARTINS, HASEGAWA E NUNES, 2002.

2.3.3 O método ED-CER

O trabalho de Miorelli (2001) tem como objetivo "*propor um método para a extração automática do Sintagma Nominal de sentenças em português, aplicando recursos de Processamento de Linguagem Natural (PLN) em Sistemas de Recuperação de Informações*"

Entretanto, entre as diversas etapas do método ED-CER, detalhadas adiante, a etapa de etiquetagem, necessária para a extração dos SN, é realizada manualmente.

O processo de etiquetagem consiste em atribuir etiquetas, informando a classe gramatical a cada palavra, como no exemplo:

"o_**AR** trabalho_**SU** descreve_**VB** as_**AR** Gramáticas_**SU** Síncronas_**AJ** de_**PR** Adjunção_**SU** de_**PR** Árvores_**SU** como_**PR** formalismo_**SU** para_**PR** projeto_**SU** de_**PR** um_**AR** módulo_**SU** (...) ._**PN**"

[17] Mais informações: http://visl.hum.sdu.dk/visl/pt
[18] Mais informações: http://www.nilc.icmc.usp.br/nilc/tools/curupira.html
[19] Mais informações: http://www.nilc.icmc.usp.br/nilc/projects/mestradorachel.html
[20] Mais informações: http://nlx.di.fc.ul.pt/

A tabela completa de etiquetas do ED-CER encontra-se no QUADRO 7, a seguir:

QUADRO 7 – Etiquetas (classes gramaticais) do método ED-CER.

ETIQUETA	GRUPO
_AR	artigos definidos e indefinidos
_NU	numerais ordinais e cardinais
_AJ	adjetivos e verbos no particípio
_CO	vírgula e conexões: conjunções aditivas, adversativas e alternativas
_PP	pronomes pessoais
_PR	Preposições
_LG	ligações: pronomes relativos e locuções (prepositivas, comparativas etc.)
_PD	pronomes demonstrativos
_PI	pronomes indefinidos
_PS	pronomes possessivos
_AV	Advérbios
_SU	Substantivos
_NP	nomes próprios
_VB	Verbos
_PN	pontuações (exceto a vírgula)

Fonte: MIORELLI, 2001.

Depois de ser etiquetado, o texto a ser analisado é submetido a dois módulos o seletor e o analisador. O método ED-CER e os seus dois principais módulos estão resumidos na FIG. 8, abaixo:

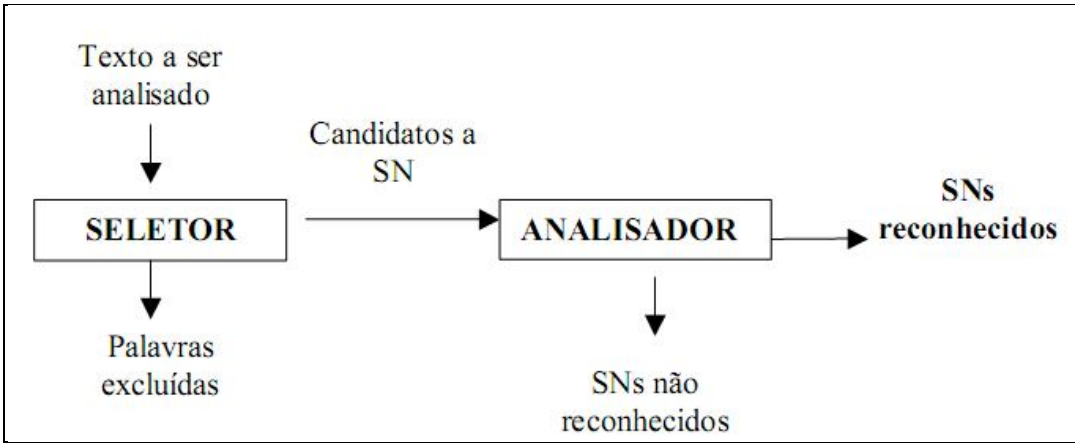


FIGURA 8 – Representação do método ED-CER.
Fonte: MIORELLI, 2001.

O módulo seletor tem como objetivo extrair da sentença analisada um candidato a SN, e seguindo à formação do SN máximo proposto por Perini (1996), o módulo seletor realiza, para cada trecho analisado, duas etapas: a etapa de corte e a etapa de exclusão de palavras. O

processamento inicia-se lendo a palavra mais à esquerda, e segue lendo palavra por palavra. (MIORELLI, 2001)

A etapa de corte é feita em dois momentos:

- ao ser encontrada uma etiqueta do tipo _VB, _LG, _PN, ou
- quando a etiqueta da primeira palavra lida for do tipo: _PR ou _CO.

A etapa de exclusão é feita nos seguintes momentos:

- quando se elimina a palavra de corte (_VB, _LG ou _PN) ou
- quando se eliminam palavras anteriores ao corte que possuam as etiquetas: _PR, _AV, _CO, _AR, _NU ou
- quando se elimina a primeira palavra lida, se ela possuir a etiqueta do tipo: _PR ou _CO.

O módulo analisador é o responsável pelo reconhecimento ou não do candidato a SN como um SN. Para tanto, utiliza a gramática representada no QUADRO 10. O programa inicia a leitura das palavras que constitua o candidato a SN, verificando sua conformidade com a gramática. Um analisador sintático reconhece os SNs de acordo com as regras estipuladas pela gramática. Os candidatos a SN que não estão de acordo com a gramática, não são extraídos como SN pelo módulo analisador.

Na construção das regras que compõem a gramática, Miorelli (2001) utilizou símbolos (*tokens*) terminais e símbolos não terminais. Os símbolos terminais são os que se realizam uma conversão direta a partir das etiquetas do texto. Os símbolos estão relacionados no QUADRO 8:

QUADRO 8 – Símbolos terminais do método ED-CER.

SÍMBOLOS TERMINAIS	
TOKENS	ETIQUETAS
det	_AR _PD _PI
qua	_AJ _NU _PS
adv	_AV
con	_CO
pre	_PR
ref	_SU _PP _NP

Fonte: MIORELLI, 2001.

Já os símbolos não terminais são símbolos intermediários utilizados na gramática para se obter o SN. Os símbolos não terminais estão relacionados no QUADRO 9:

QUADRO 9 – Símbolos não terminais.

SÍMBOLOS NÃO TERMINAIS	
PN	Pré-núcleo
MD	Modificador
AV	Advérbio
NS	Núcleo do SN
SN	Sintagma nominal

Fonte: MIORELLI, 2001.

Exemplos passo a passo de cada uma das duas etapas são apresentados no trabalho de Miorelli. (2001, p. 51)

Finalmente as regras da gramática, utilizadas pelo módulo analisador, para extrair os sintagmas nominais (símbolo SN) estão contidas no QUADRO 10, a seguir:

QUADRO 10 – Gramática do método ED-CER.

NS	→	ref			
MD	→	qua			
AV	→	AV	adv		
AV	→	adv			
MD	→	AV	MD		
MD	→	MD	con	MD	
NS	→	NS	MD		
NS	→	MD	NS		
NS	→	NS	pre	NS	
NS	→	NS	pre	det	NS
NS	→	NS	con	NS	
NS	→	NS	con	det	NS
NS	→	AV	NS		
SN	→	det	SN		
SN	→	NS			

Fonte: MIORELLI, 2001.

A autora sugere em suas considerações finais a utilização de um etiquetador que gere as etiquetas definidas em seu trabalho. E uma das funcionalidades da ferramenta Ogma

(detalhado no capitulo posterior, 3) é a aplicação de etiquetas para execução das regras adaptadas deste trabalho.

2.3.4 Escolha de descritores utilizando sintagmas nominais

Desde que a pesquisa de Kuramoto (1996 e 1999) verificou a viabilidade do uso de sintagmas nominais em sistemas de recuperação de informação, muitos estudos se deram a partir desse, principalmente estudos sobre quais seriam os sintagmas que poderiam descrever e ser relevantes para o conteúdo do documento. A pesquisa efetuada por Souza (2005) trabalha sobre esse problema de escolha dos descritores, e propõe os seguintes passos:

- extração dos sintagmas nominais do texto.
- análise de cada um dos sintagmas nominais extraídos e cálculo da pontuação do mesmo como descritor.

Souza (2005) propõe que os seguintes itens podem ser avaliados para realização dessa avaliação e conseqüente pontuação:

a) freqüência de ocorrência do SNs no texto do documento.

b) a incidência dos SN no conjunto de documentos.

c) seus níveis.

d) suas estruturas sintáticas.

e) sua ocorrência no tesauro da área.

Souza (2005) trabalhou com um corpus de 60 documentos: 30 artigos do periódico **DataGramaZero** e mais 30 artigos científicos da área de ciência da informação. Sobre esse corpus executou os seguintes passos:

1. extraem-se os SN, utilizando o *parser* Palavras e ordena-os de acordo com a freqüência de ocorrência em cada documento.
2. descartam-se os SN com ocorrência inferior. Os SN com ocorrência inferior a 1% sobre o total de SN únicos do documento.
3. classificam-se manualmente os SN de cada documento de acordo com sua relevância, detalhada no QUADRO 11, abaixo:

QUADRO 11 – Pontuação da avaliação dos SNs como descritores.

SÍMBOLO	RELEVÂNCIA DESCRITIVA DO SN	VALOR
SN***	SN extremamente relevante como descritor (SNER)	A
SN**	SN razoavelmente relevante como descritor (SNRR)	B
SN*	SN moderadamente relevante como descritor (SNMR)	C
SN-	SN não relevante como descritor	D

Fonte: SOUZA, 2005.

4. classificam-se manualmente os SN de acordo com a estrutura de cada um, como ilustrado no QUADRO 12, a seguir:

QUADRO 12 – Classificação dos sintagmas nominais (CSN).

SÍMBOLO CSN	DESCRIÇÃO	EXEMPLO
1a	Um classificador	organização
1b	Um classificador mais um sub-classificador ou qualificador	insumos básicos
2	Dois classificadores	camada de ozônio
3	Três classificadores	âmbito da representação das atividades econômicas
4	Quatro ou mais classificadores	O livro de visitas do museu de artes

Fonte: Adaptado de SOUZA, 2005.

5. calcula-se a pontuação de todos os SN, multiplicando-se a freqüência do SN no documento e no corpus pelo peso atribuído ao CSN na qual, o SN foi classificado (etapa anterior).
6. realiza-se então uma comparação entre a extração manual (passo 3) com os SN de maior pontuação. Esse processo tem como saída uma taxa de relevância geral.

Souza (2005) encontrou os seguintes valores como peso (QUADRO 13), que proporcionaram maior taxa de relevância:

QUADRO 13 – Melhores valores encontrados como peso para cada CSN.

CSN	MELHOR VALOR
1a	0,2
1b	0,8
2	1,1
3	1,4
4	1,2
5	0,8

Fonte: SOUZA, 2005.

Toda a metodologia apresentada neste item foi implementada na ferramenta OGMA e foi utilizada nos experimentos propostos por esta pesquisa. O OGMA realiza identificação da classe do sintagma nominal, bem como o cálculo da pontuação do mesmo como descritor de

forma automática. A tabela de peso, apresentada no QUADRO 13, é a mesma utilizada pelo software OGMA.

Essa implementação é discutida em detalhes no capítulo 3.1.1, que descreve a ferramenta OGMA.

2.4 Similaridade de documentos eletrônicos

A classificação está presente em todas as nossas atividades do cotidiano. Isto é, de certa forma, comprovado na neurociência na qual já se estabelece o processo de associação ou de associar como se dá o processo básico de funcionamento do cérebro humano.

> "(...) só ela [classificação] nos permite orientar-nos no mundo a nossa volta, estabelecer hábitos, semelhanças e diferenças, reconhecer os lugares, os espaços, os seres, os acontecimentos; ordená-los, agrupá-los, aproximá-los uns dos outros, mantê-los em conjunto ou afastá-los irremediavelmente." (POMBO, 2003, p.01)

O processo básico de classificação pode ser resumido em associar dois itens. O processo final de classificação corresponde a uma atividade anterior de associação. Lakoff (1987) afirma que "sem a capacidade de categorizar, nós não poderíamos atuar nem no mundo físico nem no nosso mundo social e intelectual".

Processos classificatórios existem desde a antiguidade, quando organizar o conhecimento humano era preocupação dos filósofos; entretanto a palavra classificar, que vem do latim *classis*, teve sua origem em 1733, combinando *classis* e *facere*, e somente no final do século XVII foi empregada para se referir à ordem da ciência e do conhecimento.

> "[...] num sentido geral, é reunir em classes ou grupos, que apresentam entre si certos traços de semelhança, ou mesmo de diferença. Podemos ainda dizer que a classificação é um processo mental por meio do qual podemos distinguir coisas, pelas suas semelhanças ou diferenças, estabelecer as suas relações e agrupá-las em classes de acordo com essas relações." (SOUZA, 1950, p.3)

A ciência há tempos é utilizada como base para elaboração da classificação do conhecimento. Por tratar na maioria das vezes de forma empírica com seu processo de desenvolvimento, a ciência e a comunidade dependem da classificação como inclusive a própria comunicação científica. Conforme Kwasnik:

> "O processo da descoberta e a criação do conhecimento na ciência seguiu tradicional ao trajeto da exploração, observação, descrição, análise, e síntese sistemática e testar dos fenômenos e dos fatos, conduzidas todas dentro da estrutura de uma comunicação de uma comunidade de pesquisa particular com suas metodologias e conjunto aceitado das técnicas." (KWASNIK, 1999, pág. 22)

Atualmente vivemos em uma época em que os modelos de classificação tradicionais são criticados por incentivarem uma ciência atomizada para a qual os conhecimentos gerados não convergem.

A ciência da informação adota em sua classificação bibliográfica (a classificação utilizada para organização de uma coleção de livros) muitos conceitos herdados da classificação do conhecimento. A classificação bibliográfica como disciplina acadêmica tem apenas 125 anos e seu ensino e pesquisa têm crescido lentamente. (SATIJA, 2000, p. 221)

2.4.1 Classificação de documentos

A classificação aplicada à prática da biblioteconomia corresponde a fornecer aos livros e documentos, de um modo geral, o lugar certo em um sistema de recuperação de informações, na qual existe uma coleção que abranja os vários campos do saber, sendo cada item agrupado ou representado conforme sua semelhança, diferenças e relações recíprocas com outros itens dentro da coleção. A classificação pode corresponder ainda a determinar o assunto de um documento. Ou também, traduzir os assuntos dos documentos da linguagem natural para a linguagem artificial, de indexação, de forma a ser utilizada num sistema que permita recuperar eficientemente informações.

Aprimoramentos sobre a classificação automática, ou seja, realizada sem a intervenção do homem, objetivo desta pesquisa, tornam-se cada vez mais importantes num mundo com crescimento exponencial do volume de informação.

Na presente pesquisa utilizaram-se algoritmos e medidas de similaridades para realizar uma classificação automatizada de documentos eletrônicos. Por ser a classificação uma atividade inerente ao ser humano, técnicas e algoritmos computacionais tentam aprimorar-se, obtendo resultados próximos de uma classificação feita pelo homem; essa busca por algumas técnicas inclui até o uso de inteligência artificial.

É importante definir o tamanho da estrutura do sistema de classificação (ou seja, o número de classes) de acordo com o tamanho da coleção (SVENONIOUS, 1985, p. 11). As classes principais da estrutura são de extrema importância para a boa organização e o uso do sistema classificatório.

A classificação automática toma como base as propriedades do objeto que se pretende classificar e através delas define a(s) classes(s) à(as) qual(is) pertence o objeto. Ao classificar que um documento seja similar a outro, é necessário realizar um processo de associação entre esses documentos. Um documento com metadados (incluindo descritores) torna o processo de classificação automática mais eficaz. (SVENONIOUS, 1985, pág. 13)

2.4.2 Conglomerados

Conclomerados (ou *clustering)* correspondem às técnicas que permitam subdividir um conjunto de objetos em grupos. O objetivo é fazer com que cada grupo (ou *cluster*) seja o mais homogêneo possível, levando em consideração que os objetos do grupo tenham propriedades similares e que os objetos nos outros grupos sejam diferentes. (JANSSENS, 2007)

A FIG. 9 corresponde a uma ilustração do processo simplificado. Nela um conjunto desordenado de documentos passa pelo processo de agrupamento, sendo então os documentos divididos e organizados em três assuntos.

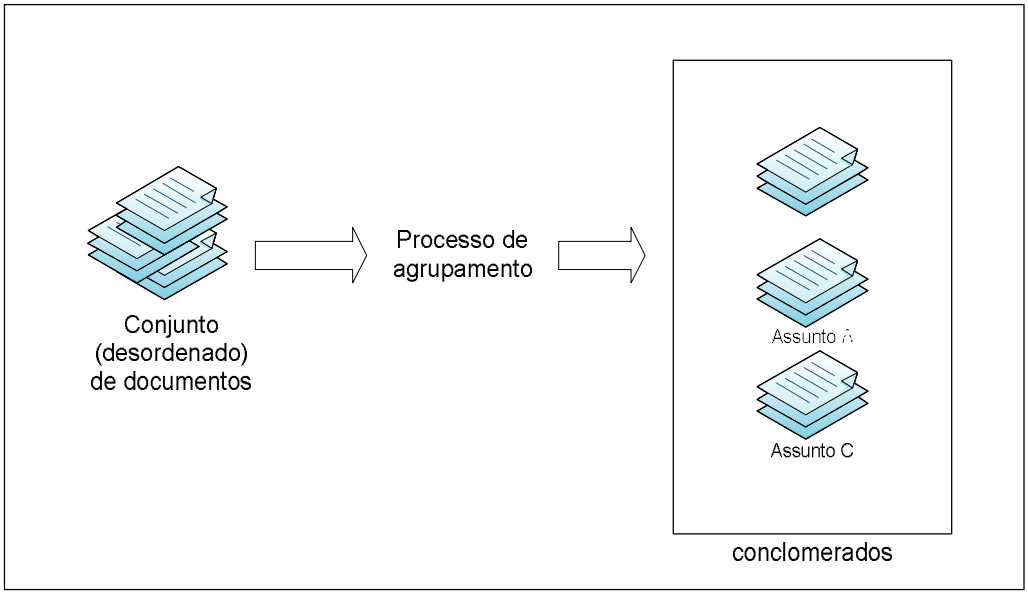


FIGURA 9 – Processo de agrupamento de documentos por assunto.
Fonte: Elaborada pelo autor desta tese.

O algoritmo de agrupamento pode funcionar, basicamente, através de duas formas:

- número de *clusters* automático – o número de categorias é definido automaticamente, geralmente com base no número de documentos da coleção.
- O número de *clusters* é pré-definido e as categorias apresentadas – As categorias já se encontram definidas antes da execução do algoritmo. Essa definição pode ser dada a partir de um conjunto de treinamento[21] (*training set*).

Um dos primeiros portais de pesquisa brasileiros, o Cadê, apresenta documentos organizados em diretórios, e a classificação de novos documentos bem como a criação de

[21] Itens selecionados de forma manual que servirão como base para o algoritmo classificar outros itens de forma automática.

categorias é feita manualmente ou pelo próprio usuário na hora da inserção do documento (site) no diretório. Essa característica faz com que a definição de um diretório apenas pelo seu nome seja suficiente, o que não ocorre no caso de uma classificação automática. O computador necessita de mais informações, além do nome, para basear a classificação.

O *computer science research paper search engine* (CORA)[22], (McCALLUM, 2000), é o projeto de um portal para pesquisa de artigos na área da ciência da computação que tem a organização dos seus diretórios (classes) feita de forma manual, porém a classificação de documentos (itens) feita de forma automática.

Para que ocorra a classificação automática dos itens nas classes, cada diretório do CORA recebe uma definição. Essa definição é composta por palavras-chaves atribuídas manualmente às categorias e a um conjunto de treinamento. O projeto ainda faz uso do algoritmo de treinamento *bootstrapping*, que refina a definição das categorias.

Uma dificuldade enfrentada na indexação tradicional de páginas da *web* diz respeito à grande quantidade de páginas a ser indexada, que exigiria um esforço humano grande, certamente impossível. Como proposta de solução para esse problema, têm-se os diretórios abertos e a indexação automática. Os diretórios são mecanismos de busca que utilizam uma estratégia para organizar as informações da *web* em que pesquisadores em ciência da informação estabelecem categorias, e indexadores atribuem manualmente páginas da *web* a essas categorias. Por exigirem um processo manual de indexação, os diretórios abrangem um universo mais restrito da *web*, quando comparados com os motores de busca baseados em palavras.

Por isso, com o objetivo de atingir maior quantidade de páginas indexadas, surgiram os diretórios abertos. *Open directory*[23] é um projeto de diretório aberto que, em vez de possuir um grupo seleto de indexadores, é formado por uma comunidade de milhares de indexadores, denominados editores. Qualquer pessoa pode ser um editor desse diretório. A desvantagem dessa alternativa é que esses editores não são especialistas em indexação.

2.4.3 Medidas de similaridade em documentos eletrônicos

Os algoritmos que resultam no valor de similaridade entre documentos trabalham com métricas que revelam o quanto um documento é similar a outro. Existem diversos algoritmos e métricas utilizados em fins diversos. Um algoritmo desse tipo pode, por exemplo, ser utilizado na grade de programação digital da televisão para fornecer programas similares ao

[22] O site do projeto não está atualizado e aparenta ter sido descontinuado.
[23] Para mais informações: http://www.dmosz.org

gosto do usuário, conforme demonstrado por Silva (2005) em um projeto denominado sistema de recomendação personalizada de programas de TV (SRPTV).

No campo da estatística há duas medidas básicas de similaridade que se expande para outros estudos: correlação e coseno.

A correlação de Pearson (ou apenas correlação) é a medida padronizada da relação entre duas variáveis. A correlação nunca pode ser maior do que 1 ou menor do que menos 1. Uma correlação próxima a zero indica que as duas variáveis não estejam relacionadas. Uma correlação positiva indica que as duas variáveis se movam juntas, e a relação é forte quanto mais a correlação se aproxime de um. Uma correlação negativa indica que as duas variáveis se movam em direções opostas, e que a relação também fique mais forte quanto mais próxima de menos 1 a correlação estiver. Duas variáveis, que estejam perfeitamente correlacionadas positivamente (r=1) movem-se essencialmente em perfeita proporção na mesma direção, enquanto dois conjuntos que estejam perfeitamente correlacionados negativamente movem-se em perfeita proporção em direções opostas. Se for 0 não existe correlação. E se for -1 existe uma correlação inversamente proporcional.

O coseno é similar à correlação, resultando em valores entre 0 e 1. Ele mede o ângulo entre dois vetores num espaço vetorial. Quanto mais próximo de 1 for o valor, mais similares serão os dois vetores.

Para se localizar a similaridade entre dois documentos em um SRI utilizando *vector space model (VSM)*, calcula-se o coseno do ângulo formado no vetor termo-por-documento. No *VSM* padrão quanto menor o ângulo, mais próximo de 1 será o coseno, e mais similar será o documento em relação àquele termo.

$$sim(\overrightarrow{d_1}, \overrightarrow{d_2}) = \cos(\widehat{\overrightarrow{d_1}\overrightarrow{d_2}}) = \frac{\overrightarrow{d_1} \cdot \overrightarrow{d_2}}{|\overrightarrow{d_1}| \cdot |\overrightarrow{d_2}|} = \frac{\sum_i w_{i,1} \cdot w_{i,2}}{\sqrt{\sum_i w_{i,1}^2} \cdot \sqrt{\sum_i w_{i,2}^2}}$$

Na qual: $w_{i,j}$ é o peso do temo t_i no documento d_j

Baeza-Yates e Ribeiro-Neto (1999) nos apresentam outras observações sobre este modelo como um todo:

- um conjunto ordenado de documentos é recuperado, fornecendo uma melhor resposta à consulta.

- documentos que têm mais termos em comum com a consulta tendem a ter maior similaridade;
- aqueles termos com maiores pesos contribuem mais para a combinação do que os que têm menores pesos;
- documentos maiores são favorecidos;
- a similaridade calculada não tem um limite superior definido.

O uso de um SRI e de um algoritmo de *clustering* para agrupar documentos envolve calcular a distância entre esses documentos na matriz. Existem além do co-seno de similaridade outras medidas, sendo que a distancia euclidiana é também muito utilizada. A distância euclidiana entre dois documentos d1 e d2 é definida por:

$$d(\overrightarrow{d_1}, \overrightarrow{d_2}) = \sqrt{\sum_i (w_{i,1} - w_{i,2})^2}$$

Na qual: $w_{i,j}$ é o peso do temo t_i no documento d_j.

A distância euclidiana necessita que quatro condições, nos vetores x, y e z, sejam válidas para atuar como medida:

1. $d(x, y) \geq 0$
2. $d(x, x) = 0$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Mais uma vez, o tamanho do documento tem grande influência quando se utiliza a distância euclidiana.

2.4.3.1 Naive Bayes

É o método de classificação baseado em inferência bayesiana. Trabalha com dados contínuos e discretos. Para dados discretos os valores de probabilidades são coletados através da contagem nos grupos dos documentos. Para dados contínuos, ele assume que os valores sigam uma função de distribuição normal, assim, as probabilidades são inferidas a partir da média e do desvio padrão de grupos dos documentos.

3.4.3.2 *K-means*

O objetivo do algoritmo *K-means* (ou K-médias) é fornecer um agrupamento de objetos de acordo com os seus próprios dados. Essa classificação é baseada em análise e

comparações entre os valores numéricos dos dados fornecidos. Dessa maneira, o algoritmo realiza um agrupamento automático sem a necessidade de nenhuma supervisão humana, ou seja, sem nenhum pré-agrupamento existente. Por causa desta característica, o *K-means* é considerado um algoritmo de mineração de dados não supervisionado.

O algoritmo analisa todas as instâncias fornecidas e as agrupa, isto é, o algoritmo vai indicar uma classe (*cluster*) e vai dizer que linhas pertencem a essa classe. O usuário fornece ao algoritmo a quantidade de classes que ele deseja (k). Este número de classes que deve ser passada para o algoritmo é chamado de *k* de onde vem a primeira letra do algoritmo: *K-means*.

Para agrupar os dados, o algoritmo faz uma comparação entre cada valor de cada instância por meio da distância. Geralmente utiliza-se a distância euclidiana para calcular o quão 'longe' uma ocorrência esteja da outra. A maneira de calcular essa distância vai depender da quantidade de atributos da tabela fornecida. Após o cálculo das distâncias o algoritmo calcula centróides para cada uma das classes. Conforme o algoritmo vá iterando, o valor de cada centróide é refinado pela média dos valores de cada atributo de cada ocorrência que pertença a esse centróide. Portanto, o algoritmo gera k centróides e agrupa as instâncias de acordo com sua distância dos centróides.

Em resumo, o algoritmo *K-means* consiste nos seguintes passos:

- escolher *k* pontos como centróides;
- associar cada instância ao centróide mais próximo;
- atualizar o centróide;
- repetir os passos até que nenhum ponto mude de centróide ou um número máximo de iterações seja executado.

2.4.3.3 Outras métricas

Calado et al. (2006) realizam um experimento utilizando as medidas de similaridade: *Amsler, bibliographic coupling, co-citacion, kNN,* SVM e *Naive Bayes* utilizando um *corpus* baseado no diretório de busca CADE. A pesquisa conclui que sejam necessárias novas experiências em outros *corpora* de documentos.

Existem outras métricas propostas em estudos para a identificação de dados similares como edge cover (SKIENA, 1997; CHAWATHE E GARCIA-MOLINA, 1997), shingsem, shingcom (BRODER, 1998), distância de edição (NIERMAN e JAGADISH, 2002), zimilarity

flooding (MELNIK et al., 2002), zhingles (BUTTLER, 2004) e Serie temporal (FLESCA et al., 2005).

O QUADRO 14, a seguir, descreve o foco e a aplicação de cada uma dessas medidas.

QUADRO 14 – Principais métricas de Similaridade.

MÉTRICA	**TRABALHO**	**FOCO**	**DADOS E/OU ESTRUTURA**
edge cover	CHAWATHE E GARCIA-MOLINA, 1997	dados estruturados	ambos
shingsem, shingcon	BRODER, 1998	frases e blocos de texto	dados
distância de edição	NIERMAN E JAGADISH, 2002	documentos XML	estrutura
similarity flooding	MELNIK et al., 2002	grafos	estrutura e/ou dados
Shingles	BUTTLER, 2004	documentos XML	estrutura
série temporal	FLESCA et al., 2005	documentos XML	estrutura

Fonte: Adaptado de GONÇALVES e MELLO, 2006.

Nesta pesquisa optou-se pelo coseno por se tratar de uma medida simples, que atende com eficiência o propósito do experimento.

2.4.3.4 Cross-validation

Cross-validation consiste em uma prática estatística na qual se divide o conjunto dos dados em subconjuntos geralmente de tamanho semelhante. Então se realiza o teste em um subconjunto; em seguida utilizam-se os outros subconjuntos para validar e confirmar o modelo/teste obtido com o subconjunto inicial.

O subconjunto utilizado para iniciar o modelo ou teste é denominado conjunto de treinamento e os outros subconjuntos são denominados conjuntos de testes.

A técnica utilizada nesta pesquisa é a *K-fold cross-validation,* na qual o conjunto de dados é dividido em K subconjuntos. Um subconjunto é utilizado como conjunto de treinamento e nos outros K-1 subconjuntos, essa técnica é repetida K vezes (denominada parâmetro *folds*) sendo que cada teste utiliza dos K subconjuntos para treinamento.

Os resultados de cada teste são combinados para produzir um único valor. Esse método traz como vantagem o fato de não se ter que determinar e fixar os dados utilizados no conjunto de treinamento. A prática com 10 *folds* conhecida como *10-fold cross-validation* é comumente utilizada e é a escolhida para esta pesquisa.

2.4.3.5 Definindo o número de conglomerados

Muitos algoritmos de agrupamento requerem como parâmetro predefinido o número de grupos, ou então outro parâmetro para definir a granularidade. A definição do número de grupos pode apresentar dificuldades de acordo com o conjunto de medidas e técnicas utilizadas. Existem alguns métodos e algoritmos para definir a quantidade de grupos de forma automática. Como por exemplo: método baseado na distância, dendrograma, curvas de sihouette, Bem-Hur, Elisseeff e Guyon.

3 METODOLOGIA E REALIZAÇÃO DA PESQUISA

O objetivo deste capítulo é descrever o experimento que deu aportes quantitativos para atingir os objetivos propostos por esta pesquisa.

3.1 Ferramentas

Duas ferramentas utilizadas no experimento e de fundamental importância nesta pesquisa estão descritas nesta seção: o OGMA e o WEKA.

3.1.1 OGMA: ferramenta para análise de texto

Nesta pesquisa optou-se pelo desenvolvimento de uma ferramenta própria devido às limitações das ferramentas disponíveis discutidas no item 2.3.2 - extração de sintagmas nominais. O OGMA (FIG. 11) possibilitou a aplicação do experimento proposto na metodologia, destacando a possibilidade de análise de um *corpus* maior devido à total automatização.

O nome Ogma foi dado em homenagem ao deus celta Ogma (nome reduzido de *Ogmious*) (FIG. 10). Este deus criou mecanismos de linguagem e engrandeceu a comunicação do povo celta.

FIGURA 10 – Figura de Ogma.
Fonte: http://www.godchecker.com

O aplicativo foi desenvolvido na ferramenta *visual studio .NET* em linguagem C#. Por se tratar de uma ferramenta para análise de texto optou-se pelo desenvolvimento em modo texto, o que não impede que sejam desenvolvidas interfaces gráficas posteriormente.

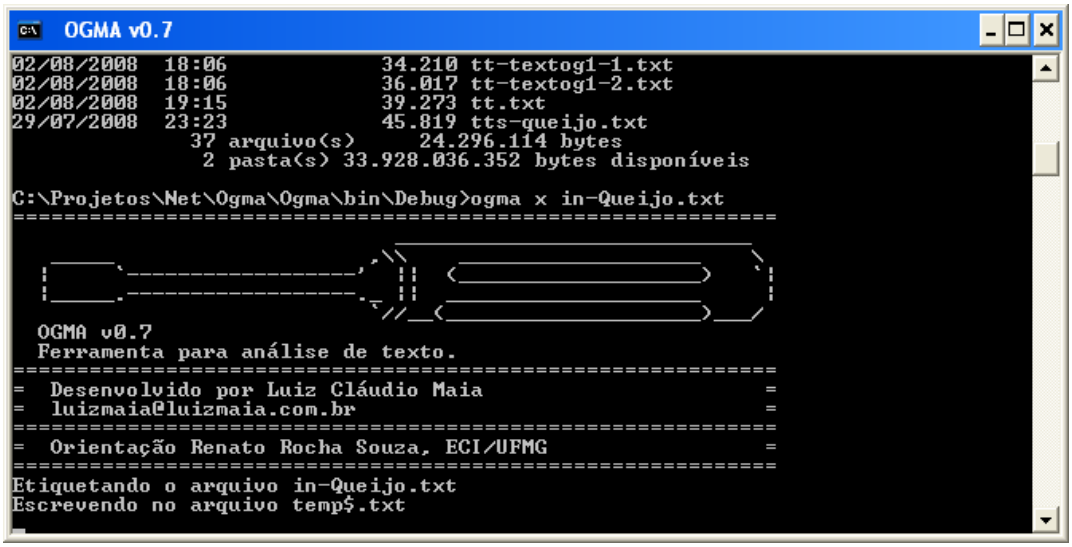


FIGURA 11 – Tela do Ogma.
Fonte: Elaborada pelo autor desta tese

O primeiro desafio na construção dessa ferramenta foi à elaboração de um léxico da língua portuguesa completo o suficiente para permitir análises e conseqüente etiquetagem do texto.

A primeira etapa para a construção desse dicionário foi a adaptação de arquivos com o vocabulário utilizado pelo BR/ISPELL[24]. Essa ferramenta foi desenvolvida para verificar a ortografia de projetos de código aberto. Através da adaptação desses arquivos foi possível a construção de um arquivo de dados (optou-se pelo uso do access) com uma tabela de 41978 nomes e adjetivos (FIG. 12).

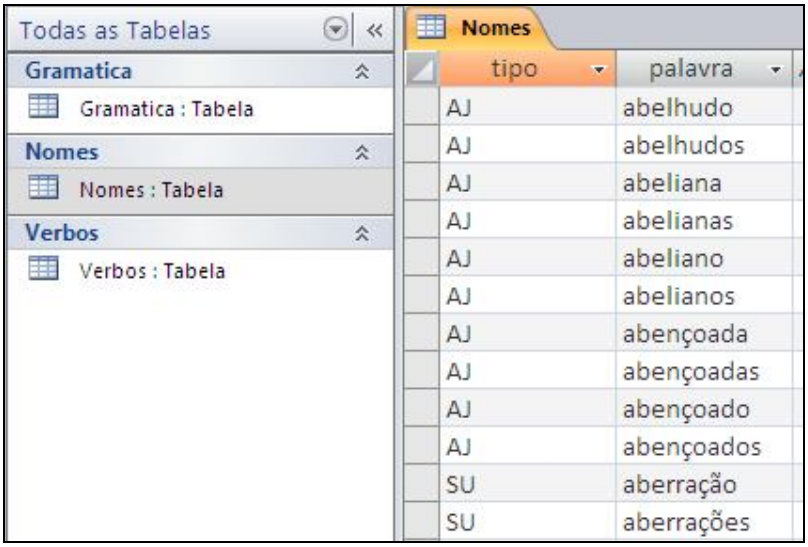


FIGURA 12 – Dicionário do Ogma: Tabela de nomes e adjetivos.
Fonte: Elaborada pelo autor desta tese.

[24] Para mais informações: http://www.ime.usp.br/~ueda/br.ispell

Outro item necessário era a identificação dos verbos. Utilizando a ferramenta conjugue[25] e uma base de dados de 5000 verbos conseguiram-se reunir em outra tabela do banco de dados 292.720 verbos. Devido a regras de identificação de sintagmas nominais utilizadas nesta pesquisa também foi necessário identificar os verbos no particípio; esses verbos receberam uma identificação diferenciada 'VP' no lugar de 'VB'. A FIG. 13 demonstra parte da tabela de verbos, e o verbo 'abafado' mostra como a diferenciação foi feita.

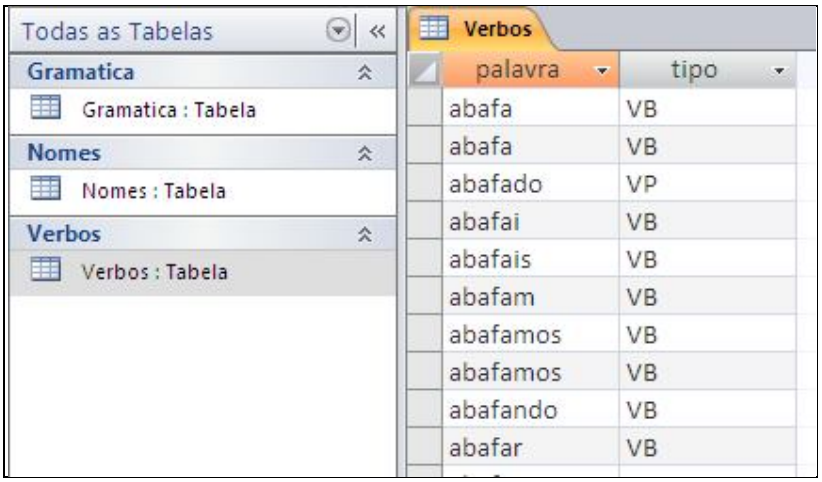


FIGURA 13 – Dicionário do Ogma: tabela de verbos.
Fonte: Elaborada pelo autor desta tese.

Finalmente através de um processo manual de digitação, tendo como base a gramática de Tufano (1990) conseguiram-se reunir 475 palavras de diversas classes gramaticais. Essas palavras são as mesmas que foram utilizadas para compor a lista de *stopwords* (ANEXO II).

[25] O conjugue é um programa para linux capaz de conjugar verbos da língua portuguesa, a partir de um banco de regras pré-definidas de conjugação.

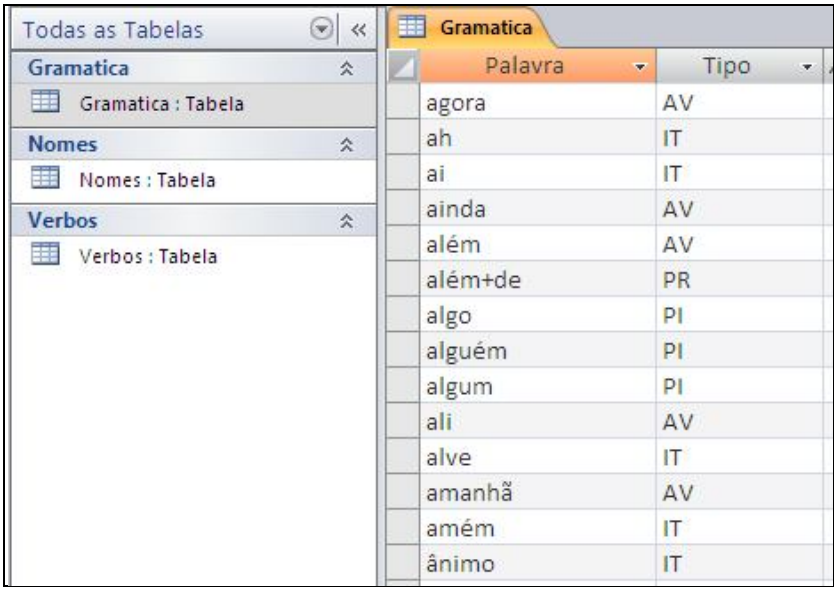


FIGURA 14 – Dicionário do Ogma: tabela gramatical.
Fonte: Elaborada pelo autor desta tese.

Esse processo de identificação das classes gramaticais foi útil na etiquetagem do texto analisado conforme o QUADRO 15 demonstra a associação das etiquetas utilizadas no Ogma com as etiquetas propostas no modelo ED-CER.

QUADRO 15 – Etiquetas utilizadas no Ogma e no ED-CER.

ETIQUETA OGMA	ETIQUETA ED-CER	CLASSE GRAMATICAL
AD	AR	Artigo definido
AI	AR	Artigo indefinido
AJ	AJ	Adjetivo
AV	AV	Advérbio
CJ	CO	Conjunção (Aditiva, adversativa, alternativa)
IT	*	Interjeição
NC	NU	Números cardinais
NM	NU	Números multiplicativos
NO	NU	Números ordinais
NP	NP	Nome próprio
NR	NU	Número romano
PS	PS	Pronome possessivo
PD	PD	Pronome demonstrativo
PI	PI	Pronome indefinido
PL	LG	Pronome relativo
PN	PN	Pontuação (exceto virgula)
PP	PP	Pronome pessoal
PR	PR	Preposições
SU	SU	Substantivo
VB	VB	Verbos
VG	CO	Vírgulas
VP	AJ	Verbos no Particípio

Fonte: Elaboradas pelo autor desta tese.

O texto deve ser tratado previamente, transformando expressões para melhorar o processo de etiquetagem. O seguinte conjunto é utilizado no Ogma:

do que → que
ao → a o
aos → a os
pela → por a
pelas → por a
pelos → por o
pelo → por e
neste → em este
nisto → em isto
nuns → em uns
numas → em umas
num → em um
numa → em uma
duns → de uns
dumas → de umas
dum → de um
duma → de uma
nos → em os
dos → de os
do → de o
das → de as
da → de a
nas → em as
no → em o
na → em a
ás → aos as
a fim de → a+fim+de
a que → a+que
a qual → a+qual
a respeito de → a+respeito+de
abaixo de → abaixo+de
acerca de → acerca+de
acima de → acima+de
além de → além+de
antes de → antes+de
ao invés de → ao+invés+de
ao redor de → ao+redor+de
apesar de → apesar+de
as quais → as+quais
até a → até+a
bem como → bem+como
como também → como+também
como um → como+um
de acordo com → de+acordo+com
debaixo de → debaixo+de
defronte de → defronte+de
dentreo de → dentreo+de
depois de → depois+de
diante de → diante+de
e não → e+não
em cima de → em+cima+de
em face de → em+face+de
em frente a → em+frente+a
em frente de → em+frente+de
em lugar de → em+lugar+de
em vez de → em+vez+de
mas ainda → mas+ainda
mas também → mas+também
não obstante → não+obstante
não obstante → não+obstante
não sí → não+sí
não só → não+só
no caso de → no+caso+de
no entanto → no+entanto
o qual → o+qual
o que → o+que
os quais → os+quais
para com → para+com
perto de → perto+de
por conseguinte → por+conseguinte
por isso → por+isso
por trás de → por+trás+de

O próximo desafio, depois de elaborado o dicionário, foi o tratamento das ambigüidades. Por exemplo, a palavra “mato” pode ser verbo ou substantivo dependendo do contexto e da posição.

Exemplos:

*Eu **mato** o rato*

e

*O **mato** estava grande*

Para contornar essa dificuldade o OGMA formou uma lista com todas as combinações encontradas e submeteu frase a frase às regras para extração dos SNs.

Texto etiquetado pelo Ogma:

1) *Eu/**PP** mato/**VBSU** o/**AD** rato/**AJSU***
2) *O/**AD** mato/**VBSU** estava/**VB** grande/**AJ***

Então o OGMA submeteu, às regras de extração definidas, duas versões de cada frase. No caso do primeiro exemplo foram enviadas duas possibilidades:

*Eu/**PP** mato/**VB** o/**AD** rato/**AJ***

*Eu/**PP** mato/**SU** o/**AD** rato/**AJ***

Os sintagmas nominais encontrados se integram em uma lista geral de sintagmas nominais da frase, eliminando os duplicados. Esse tratamento possibilitou resolver o problema da ambigüidade de forma bem eficiente.

Para extrair os SNs, o seguinte conjunto de regras foi utilizado, regra por regra na ordem em que aparecem no QUADRO 16:

QUADRO 16 – Regras de extração de SN do método OGMA.

AR → AD	
AR → AI	
AJ → VP	AV →AV ad
NU → NR	MD →AV MD
NU → NC	MD →MD co MD
CO → VG	NS →NS MD
CO → CJ	NS →MD NS
de → AR	NS →NS pr NS
de → PD	NS →NS pr de NS
de → PI	NS →NS co NS
qu → AJ	NS →NS co de NS
qu → NU	NS →AV NS
qu → PS	SN →de SN
ad → AV	
co → CO	
pr → PR	
re → SU	
de → PP	
re → NP	
NS →re	
MD →qu	
SN →NS	
AV →ad	

Fonte: Elaboradas pelo autor desta tese.

Uma regra foi modificada em relação às regras do ED-CER: nova regra: ***de*** ← ***PP*** substituindo a regra do ED-CER ***re*** ← ***PP***. A modificação dessa regra se deu para melhorar a extração de SN com base em diversos testes executados.

Sendo o OGMA uma ferramenta projetada para auxiliar na execução de todo o experimento proposto na metodologia, essa ferramenta deveria ser capaz de:

a) extrair os sintagmas nominais.

b) atribuir pesos aos sintagmas nominais extraídos de acordo com a freqüência em que aparecem no texto.
c) atribuir pesos aos sintagmas nominais extraídos de acordo com a freqüência em que aparecem no texto e dentro de outros sintagmas nominais.
d) Identificar a classe do sintagma nominal (CSN) extraído de acordo com a metodologia proposta por SOUZA (2005) e explicada no item 2.3.4 deste trabalho.
e) Calcular a pontuação de cada sintagma nominal extraído (relevância como descritor) utilizando a mesma metodologia.
f) Extrair termos e atribuir pesos de acordo com sua freqüência no texto.
g) Extrair termos, exceto os constantes na lista de *stopwords*, e lhes atribuir pesos de acordo com sua freqüência no texto.
h) Calcular a similaridade entre duas listas de termos (extraídas do documento) utilizando o coseno.

O algoritmo de cada uma das funções acima do OGMA encontra-se no ANEXO VIII.

Criaram-se diversos comandos, cada um podendo realizar uma função específica. O primeiro recurso trabalhado foi o de etiquetagem (anotação) do texto. Para utilizá-lo no Ogma, utilizaram-se três parâmetros: o primeiro o comando "E", o segundo o nome do arquivo origem em formato texto e o terceiro parâmetro, o arquivo de saída no qual seria armazenado o texto etiquetado.

Sintaxe:

"ogma e textooriginal.txt textoetiquetado.txt"

Em seguida criou-se a opção para extração de sintagmas nominais, opção "S". Esta opção analisou o arquivo de entrada, que já estava etiquetado e faz a aplicação das regras de extração. Os sintagmas nominais foram salvos em outro arquivo, na ordem em que aparecem. Para se utilizar essa função forneceu-se como primeiro parâmetro o "S" em seguida o nome do arquivo contendo o texto já etiquetado e o terceiro parâmetro o arquivo de saída.

Sintaxe:

"ogma s textoetiquetado.txt relacaosn.txt"

Também se criou uma opção para visualização rápida dos sintagmas nominais sem necessitar de passar pelas duas etapas anteriores. Esta opção "X" recebe apenas um parâmetro relativo ao texto que se pretende analisar.

Sintaxe:

"ogma x textooriginal.txt"

A próxima etapa de implementação foram as opções que permitiram gerar as tabelas de termo e peso para cada um dos métodos propostos.

A primeira opção criada gerou uma tabela contendo na primeira coluna a lista de todos os termos utilizados, na segunda coluna o número de vezes em que o termo apareceu em todo o texto, e na terceira coluna a porcentagem que aquele termo respondeu na composição de todo o documento. Para utilizar essa opção, "TT", deve-se fornecer como primeiro parâmetro o texto a ser analisado, e como segundo parâmetro o arquivo de saída.

Sintaxe:

"ogma tt textooriginal.txt tabtermos.txt"

Seguindo a mesma linha criou-se a opção para geração da tabela ignorando as palavras que apareciam na lista de *stopwords*. A relação completa das palavras que compuseram esta lista encontra-se no ANEXO II deste trabalho. Para gerar essa tabela, utilizou-se a opção "TTS", com dois parâmetros: no primeiro, o arquivo texto de entrada e no segundo o arquivo de saída no qual será armazenada a tabela. O arquivo de saída possuia também 3 colunas e seguiu a especificação da tabela gerada pela opção de termos, "TT".

Sintaxe:

"ogma tts textooriginal.txt tabtermos.txt"

A primeira opção de geração de tabelas relativas a sintagmas nominais foi a "TS". Nessa opção também foi gerada uma tabela com três colunas: a primeira contém uma lista de sintagmas nominais, a segunda o número de vezes que aquele sintagma nominal aparece no texto, e a terceira o cálculo em relação a todo o documento. A opção trabalha com dois parâmetros: o primeiro a relação de sintagmas nominais, um em cada linha, e o segundo o arquivo de saída. A relação de sintagmas nominais foi gerada pela opção "S" como visto anteriormente.

Sintaxe:

"ogma ts relacaodesn.txt tabsn.txt"

Esta opção, entretanto considera apenas sintagmas nominais únicos. Se dois sintagmas nominais são localizados, por exemplo, "Gestão" em um parágrafo e "Gestão do conhecimento" em outro a opção "TS" não considerará que o primeiro SN "Gestão"

apareceu duas vezes. Optou-se por criar uma segunda opção “TC” que contabilizasse também os sintagmas nominais dentro dos outros sintagmas nominais encontrados. Esta opção utiliza os mesmo parâmetros da opção anterior a “TS”.

Sintaxe:

“ogma tc relacaodesn.txt tabsn.txt”

A próxima tabela seria a opção “TR” que gera uma tabela com os sintagmas nominais pontuados de acordo com a metodologia proposta por SOUZA (2005). Para cumprir essa função, foi preciso etiquetar novamente a relação de sintagmas nominais, fornecida como parâmetro, para descobrir a classe de sintagma nominal (CSN), item necessário para o cálculo da pontuação. Essa etiquetagem é realizada internamente pelo OGMA para que os parâmetros permanecessem os mesmos das opções de geração de tabelas de sintagmas nominais. A tabela seguiu o mesmo formato das anteriores com o acréscimo de uma quarta coluna onde foi salva a classificação (CSN) encontrada para o sintagma nominal.

Sintaxe:

“ogma tr relacaodesn.txt tabsn.txt”

O processo de extração de SN realizada pelo método ED-CER resultou em uma lista de SNs na sua forma máxima. Nesta pesquisa utilizou-se também a opção de SN aninhados. O SN aninhado considera também como descritor os núcleos que compõem o SN máximo.

Por exemplo:

“A gestão do conhecimento nas organizações nacionais” corresponde a um SN máximo.

Entretanto existem três SN aninhados:

1) “A Gestão”

2) “conhecimento”

3) “as organizações nacionais.”

Realizou-se então uma adaptação no método “TR”, descrito acima, para considerar não só o sintagma nominal máximo, mas também todos os aninhados. Duas opções foram criadas: a “TCA” que realiza a extração dos SN mais os aninhados e a “TRA” que além desta extração os pontua na metodologia de Souza (2005).

Sintaxe:

“ogma tra relacaodesn.txt tabsn.txt” ou “ogma tca relacaodesn.txt tabsn.txt”

A próxima implementação foi relativa ao cálculo da similaridade (discutidas no item 2.4.3). Esse cálculo faz a aplicação da fórmula do coseno em duas tabelas, cada uma relativa a um arquivo, e obtém valores próximos a 1 à medida que os dois documentos comparados sejam semelhantes. Quanto mais próximo de 1 mais semelhante é um documento do outro. O resultado 1 significa que os documentos sejam iguais.

Sintaxe:

"ogma i tabela1.txt tabela2.txt"

Para facilitar a utilização do ogma, sem a necessidade de vários comandos e etapas para calcular a similaridade, criaram-se três opções adicionais "IT", "IC" e "IR" que calculam respectivamente a similaridade entre dois textos utilizando os métodos: por termos, por sintagmas nominais e por sintagmas nominais pontuados.

No ANEXO IV encontram-se exemplos de utilização, contendo arquivos de entrada e arquivos de saída, de cada uma das opções do OGMA descritas anteriormente.

A FIG. 15, a seguir, demonstra a utilização do OGMA para o cálculo da similaridade entre dois documentos.

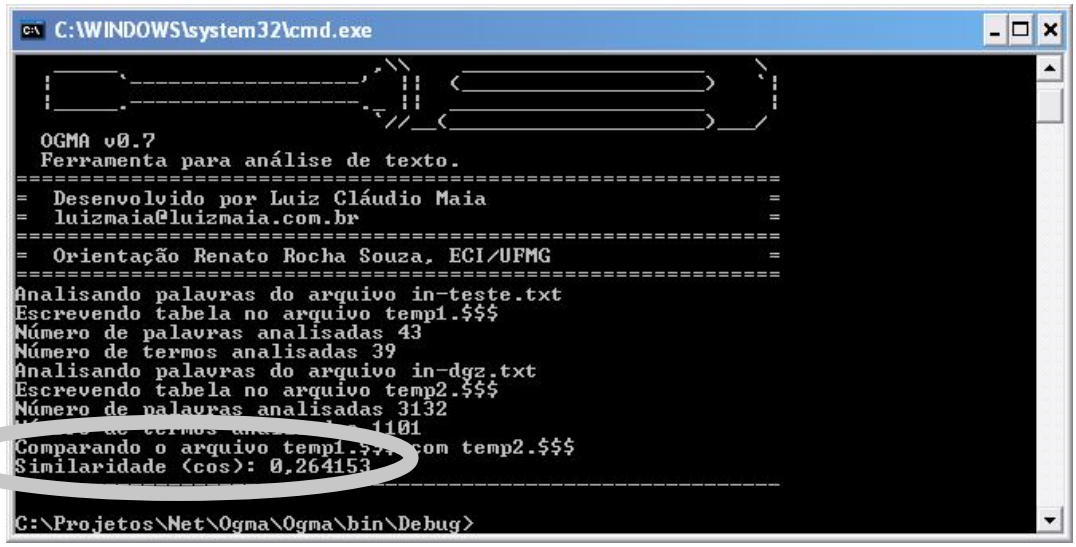


FIGURA 15 – Resultado do cálculo da similaridade entre dois documentos pelo Ogma.
Fonte: OGMA.

Comparações de outras ferramentas existentes para extração de sintagmas nominais com o OGMA foram realizadas. Quando aplicamos, ao texto abaixo, os métodos ED-CER, OGMA e VISL obtêm-se a extração de SN iguais para os métodos ED-CER (manual) e OGMA (automático), comprovando que a automação foi eficaz. Além disso, realizando uma comparação com o resultado obtido pelo VISL, obtém-se

ainda uma melhoria na identificação do sintagma nominal “o benefício”, na qual o VISL divide erroneamente em dois SN.

QUADRO 17 – Comparação entre extração de SN pelo ED-CER, OGMA e VISL.

Texto original	O novo cálculo das aposentadorias resulta em valores menores do que os atuais para quem perde o benefício com menos tempo de contribuição e idade.
Sintagmas extraídos pelo ED-CER	O novo cálculo de as aposentadorias; valores menores-do-que os atuais; o benefício com menos tempo de contribuição e idade;
Sintagmas extraídos pelo Ogma	O novo cálculo de as aposentadorias; valores menores; o benefício com menos tempo de contribuição e idade;
Sintagmas extraídos pelo VISL	% o novo cálculo de as aposentadorias % valores menores do que os atuais % o benefício % menos tempo de contribuição e idade

Fonte: Elaborado pelo autor da tese.

Outros testes comparativos entre o VISL e o OGMA foram realizados, inclusive para aprimoramento e ajuste das regras. Alguns dos resultados estão no ANEXO I desta tese.

3.1.2 WEKA - Waikato environment for knowledge analysis

O programa Weka - *Waikato environment for knowledge analysis* começou a ser escrito em 1993, usando Java, na Universidade de Wakato localizada na Nova Zelândia.

A licença utilizada pelo Weka é a *General public license* – GPL, sendo o pacote Weka formado por um conjunto de implementações de algoritmos de diversas técnicas de mineração de dados e textos.

O Weka agrega diversos algoritmos provenientes de diferentes abordagens/paradigmas na subárea da inteligência artificial dedicada ao estudo da aprendizagem por parte de máquinas. Entre esses, algoritmos de classificação e agrupamento.

Abaixo a FIG. 19 corresponde à tela do *Weka Explorer* com os dados do *corpus* JORNAIS04 em análise.

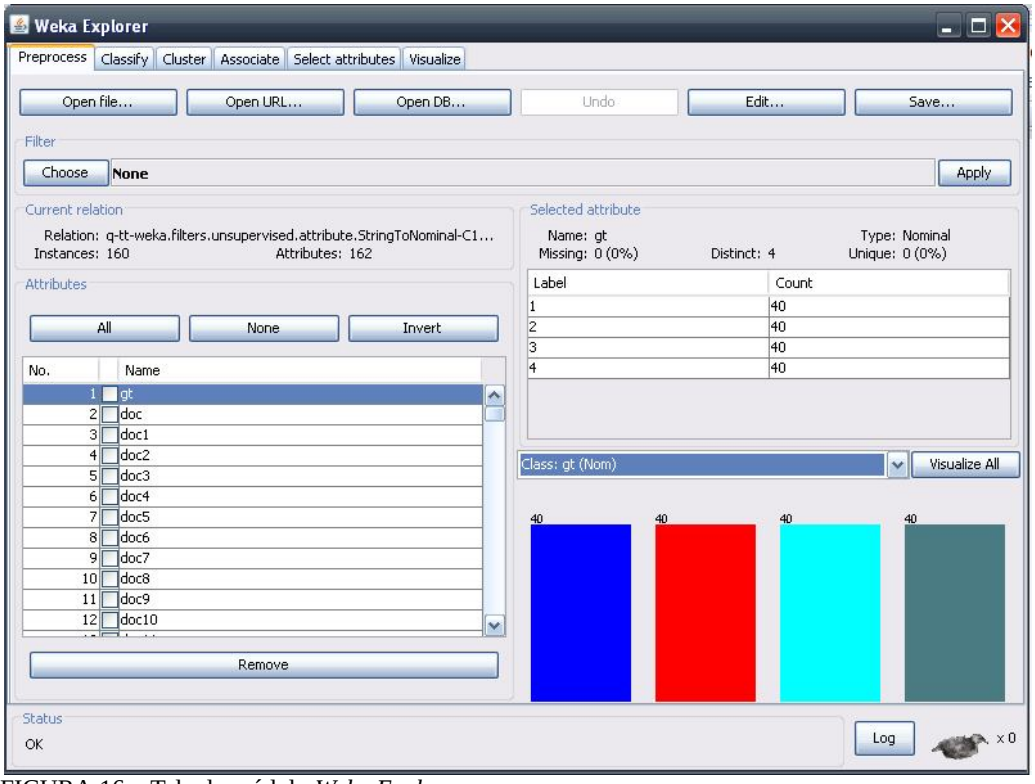


FIGURA 16 – Tela do módulo *Weka Explorer*.
Fonte: *Weka Explorer*.

3.2 *Corpora* – coleção de teste

Nesta pesquisa foram utilizados dois *corpora*:

O primeiro constituído de 50 artigos selecionados do Encontro Nacional de Pesquisa em Ciência da Informação - ENANCIB 2005. Neste congresso os trabalhos apresentados foram divididos em grupos de trabalho (GT) de acordo com o assunto tratado no artigo.

Dessa forma, selecionaram-se aleatoriamente 10 documentos de 5 grupos de trabalhos – GT.

Foram selecionados os cinco primeiros grupos:

GT 1: Estudos históricos e epistemológicos da informação

GT 2: Organização do conhecimento e representação da informação

GT 3: Mediação, circulação e uso da informação

GT 4: Gestão de unidades de informação

GT 5: Política, ética e economia da informação

A relação com o título e autor de cada um dos 50 artigos, encontra-se no ANEXO II desta pesquisa. Os artigos foram convertidos do formato original PDF em

arquivos texto pela ferramenta *Adobe Acrobat*. Os arquivos resultantes (formato texto) ficaram entre 21 kbytes, 3156 termos (texto 9 do GT 5) e 213 kbytes, 29857 termos (texto 10 do GT 2). O QUADRO 18 possui a média de termos dos textos selecionados dos cinco GTs.

A esse *corpus* deu-se o nome de "ENANCIB05".

QUADRO 18 – Composição do corpus ENANCIB05.

SEÇÕES	NÚMERO TEXTOS	NÚMERO TERMOS ÚNICOS	MÉDIA
GT 1	10	21280	2128
GT 2	10	21504	2150
GT 3	10	18913	1891
GT 4	10	18756	1875
GT 5	10	20039	2003
TOTAL	50	100492	2009

Fonte: Elaborado pelo autor da tese.

Do total de 100492 termos, 22320 são distintos e 6642 (7%) são únicos.

Também se optou por utilizar um segundo *corpus*, formado por textos menores e de conteúdo jornalístico. O objetivo foi avaliar o comportamento em um *corpus* diferente e com conteúdo bem definido em relação aos assuntos.

Para isso extraiu-se do site do jornal **Hoje em Dia** todas as notícias de 2004. Em seguida realizou-se um trabalho de classificação manual, trocando o nome dos arquivos de acordo com o caderno do qual o texto foi retirado. Após esse processo, realizou-se seleção aleatória de 40 notícias dos seguintes cadernos: Informática, Turismo e Veículos.

Para compor este *corpus*, e dificultar a etapa do agrupamento automático, decidiu-se adicionar mais um tema. Utilizou-se outro corpus que possuísse essa separação por temas disponibilizado pela linguateca, o TeMario[26].

Do TeMario foram retirados os textos do **Jornal do Brasil**, seção Internacional (Internacional 20 textos, 12.098 termos e média de termos 604) e **Folha de São Paulo**, seção Mundo (20 textos, 13.739 termos e média de termos 686) para compor esta nova temática.

O novo *corpus* denominado "JORNAL04", segundo deste experimento, ficou então com 160 textos divididos entre quatro temas: Informática, Turismo, Veículos e

[26] Para mais informações: http://www.linguateca.pt/Repositorio/TeMario/

Mundo. O QUADRO 19, abaixo, sintetiza esses dados, mostrando também o número de palavras únicas por seção e o número médio de palavras únicas por texto de cada seção.

QUADRO 19 – Composição do corpus JORNAIS04.

SEÇÕES	NÚMERO TEXTOS	NÚMERO TERMOS ÚNICOS	MÉDIA
Informática	40	15182	379
Turismo	40	16119	402
Veículos	40	9156	228
Mundo	40	12437	646
TOTAL	160	52894	330

Fonte: Elaborado pelo autor da tese.

3.3 Experimento prospectivo

O experimento prospectivo foi aplicado somente sobre o corpus ENANCIB05, e constituiu-se das seguintes etapas, sintetizadas na FIG. 17, abaixo:

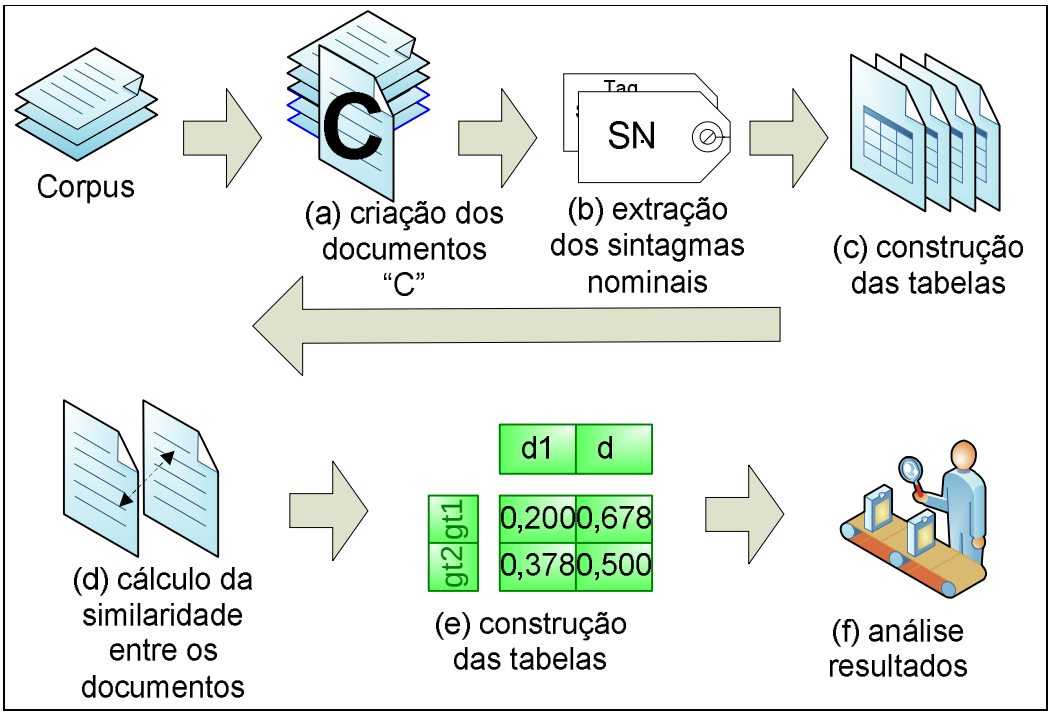


FIGURA 17 – Etapas do experimento prospectivo.

Fonte: Elaborada pelo autor da tese.

a) Criação de um arquivo denominado "C" com a concatenação dos 5 primeiros arquivos de cada GT.
b) Extração dos sintagmas nominais dos textos analisados. (item 3.5.1)
c) Construção das tabelas de descritores com os pesos por: termo, termos sem *stopwords*, sintagmas nominais e sintagmas nominais pontuados. (item 3.5.2)
d) Cálculo do índice de similaridade entre as tabelas de cada arquivo. (item 3.5.3)
e) Criação de 4 quadros comparativos dos arquivos de 6 a 10 de cada GT com o arquivo "C".
f) Análise dos resultados. (Cap. 4, desta tese)

Para o experimento prospectivo utilizou-se apenas o *corpus* do ENANCIB05. Tomaram-se os textos de 1 a 5 para constituírem um conjunto de treinamento que foi utilizado na técnica de agrupamento proposta. Dessa forma foi adicionado, ao *corpus* e a cada GT, um arquivo contendo os 5 primeiros textos concatenados de cada grupo. A esses arquivos se denominou “C”. Em seguida foi realizada a construção das 4 tabelas de termos e pesos para cada um dos 50 arquivos originais do *corpus* mais os cinco arquivos “C”. : - Uma tabela de termos, uma tabela de termos sem *stopwords*, uma tabela de sintagmas nominais e uma tabela de sintagmas nominais pontuados.

O objetivo foi simular o agrupamento dos arquivos de 6 a 10, utilizando um agrupamento prévio, os arquivos de 1 a 5, para cada um dos 4 métodos. Os arquivos de 6 a 10, total de 25 arquivos, foram agrupados comparando-se o valor de similaridade obtido com o arquivo “C” de cada GT.

O resultado desse experimento prospectivo está exposto e discutido no item seguinte (3.4).

3.4 Experimento consolidado

O experimento consolidado foi aplicado nos dois *corpora* e constituiu-se das seguintes etapas:

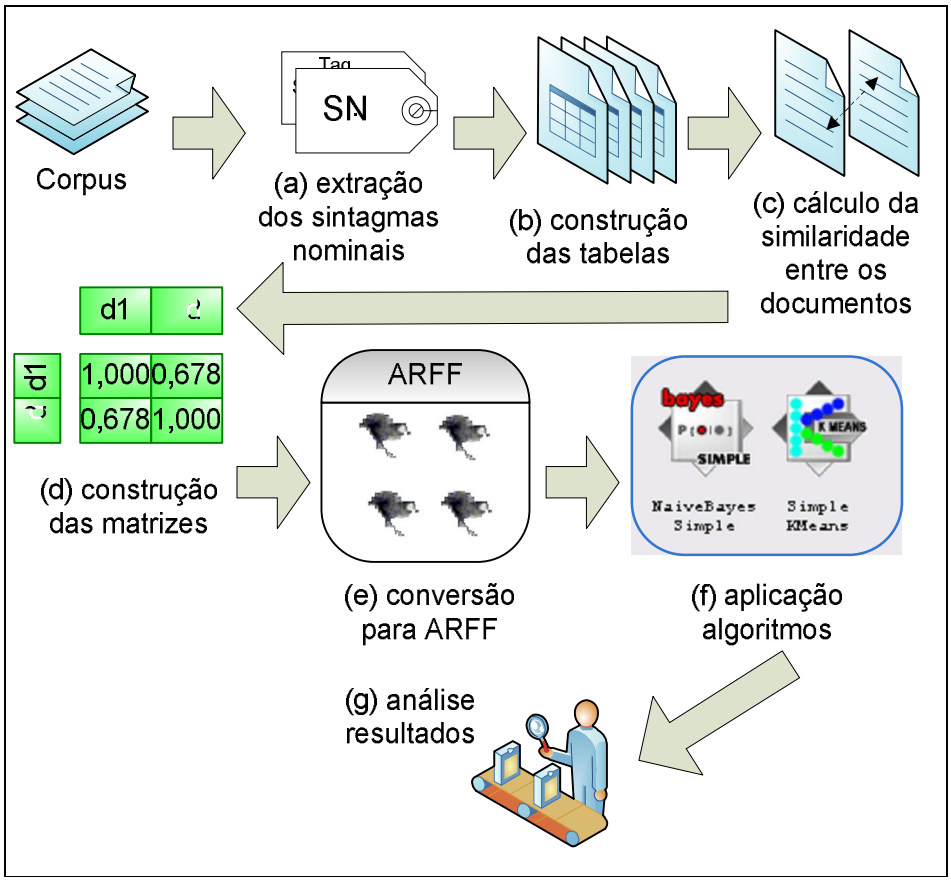


FIGURA 18 – Etapas do experimento consolidado.
Fonte: Elaborada pelo autor da tese.

a) Extração dos sintagmas nominais dos textos analisados. (item 3.5.1)
b) Construção das tabelas de descritores com os pesos por: termo, termos sem *stopwords*, sintagmas nominais e sintagmas nominais pontuados. No *corpus* JORNAIS04 utilizaram-se também os sintagmas nominais aninhados e sintagmas nominais aninhados pontuados. (item 3.5.2)
c) Cálculo do índice de similaridade entre as tabelas de cada arquivo. (item 3.5.3)

d) Criação de um arquivo para cada método, contendo a matriz com os resultados encontrados no item anterior. A matriz terá o número de linhas e o número de colunas igual ao número de documentos. (item 3.5.6)
e) Conversão dessas tabelas para o formato Arff do *software* Weka. (item 3.5.5 e item 3.5.6)
f) Aplicação dos algoritmos *Naive Bayes* e *Simplekmeans*. (item 3.5.7)
g) Análise dos resultados. (cap. 4, desta tese)

Para o experimento consolidado utilizou-se além do *corpus* ENANCIB05, do *corpus* JORNAIS04. Conforme apontado nos resultados obtidos no experimento prospectivo verificou-se a necessidade de um *corpus* com mais documentos e com temas mais definidos.

Também após a aplicação do experimento prospectivo, verificou-se a necessidade de se aprimorar a metodologia da análise dos resultados, e para isso utilizou-se o *software* WEKA gerando uma análise com um número maior de indicadores quantitativos.

O resultado deste experimento prospectivo está exposto e discutido no capítulo seguinte.

3.5 Etapas da pesquisa

A primeira etapa da pesquisa consistiu na construção e na validação da ferramenta OGMA. A segunda etapa foi submeter o *corpus* ENANCIB05 a um experimento prospectivo descrito no item 3.3. A terceira etapa foi submeter os *corpora* ENANCIB05 e JORNAIS04 ao experimento consolidado descrito no item 3.4.

Nos itens seguintes estão detalhados, etapa por etapa, os experimentos realizados.

3.5.1 Extração dos sintagmas nominais

A primeira etapa de ambos os experimentos foi a extração dos sintagmas nominais. Para extrair os sintagmas nominais dos arquivos utilizou-se a ferramenta OGMA. Primeiro a tarefa de etiquetar (marcar) cada um dos arquivos do corpus foi realizada, através da opção “E” do OGMA. Em seguida os arquivos etiquetados foram

submetidos a outro comando do OGMA, o “S” que, com base nas regras estabelecidas de composição de sintagmas nominais, realizou a extração dos mesmos.

3.5.2 Construção das tabelas de descritores e pesos

Nesta etapa utilizaram-se as opções de geração de tabelas do OGMA. As tabelas de termos e termos sem *stopwords* foram geradas através do comando “TT” e “TTS”, respectivamente. O parâmetro necessário para essa opção do OGMA foram os arquivos com os textos originais.

Para gerar as tabelas de sintagmas nominais foram utilizadas as opções do OGMA: “TC”, “TR”, “TCA” e “TRA”.

A opção “TC” recebeu como parâmetro um arquivo com o resultado da extração dos sintagmas nominais e a partir dele gerau uma tabela com a quantidade de vezes que cada um aparecesse no texto.

A opção “TR” também recebeu como parâmetro um arquivo com o resultado da extração dos sintagmas nominais. O OGMA realizou a etiquetagem desse arquivo para poder identificar a classe do sintagma (CSN). Após a identificação da classe de cada sintagma nominal o programa realiza o cálculo da pontuação (SOUZA, 2005) gerando a tabela de “sintagmas nominais pontuados”.

As opções “TCA” e “TRA” realizaram o mesmo procedimento do “TC” e “TRA”, entretanto considerando também os SN aninhados.

3.5.3 Cálculo da similaridade entre as tabelas

Para calcular a similaridade (através do Coseno) utilizou-se a opção “I” do OGMA. O OGMA recebeu dois parâmetros que correspondessem ao nome dos arquivos das duas tabelas que se pretendiam comparar. Essa opção resultou em valores entre 0 (documento nada similar) e 1 (documento igual).

No *corpus* ENANCIB05, calculou-se a similaridade entre todos os 55 arquivos, utilizando-se cada uma das 4 tabelas geradas, totalizando, portanto 3025 comparações para cada tipo de tabela e 12100 comparações entre arquivos no total.

No *corpus* JORNAIS04, calculou-se a similaridade entre todos os 160 arquivos, utilizando cada uma das 6 tabelas geradas, totalizando, portanto 25600 comparações para cada tipo de tabela e 153600 comparações entre arquivos no total.

3.5.4 Automação das etapas pelo OGMA

Para realizar todo o procedimento foi criado um arquivo de lote para a realização de grande parte das etapas:

```
@ECHO OFF
FOR %%a IN (texto*.txt) DO ogma e %%a e-%%a
FOR %%a IN (texto*.txt) DO ogma s e-%%a s-%%a
FOR %%a IN (texto*.txt) DO ogma e s-%%a se-%%a
FOR %%a IN (texto*.txt) DO ogma tt %%a tt-%%a
FOR %%a IN (texto*.txt) DO ogma tts %%a tts-%%a
FOR %%a IN (texto*.txt) DO ogma tc s-%%a tc-%%a
FOR %%a IN (texto*.txt) DO ogma tr s-%%a tr-%%a
FOR %%a IN (texto*.txt) DO ogma tca s-%%a tr-%%a
FOR %%a IN (texto*.txt) DO ogma tra s-%%a tr-%%a


FOR %%a IN (texto*.txt) DO FOR %%b IN (texto*.txt)
        DO ogma i tt-%%a tt-%%b tt-rst.txt

FOR %%a IN (texto*.txt) DO FOR %%b IN (texto*.txt)
        DO ogma i tts-%%a tts-%%b tts-rst.txt

FOR %%a IN (texto*.txt) DO FOR %%b IN (texto*.txt)
        DO ogma i tr-%%a tr-%%b tr-rst.txt

FOR %%a IN (texto*.txt) DO FOR %%b IN (texto*.txt)
        DO ogma i tc-%%a tc-%%b tc-rst.txt

FOR %%a IN (texto*.txt) DO FOR %%b IN (texto*.txt)
        DO ogma i tra-%%a tra-%%b tra-rst.txt

FOR %%a IN (texto*.txt) DO FOR %%b IN (texto*.txt)
        DO ogma i tca-%%a tca-%%b tca-rst.txt
```

FIGURA 19 – Arquivo de lote (BAT) para realizar a maioria das etapas do experimento automaticamente.
Fonte: Elaborada pelo autor da tese.

3.5.5 Arquivo ARFF

Para realizar o uso dos dados obtidos pelo software OGMA e aplicá-los aos algoritmos fornecidos pelo WEKA, foi necessário convertê-los para um formato próprio, o ARFF. Esse formato utilizado pelo WEKA, possui a descrição do tipo de cada atributo.

O formato ARFF consistiu basicamente em duas partes: A primeira contém uma lista de todos os atributos. Nessa parte foi definido o tipo do atributo, ou seja, os valores

que cada um pode representar. Quando os valores foram definidos, esses deveriam estar entre chaves e separados por vírgulas, como no caso abaixo do atributo grupo do *corpus* ENANCIB05.

```
@relation tc
@attribute grupo {1,2,3,4,5}
@attribute d1 numeric
@attribute d2 numeric
(...)
@attribute d49 numeric
@attribute d50 numeric
```

FIGURA 20 – Relação de atributos e tipos arquivo ARFF.
Fonte: Elaborada pelo autor da tese.

A segunda parte consistiu das instâncias ou dados a serem trabalhados, no caso desta pesquisa os valores de similaridade entre cada documento.

A FIG. 21, abaixo, mostra a parte inicial desses dados para o *corpus* ENANCIB05.

```
@data

1,1,0.120427,0.38819,0.236341,0.602135,0.233898,0.2858
1,0.120427,1,0.18425,0.242167,0.231225,0.148908,0.2082
1,0.38819,0.18425,1,0.392316,0.506721,0.295181,0.33826
1,0.236341,0.242167,0.392316,1,0.482736,0.092997,0.176
1,0.602135,0.231225,0.506721,0.482736,1,0.222973,0.321
1,0.233898,0.148908,0.295181,0.092997,0.222973,1,0.278
1,0.285848,0.208223,0.338263,0.176744,0.321943,0.27861
1,0.155883,0.114444,0.162276,0.13444,0.272922,0.118303
1,0.169311,0.191653,0.235254,0.346034,0.328081,0.11297
1,0.208496,0.20923,0.264382,0.409957,0.372951,0.024333
2,0.212423,0.15218,0.284623,0.147364,0.268231,0.269773
2,0.19752,0.169699,0.296686,0.252445,0.263641,0.173051
2,0.040793,0.089397,0.062395,0.086465,0.073076,0.05269
```

FIGURA 21 – Dados do arquivo ARFF.
Fonte: Elaborada pelo autor da tese.

3.5.6 Geração da matriz e conversão para utilização no software WEKA

Para utilizar o software WEKA foi necessário consolidar os valores encontrados na comparação entre os documentos em uma matriz, na qual cada linha (atuando como instância) representava um documento e cada coluna o documento a ser comparado (atuando como atributo).

Inicialmente, o arquivo com a matriz foi importado pelo WEKA, utilizando o formato *comma separed values (CSV)* e depois utilizando o próprio WEKA aplicou-se o filtro *StringToNominal* na primeira coluna (instância que indica a qual grupo pertence o documento), transformando aquele valor em um valor nominal que permitirá a análise pelos algoritmos *Naive Bayes* e *simplekmeans.* Após a aplicação do filtro converteu-se o arquivo em formato *Arff* utilizando o próprio WEKA.

3.5.7 Aplicação dos algoritmos de classificação/Naive Bayes e agrupamento/simplekmeans

Os dados então são submetidos aos algoritmos de *Naive Bayes* e *simplekmeans* implementados pelo Weka.

No *Naive Bayes* foi utilizado como parâmetro de teste o *Cross-validation* com *folds* igual a 10. Já no algoritmo *simplekmeans*, dois parâmetros de testes foram utilizados: o número de clusters igual ao número de temas do *corpus* e o valor de semente (*seed)* 10.

4 ANÁLISE DOS RESULTADOS

Este capítulo descreve os resultados encontrados na realização dos experimentos propostos no capítulo anterior. Durante toda a apresentação dos dados utilizaram-se as seguintes nomenclaturas para definir o método de análise (QUADRO 20):

QUADRO 20 – Siglas dos métodos de análise utilizados nos experimentos.

SIGLA	MÉTODO	DESCRIÇÃO
TT	Termo	As tabelas de descritores contêm todas as palavras do documento e seus respectivos pesos.
TTS	Termo sem *stopwords*	A tabela de descritores é construída com base em todas as palavras do documento, com exceção das presnetes na lista de *stopwords*.
TC	Sintagmas Nominais (máximos)	A tabela de descritores é construída com base nos sintagmas nominais extraídos de cada documento.
TR	Sintagmas Nominais (máximos) Pontuados	A tabela de descritores é construída de acordo com o cálculo realizado da pontuação como descritor de cada sintagma nominal.
TCA	Sintagmas Nominais (incluindo os aninhados)	A tabela de descritores é construída com base nos sintagmas nominais máximos e aninhados extraídos de cada documento.
TRA	Sintagmas Nominais (incluindo os aninhados) Pontuados	A tabela de descritores é construída de acordo com o cálculo realizado da pontuação como descritor de cada sintagma nominal máximo e aninhado.

Fonte: Elaborada pelo autor da tese.

4.1 Experimento prospectivo

A primeira etapa do experimento prospectivo correspondeu à geração das tabelas de indexação, contendo todos os descritores, para cada documento do *corpus*, sendo uma tabela para cada método. As tabelas foram utilizadas posteriormente no cálculo de similaridade entre os documentos.

De acordo com o método utilizado, o número de linhas (descritores) contidos na tabela foi diferente. O método que teve o maior número de linhas foi o de termos. Nesse

método cada palavra única do documento correspondeu a uma linha da tabela de descritores.

Na TAB. 1 estão os valores médios do número de linhas dessas tabelas para o *corpus* ENANCIB05.

TABELA 1 – Número médio de descritores por documento e método, *corpus* ENANCIB05

Método	Número médio de linhas da tabela indexação (sem doc. C)
TT – Termos	1503
TTS – Termos sem *stopwords*	1422
TC – Sintagmas Nominais	1030
TR – Sintagmas Nominais pontuados	1030

Fonte: Experimento.

O experimento prospectivo, que envolveu apenas a análise do corpus ENANCIB05, teve como resultado uma tabela contendo o valor de similaridade entre todos os documentos da coleção. A tabela descreveu nas colunas inicias as informações dos dois documentos comparados, o número do documento e o grupo de trabalho (GT) à qual pertencia. Para ilustração, as linhas iniciais são apresentadas na TAB. 2.

TABELA 2 – Similaridade entre documentos *corpus* ENANCIB05

Documento 1		Documento 2		Método de Análise			
N	GT	N	GT	SN (TC)	SN Pontuados (TR)	Termos sem *Stopwords* (TTS)	Todos os termos (TT)
1	1	1	1	1	1	1	1
1	1	2	1	0,120427	0,130456	0,497726	0,746339
1	1	3	1	0,38819	0,537161	0,628989	0,811402
1	1	4	1	0,236341	0,528576	0,632818	0,693262
1	1	5	1	0,602135	0,666978	0,67773	0,656516
1	1	6	1	0,233898	0,106388	0,450408	0,735704
1	1	7	1	0,285848	0,475754	0,508482	0,599546
(...)							

Fonte: Experimento.

Para obter valores que permitam uma comparação entre os métodos, o desvio, a média, o valor máximo (eliminando o valor 1) e o valor mínimo foram calculados, e estão representados na TAB. 3:

TABELA 3 – Comparações entre o valor de similaridade *corpus* ENANCIB05

	Método de Análise			
	SN (TC)	**SN Pontuados (TR)**	**Termos sem *Stopwords* (TTS)**	**Todos os termos (TT)**
Desvio	0,110434	0,099392	0,092177	0,082405
Média	0,198062	0,12763	0,402754	0,576589
Máximo	0,773313	0,697429	0,763096	0,811628
Mínimo	0,009653	0,002343	0,092177	0,082405

Fonte: Experimento.

Observaram-se, pela média dos valores obtidos em cada método, os índices de similaridade considerando todos os termos (TT), 0,57; e o termos sem *stopwords (TTS)*, 0,40, foram maiores do que os envolvendo sintagmas nominais (0,19 e 0,12).

Entretanto os valores máximos (0,77 para sintagmas nominais e 0,81 para todos os termos) não revelaram uma diferença tão grande em relação à média encontrada entre os métodos.

A seguir estão representados os quatro quadros comparativos, um para cada método propostos no experimento prospectivo. Foram apresentados os valores obtidos dos documentos de 6 a 10 de cada GT em comparação com o documento “C” de cada GT. O documento “C” representou o conjunto de treinamento e correspondeu aos documentos de 1 a 5. Os valores destacados em cinza são os maiores valores e indicaram o GT do qual o documento comparado foi atribuído automaticamente.

Na TAB. 4 podem-se examinar os valores de similaridade entre os 25 documentos, 5 de cada GT (linhas) com o documento "C" de cada GT (colunas). O método utilizado para construção das tabelas de comparação entre cada arquivo é o que considera todas as palavras do documento (TT).

TABELA 4 – Resultado agrupamento método TT, *corpus* ENANCIB05

		Corpus ENANCIB05 – Método TT						
	d	**GT 1**	**GT 2**	**GT 3**	**GT 4**	**GT 5**	**Desvio**	**Dif. GT**
GT 1	**6**	0,750343	0,725328	**0,791989**	0,719072	0,738966	0,028862	0,041646
	7	**0,690083**	0,629247	0,641269	0,578648	0,568865	0,049399	0
	8	**0,741243**	0,656451	0,725794	0,677969	0,689305	0,034805	0
	9	0,775432	0,700178	**0,78635**	0,69583	0,73335	0,04174	0,010918
	10	**0,848959**	0,646902	0,814639	0,712767	0,726771	0,081434	0
GT 2	**6**	0,757639	0,720962	**0,760034**	0,737618	0,701953	0,024664	0,039072
	7	0,640925	**0,731167**	0,691999	0,654152	0,657496	0,036557	0
	8	0,589787	**0,63306**	0,626489	0,595743	0,623775	0,019584	0
	9	0,634493	0,668233	**0,671527**	0,655756	0,646823	0,015293	0,003294
	10	0,63578	**0,72008**	0,696591	0,706227	0,653947	0,035958	0
GT 3	**6**	0,628952	0,670866	**0,679056**	0,677154	0,625252	0,026818	0
	7	0,781671	0,74428	**0,832862**	0,780207	0,765036	0,032763	0
	8	0,746251	0,732203	**0,792035**	0,755636	0,741427	0,023138	0
	9	0,701701	0,708082	**0,740395**	0,69158	0,735396	0,021408	0
	10	0,720086	0,6811	**0,777417**	0,768068	0,73443	0,038772	0
GT 4	**6**	0,775218	0,769214	**0,841536**	0,810214	0,769732	0,031919	0,031322
	7	0,613242	0,679439	**0,689411**	0,685999	0,661156	0,031366	0,003412
	8	0,505671	0,534687	0,536926	0,532222	**0,541093**	0,014051	0,008871
	9	0,669193	0,726998	0,752477	**0,795452**	0,719122	0,046311	0
	10	0,645066	0,655441	**0,706177**	0,687374	0,647467	0,0271	0,018803
GT 5	**6**	0,605039	0,638319	0,660447	0,63809	**0,665314**	0,02386	0
	7	0,74603	0,744117	**0,784365**	0,741034	0,76854	0,018862	0,015825
	8	0,728278	0,691787	**0,799753**	0,741109	0,753971	0,03932	0,045782
	9	0,645418	0,646231	0,65359	0,611102	**0,654841**	0,017911	0
	10	0,756069	0,778968	**0,841344**	0,79227	0,809774	0,032186	0,03157

Fonte: Experimento.

O método classificou corretamente 14 documentos de 25 (56%). Os números de documentos classificados corretamente por GT foram: três do GT1 (7, 8 e 10), três do GT 2 (7, 8 e 10), todos os cinco do GT 3, o documento 9 do GT4 e dois do GT5 (6 e 9).

A TAB. 5 apresenta os valores de similaridade obtidos dos mesmos documentos, porém utilizando a comparação das tabelas de palavras sem as *stopwords* (TTS).

TABELA 5 – Resultado agrupamento método TTS, *corpus* ENANCIB05

		Corpus ENANCIB05 – Método TTS						
	d	**GT 1**	**GT 2**	**GT 3**	**GT 4**	**GT 5**	**Desvio**	**Dif. GT**
GT 1	**6**	0,510876	0,565064	**0,577322**	0,469999	0,525484	0,043171	0,066446
	7	**0,627515**	0,525856	0,54927	0,444887	0,429662	0,080854	0
	8	**0,533368**	0,424315	0,481177	0,415101	0,432544	0,049594	0
	9	0,6771	0,601618	**0,703266**	0,555661	0,615645	0,059418	0,026166
	10	**0,794109**	0,535758	0,757641	0,601155	0,630082	0,108677	0
GT 2	**6**	**0,621971**	0,546541	0,602222	0,553743	0,498492	0,048788	0,07543
	7	0,447444	**0,599009**	0,517126	0,462554	0,467062	0,061919	0
	8	0,348091	**0,453826**	0,405494	0,361884	0,408333	0,041965	0
	9	0,518562	0,562748	**0,569887**	0,527256	0,518857	0,024891	0,007139
	10	0,475332	**0,587485**	0,557936	0,558587	0,476921	0,051721	0
GT 3	**6**	0,472252	0,529408	**0,553645**	0,531099	0,45781	0,041439	0
	7	0,632686	0,546675	**0,679478**	0,615996	0,598065	0,048542	0
	8	0,602936	0,59559	**0,664991**	0,612284	0,592873	0,029624	0
	9	0,451622	0,502187	0,496717	0,434678	**0,528192**	0,038463	0,031475
	10	0,628113	0,546522	**0,707645**	0,681421	0,638179	0,061591	0
GT 4	**6**	0,610659	0,612996	**0,711159**	0,672154	0,613091	0,045638	0,039005
	7	0,438718	0,512252	**0,538509**	0,530543	0,47986	0,041009	0,007966
	8	0,379074	**0,450244**	0,437603	0,424184	0,407007	0,027772	0,02606
	9	0,465446	0,574072	0,592054	**0,67009**	0,54679	0,074146	0
	10	0,435985	0,490189	**0,528072**	0,516087	0,450021	0,040219	0,011985
GT 5	**6**	0,390616	0,459151	0,456384	0,433255	**0,486023**	0,035736	0
	7	0,568353	0,617453	**0,635914**	0,57293	0,612424	0,029466	0,02349
	8	0,561026	0,485526	**0,65813**	0,558836	0,588791	0,062169	0,069339
	9	0,442058	**0,461939**	0,435729	0,37734	0,46138	0,03462	0,000559
	10	0,599072	0,620614	**0,732911**	0,633909	0,673245	0,052706	0,059666

Fonte: Experimento.

O método classificou corretamente 12 documentos de 25 (48%). Os números de documentos classificados corretamente por GT foram: três do GT1 (7, 8 e 10), três do GT 2 (7, 8 e 10), quatro do GT 3 (exceção do 9), o documento 9 do GT4 e um do GT5 (6).

A TAB. 6 apresenta os valores de similaridade obtidos dos documentos utilizando a comparação apenas pelos sintagmas nominais extraídos de cada um (TC).

TABELA 6 – Resultado agrupamento método TC, *corpus* ENANCIB05

		Corpus ENANCIB05 – Método TC						
	d	**GT 1**	**GT 2**	**GT 3**	**GT 4**	**GT 5**	Desvio	Dif. GT
GT 1	**6**	**0,323129**	0,183046	0,262268	0,198089	0,258322	0,056167	0
	7	**0,41499**	0,211974	0,242225	0,19782	0,210916	0,090585	0
	8	**0,250158**	0,209895	0,157748	0,125666	0,199749	0,048166	0
	9	**0,263295**	0,13422	0,226877	0,160756	0,222466	0,052652	0
	10	0,295862	0,597394	**0,653255**	0,616062	0,247711	0,193749	0,357393
GT 2	**6**	**0,426067**	0,277211	0,346601	0,305966	0,389961	0,060474	0,148856
	7	0,197692	**0,475209**	0,438381	0,424169	0,192602	0,138623	0
	8	**0,092577**	0,086516	0,051399	0,049439	0,070452	0,019694	0,006061
	9	**0,217095**	0,191432	0,173255	0,187043	0,169741	0,018772	0,025663
	10	0,192531	0,181609	0,184931	**0,196766**	0,170636	0,010156	0,015157
GT 3	**6**	0,35328	0,293714	0,33969	0,335225	**0,378765**	0,031001	0,039075
	7	**0,3174**	0,159008	0,281677	0,199275	0,275829	0,06521	0,035723
	8	0,449649	0,293526	0,438873	0,386172	**0,453175**	0,067528	0,014302
	9	0,15311	0,1282	0,15062	0,118311	**0,236537**	0,046657	0,085917
	10	**0,363698**	0,099203	0,136835	0,169908	0,174285	0,102329	0,226863
GT 4	**6**	**0,441864**	0,26748	0,428767	0,430025	0,430849	0,074153	0,011839
	7	0,386988	0,318343	0,384118	0,393798	**0,447731**	0,045971	0,053933
	8	0,116749	**0,150292**	0,098391	0,149883	0,104422	0,024763	0,000409
	9	**0,411997**	0,266746	0,339337	0,346938	0,34218	0,05146	0,065059
	10	0,301856	0,210612	0,313807	0,317316	**0,322591**	0,046813	0,005275
GT 5	**6**	**0,29107**	0,108409	0,137466	0,117033	0,185244	0,075041	0,105826
	7	0,377887	0,246122	0,37638	**0,395194**	0,387813	0,062277	0,007381
	8	0,297512	0,142613	0,27455	0,277428	**0,309074**	0,067287	0
	9	**0,076033**	0,075056	0,055374	0,059919	0,071284	0,009337	0,004749
	10	0,27595	0,17198	**0,301803**	0,233551	0,300532	0,054862	0,001271

Fonte: Experimento.

O método classificou corretamente 6 documentos de 25 (24%). Os números de documentos classificados corretamente por GT foram: quatro do GT1 (exceção do 10), um do GT 2 (7), nenhum do GT 3, nenhum do GT4 e um do GT5 (8).

A TAB. 7 apresenta os valores de similaridade obtidos dos documentos utilizando a comparação apenas pelos sintagmas nominais extraídos e pontuados de acordo com a classe (CSN) de cada um (TR).

TABELA 7 – Resultado agrupamento método TR, *corpus* ENANCIB05

		Corpus ENANCIB05 – Método TR						
	d	**GT 1**	**GT 2**	**GT 3**	**GT 4**	**GT 5**	**Desvio**	**Dif. GT**
GT 1	**6**	0,165185	0,128215	**0,186484**	0,142187	0,157506	0,022241	0,021299
	7	**0,582136**	0,193857	0,291461	0,15155	0,146198	0,182345	0
	8	**0,352336**	0,211151	0,222767	0,131868	0,173447	0,082944	0
	9	**0,536366**	0,162956	0,303049	0,135039	0,170838	0,166734	0
	10	0,547539	0,453294	**0,574472**	0,430952	0,2071	0,145011	0,026933
GT 2	**6**	**0,510815**	0,226227	0,311737	0,205659	0,211618	0,128942	0,284588
	7	0,18196	**0,382428**	0,347505	0,337993	0,143493	0,107992	0
	8	**0,057026**	0,051584	0,042049	0,029997	0,041311	0,010411	0,005442
	9	**0,263343**	0,159864	0,171151	0,138498	0,123036	0,054791	0,103479
	10	**0,22732**	0,15038	0,173853	0,148544	0,117068	0,041024	0,07694
GT 3	**6**	0,243654	0,209938	0,237742	**0,248958**	0,209539	0,018887	0,011216
	7	0,172098	0,096395	**0,184702**	0,13222	0,15032	0,034791	0
	8	0,270896	0,234491	**0,292871**	0,250759	0,250777	0,022474	0
	9	0,084385	0,090314	0,098906	0,083399	**0,20212**	0,050852	0,103214
	10	**0,159491**	0,077638	0,087414	0,111966	0,10445	0,031722	0,072077
GT 4	**6**	0,267715	0,205956	0,32055	0,334198	**0,360424**	0,06145	0,026226
	7	0,281739	0,245202	**0,30931**	0,304568	0,288699	0,025387	0,004742
	8	0,043816	**0,099809**	0,047589	0,097293	0,060312	0,026995	0,002516
	9	0,206297	0,180634	0,227448	**0,237844**	0,20156	0,022495	0
	10	0,247276	0,154703	0,228239	**0,251574**	0,186037	0,04189	0
GT 5	**6**	0,153263	0,081294	0,122036	0,082014	**0,155521**	0,036423	0
	7	0,209988	0,15598	**0,250955**	0,228169	0,202409	0,035291	0,048546
	8	0,194978	0,121891	0,222809	0,189677	**0,260411**	0,050945	0
	9	0,021283	0,02972	0,020482	0,021668	**0,0346**	0,00629	0
	10	**0,296944**	0,149259	0,277813	0,16898	0,208683	0,065187	0,088261

Fonte: Experimento.

O método classificou corretamente 11 documentos de 25 (44%). Os números de documentos classificados corretamente por GT foram: três do GT1 (7, 8 e 9), um do GT 2 (7), dois GT 3 (7 e 8), dois do GT4 (9 e 10) e três do GT5 (6, 8 e 9).

O resultado apontado pelo experimento prospectivo demonstrou que os métodos de análise por termo obtiveram melhor classificação do que os que utilizaram sintagmas nominais, sendo o método de sintagmas nominais pontuados atingiu 44% de

classificações corretas, apresentando uma melhora significativa em relação ao método que considerou apenas os sintagmas nominais que obtiveram 24%.

O experimento prospectivo possibilitou algumas conclusões:

1) O *corpus* ENANCIB05 não apresentou uma divisão temática certa, percebeu-se que mesmo os pesquisadores que participam do evento ficam em dúvida em relação ao GT mais pertinente para os trabalhos que irão submeter. Portanto para cumprir os objetivos desta pesquisa, foi necessário que no experimento consolidado se trabalhasse com outro corpus, que tivesse a característica de divisão por assunto bem definida.
2) O número de documentos também não se apresentou significativo; como metade do *corpus* foi utilizado para treinamento restaram apenas 25 documentos, 5 de cada grupo.
3) A metodologia de agrupamento baseada no maior valor de similaridade encontrado precisava ser melhorada.
4) Observou-se que a soma dos valores da coluna que apontava a diferença do maior valor para o valor do GT correto (TAB. 8) foram elevadas para os métodos que utilizam SN (TC e TR).

TABELA 8 – Soma da diferença entre o GT correto

Método	Soma da diferença entre o GT correto (última coluna)
TT – Termos	0,250515
TTS - Termos sem *stopwords*	0,444726
TC - Sintagmas Nominais	1,210752
TR - Sintagmas Nominais Pontuados	0,875479

Fonte: Experimento.

4.2 Experimento consolidado

Com as limitações apresentadas pelos resultados do experimento prospectivo foram realizadas as seguintes ações:

1) Consolidação do segundo *corpus* (o JORNAIS04) com um número maior de documentos e contendo documentos com conteúdo melhor definido para cada tema.
2) Utilização da ferramenta de análise WEKA, possibilitando melhor análise estatística.

Os documentos dos *corpora* ENANCIB05 e JORNAIS04 foram comparados entre si, utilizando os métodos propostos.

A aplicação dos algoritmos de *Naive Bayes* e *simplekmeans* através do WEKA nesses dados apresentaram os seguintes resultados consolidados na TAB. 9 para o *corpus* do ENANCIB05:

TABELA 9 – Resultado Weka para o *corpus* ENANCIB05

Método	***Naive Bayes***		***SimpleKMeans***	
	Classificados corretamente	%	Agrupados corretamente	%
TTS - Termos sem *stopwords*	26/50	52%	22/50	44%
TT – Termos	25/50	50%	21/50	42%
TR - Sintagmas Nominais Pontuados	23/50	46%	20/50	40%
TC - Sintagmas Nominais	12/50	24%	15/50	30%

Fonte: Experimento.

Os dados da TAB. 9 estão representados na FIG. 22, abaixo:

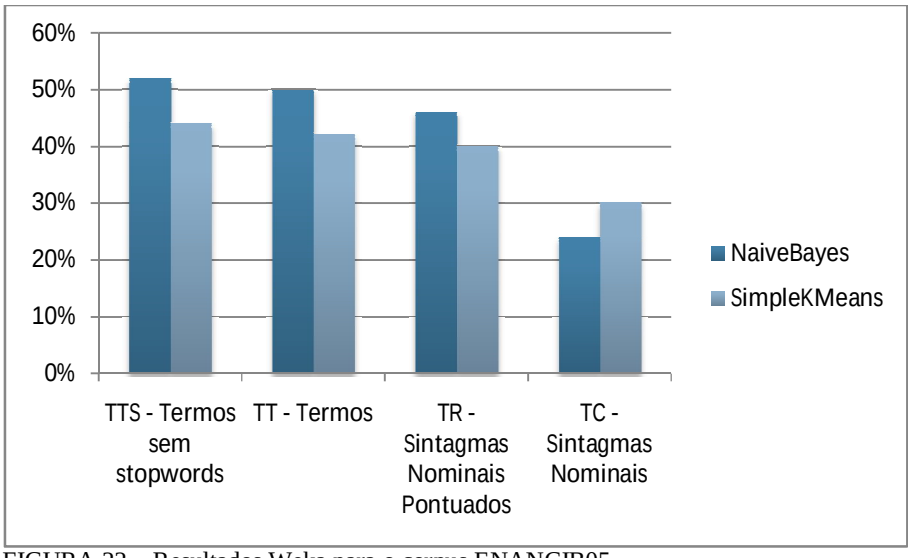


FIGURA 22 – Resultados Weka para o corpus ENANCIB05.
Fonte: Experimento.

Mesmo utilizando os algoritmos da ferramenta WEKA no corpus ENANCIB05, não foi possível obter conclusões sobre qual o melhor método, visto que a diferença do número de documentos classificados corretamente entre os métodos de termos e sintagmas nominais pontuados foi muito pequena. Apenas o método de sintagmas nominais simples apresentou resultados inferiores aos outros métodos.

Os resultados foram bem semelhantes aos encontrado no experimento prospectivo; entretanto o uso da técnica de *cross validation* para seleção do conjunto de treinamento utilizada pelo WEKA fez com que o método que usou termos sem *stopwords* tivesse um resultado um pouco melhor (52%) que no uso de termos apenas (%50), conforme a TAB. 10. No experimento prospectivo o corpus de treinamento era fixo, composto dos 5 primeiros artigos de cada GT.

TABELA 10 – Comparação experimento prospectivo x experimento consolidado

	Experimento prospectivo		**Experimento consolidado (*Naive Bayes*)**	
Método	**Doc. corretos**	**%**	**Doc. corretos**	**%**
TT – Termos	14/25	56%	25/50	50%
TS - Termos sem *stopwords*	12/25	48%	26/50	52%
TR - Sintagmas Nominais Pontuados	11/25	44%	23/50	46%
TC - Sintagmas Nominais	6/25	24%	12/50	24%

Fonte: Experimento.

Para o corpus JORNAIS04 foram utilizados, além dos quatro métodos anteriormente descritos (termos - TT, termos sem stopwords - TS, sintagmas nominais – TC e sintagmas nominais pontuados - TR), mais dois: um considerando os sintagmas nominais aninhados (TCA) e outro considerando os sintagmas nominais aninhados pontuados (TRA).

Para cada documento da coleção gerou-se uma tabela com a lista de descritores utilizados para a indexação do documento e os respectivos pesos. Essa tabela foi a referência na comparação entre os documentos.

Na TAB. 11 é apresentado o número médio de descritores das tabelas geradas a partir dos documentos do *corpus* JORNAIS04.

TABELA 11 – Número médio de descritores por documento e método, corpus JORNAIS04

Método	Número médio de descritores por documento
TT – Termos	331
TTS – Termos sem *stopwords*	296
TC – Sintagmas Nominais	160
TR – Sintagmas Nominais pontuados	160
TCA – SN Aninhados	245
TRA – SN Aninhados e pontuados	245

Fonte: Experimento.

O número médio de descritores por método de cada documento também é apresentado no gráfico da FIG. 23:

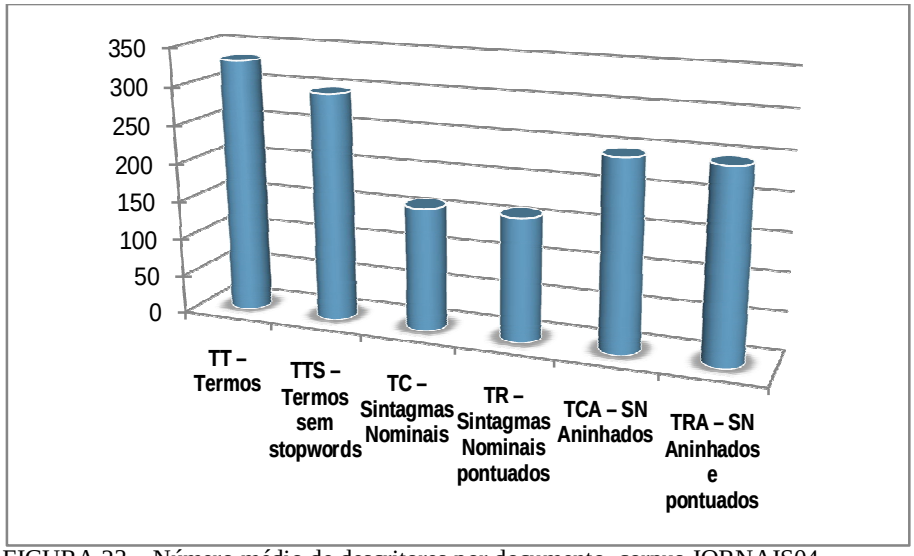


FIGURA 23 – Número médio de descritores por documento, corpus JORNAIS04.
Fonte: Experimento.

O número de descritores de cada tabela gerada para cada documento do *corpus*, pelos métodos, encontra-se no ANEXO VII.

O experimento, utilizando o WEKA algoritmo classificador *Naive Bayes*, foi repetido com o corpus JORNAIS04 e o resultado consolidado se encontra na TAB. 12.

TABELA 12 – Resultado Weka/*Naive Bayes*, corpus JORNAIS04

Corpus: JORNAIS04		
Método	***Naive Bayes***	
	Classificados corretamente	%
TS - Termos sem *stopwords*	147/160	91%
TC - Sintagmas Nominais	147/160	91%
TRA - SN Aninhados Pontuados	137/160	85%
TCA - SN Aninhados	136/160	85%
TR - Sintagmas Nominais Pontuados	132/160	82%
TT - Termos	106/160	66%

Fonte: Experimento.

O arquivo com o quadro comparativo de cada método também foi submetido ao algoritmo *SimpleKMeans* do WEKA. Os resultados encontram-se na TAB. 13:

TABELA 13 – Resultado Weka/*SimpleKMeans*, corpus JORNAIS04

Corpus: JORNAIS04		
Método	***SimpleKMeans***	
	Agrupados corretamente	%
TRA - SN Aninhados Pontuados	129/160	81%
TS - Termos sem *stopwords*	126/160	79%
TCA - SN Aninhados	109/160	68%
TC - Sintagmas Nominais	89/160	56%
TT - Termos	80/160	50%
TR - Sintagmas Nominais Pontuados	66/160	43%

Fonte: Experimento.

As informações geradas pelo aplicativo WEKA, algoritmo *Naive Bayes* estão no ANEXO V e as do *simplekmeans* no ANEXO VI.

Observa-se que os melhores resultados foram obtidos, utilizando a comparação entre tabelas de descritores contendo a relação termos sem *stopwords* (TTS) e as tabelas de sintagmas nominais aninhados e pontuados (TRA), utilizando o algoritmo de

classificação *Naive Bayes*. Ambos classificaram 147 documentos corretamente, um total de 91% do *corpus*.

Os resultados atingidos pelo *simplekmeans* para agrupamento foram inferiores ao classificador *Naive Bayes* em ambos os experimentos. A FIG. 24 demonstra o resultado alcançado nos seis métodos utilizando o *simplekmeans* e o *Naive Bayes* para o corpus JORNAIS04:

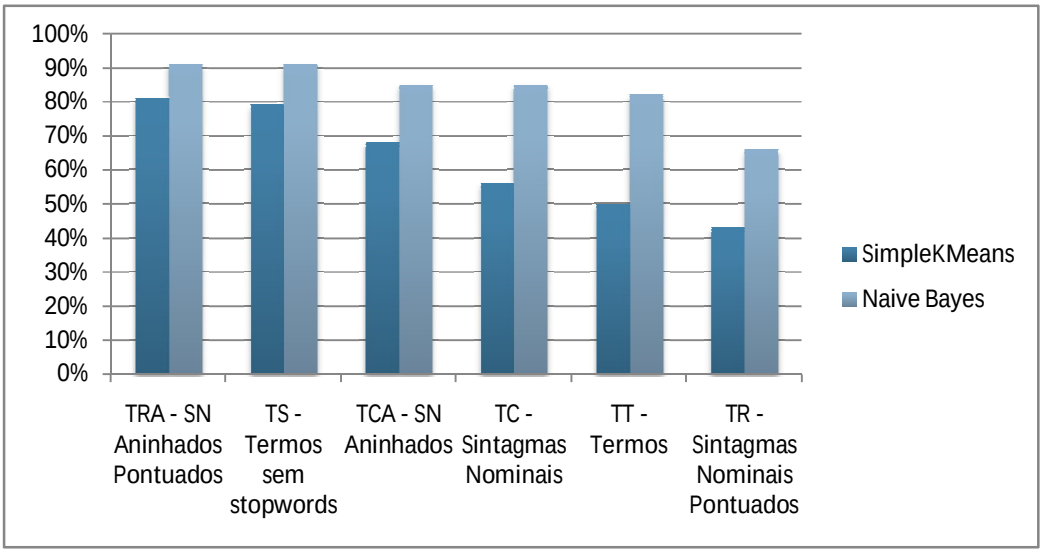


FIGURA 24 – Resultados Weka para o corpus JORNAIS04.

Os resultados dos experimentos apontam que os métodos que envolveram o uso de sintagmas nominais na classificação automatizada de documentos apresentaram índices semelhantes ao dos termos sem *stopwords*. Por exemplo, no corpus JORNAIS04, os sintagmas nominais e os termos sem *stopwords* atingiram o mesmo resultado com o *Naive Bayes*, obtendo 91% (147 documentos) de classificação correta. No algoritmo de cluster *simplekmeans*, o sintagma nominal aninhado e pontuado se apresentou *um* pouco melhor que o termo sem *stopwords;* classificando corretamente 129 documentos (81%) contra 126 (79%) dos termos sem *stopwords*.

Apesar de resultados similares, o uso de sintagmas nominais envolveu um processamento computacional muito maior que o uso de termos sem *stopwords*. Como visto, para o computador extrair os sintagmas nominais foi necessário todo um processo de etiquetagem (anotação) e aplicação de regras gramaticais ao documento, enquanto que no uso do método de termos sem *stopwords* o processo foi bem mais simples. Dessa forma, a relação custo *versus* benefício de se utilizarem sintagmas nominais na

classificação e no agrupamento de documentos eletrônicos não se apresentou interessante.

Um exemplo simples de comparação demonstra uma das dificuldades na classificação usando SN: Em dois textos, um contendo "a caneta azul falhou" e o outro "a caneta vermelha falhou", na extração dos SN teríamos:

texto1: "a caneta azul"

texto2: "a caneta vermelha"

Na comparação entre o texto1 e o texto2, como não existem sintagmas nominais em comum entre os textos, não é indicado nenhuma similaridade. Já na metodologia por termos teríamos 3/4 de semelhança, pois entre as quatro palavras que compõem a frase apenas o adjetivo que indica a cor da caneta seria desconsiderado.

Em contraponto, os piores resultados de classificação e agrupamento ficaram para o uso de todas as palavras (termos) que compõe o documento. Isto pode indicar a importância de se submeter o documento a um tratamento prévio antes da elaboração da tabela de termos e pesos, mesmo que seja somente a retirada das *stopwords*.

Por fim, o método que se utilizou de sintagmas nominais pontuados (SOUZA, 2005) na classificação, de forma geral, se mostrou mais eficiente que o uso de sintagmas nominais sem a aplicação da pontuação, nos experimentos. Além disso, também se observou que:

- Os resultados da pontuação são melhores se aplicados a tabelas que contenham não só sintagmas nominais máximos, mas também sintagmas nominais aninhados.
- Os pesos aplicados a cada classe de sintagma nominal (CSN) devem ser revistos de acordo com a linguagem utilizada no *corpus*. Por exemplo, em textos científicos (*corpus* ENANCIB05) existe um número maior de sintagmas nominais com mais de um qualificador do que em textos jornalísticos (*corpus* JORNAIS04).

5 CONSIDERAÇÕES FINAIS

Os documentos utilizados nos *corpora* desta tese foram elaborados através da linguagem natural. Essa linguagem em alguns casos foi ambígua semanticamente e apresentou problemas decorrentes da diferença do vocabulário usado nos textos. Os métodos de descoberta de conglomerados ou classificação se mostraram extremamente dependentes de técnicas de pré-processamento dos textos que visassem a padronizá-los, minimizando os problemas do vocabulário e representando seu conteúdo de forma mais correta e fácil de ser trabalhada pela máquina.

A presente pesquisa focou justamente esse ponto, buscando demonstrar que a utilização de sintagmas nominais é capaz de representar o conteúdo dos documentos, servindo como descritores ou características para o processo de classificação ou descoberta de conglomerados, melhorando a precisão desse processo.

Apesar dos resultados encontrados nesta pesquisa, devemos considerar o que Chomsky, principal responsável pela formulação da gramática gerativa, disse sobre hipóteses na área da lingüística:

> " (...) devemos admitir que nossa compreensão de fenômenos não triviais é sempre extremamente limitada. Se isto é verdadeiro para a Física, é mais verdadeiro ainda para a lingüística. Compreendemos fragmentos do real, e podemos estar certos de que toda hipótese interessante significativa só será parcialmente verdadeira (...)" (CHOMSKY, 1977, p.173 – tradução livre do autor)

Os objetivos especificados nesta pesquisa foram alcançados e podemos relacionar as principais contribuições desta tese:

- A criação de uma ferramenta inédita, o OGMA, para extração de sintagmas nominais da língua portuguesa utilizada no Brasil; (seção 3.1.1)
- Automação do método de pontuação de sintagmas nominais proposto por SOUZA (2005), conseqüentemente, automatizando processos de indexação e extração de descritores. (seção 2.3.4 e 3.5.4)
- A proposta de classificação de documentos descrita na metodologia, incluindo seus seis métodos: Ttrmos, termos sem *stopwords*, sintagmas nominais, sintagmas nominais aninhados, sintagmas nominais pontuados e sintagmas nominais aninhados pontuados. (seção 3.5.2)

- A sistematização do processo de descoberta e de análise de *clusters* de documentos utilizando sintagmas nominais, indicando as etapas e técnicas associadas a cada uma delas; (seção 3.5)
- A apresentação dos resultados de experimentos os quais avaliam diferentes métodos de construção de índices comparando-os entre si. (Cap. 4)

Este trabalho representa uma continuidade de estudos nas áreas de ciência da informação, computação e lingüística, e novos trabalhos podem a partir dos resultados alcançados aqui, elaborar novos experimentos. A ferramenta OGMA, a primeira ferramenta específica para extração de sintagmas nominais em português, estará disponível e facilitará novas pesquisas.

Durante este estudo, pesquisa e experimento realizado, novas hipóteses, sugestões e idéias surgiram. Algumas dessas estão apresentadas a seguir:

- Uso de sintagmas nominais em processos automatizados pelo OGMA de indexação e extração de palavras-chave ou descritores.
- Uso de sintagmas nominais na classificação de documentos utilizando novas propostas de metodologia.
- Utilização da metodologia posposta em um *corpus* diferente, variando a quantidade de documentos, número de termos ou mesmo o estilo de texto.
- Uso de sintagmas nominais na elaboração de metodologias que possibilitem a construção de ontologias[27].
- Elaboração de técnicas e ferramentas para descoberta de conhecimento em textos (*Knowledge Discovery from Text* – KDT).
- Utilização dos sintagmas nominais para criação de interfaces de recuperação de informação.
- Ampliação dos estudos envolvendo a alteração dos pesos na metodologia proposta por SOUZA (2005) e utilizada por este trabalho.

Certamente este não é o fim.

[27] Geralmente, o termo “Ontologia” com ‘O’ maiúsculo é empregado para representar o ramo da metafísica. Nos demais casos (analogias, na verdade), o termo “ontologia” é empregado com o ‘o’ minúsculo. A ontologia oferece uma estrutura básica (um conjunto de conceitos e termos que descrevem algum domínio) na qual uma base de conhecimento pode ser construída.

REFERÊNCIAS

AIRES, R. V. X. **Implementação, adaptação, combinação e avaliação de etiquetadores para o português do brasil.** Dissertação de Mestrado. Outubro, 2000.

ANDERSON, J.,; PEREZ-CARBALLO, J.. The nature of indexing: how humans and machines analyze messages and texts for retrieval. Part I: Research, and the Nature of Human Indexing. **Information Processing and Management**, n. 37, 2001. p. 231-254.

ARAÚJO JR, R.H.; **Precisão no processo de busca e recuperação da informação.** Tese Doutorado em Ciência da Informação. Brasília: Universidade de Brasília, 2006.

BAEZA-YATES, R.; RIBEIRO-NETO, B.. **Modern information retrieval.** New York: ACM Press, 1999.

BARRETO, A.A; Mudança estrutural na condição da escrita. **DataGramaZero - Revista de Ciência da Informação.** v. 9, n. 1, dez, 2008.

BICK, Automatic parsing of portuguese. In: García, Laura Sánchez (ed.), **Anais...** II Encontro para o Processamento Computacional de Português Escrito e Falado. Curitiba: CEFET-PR. 1996.

BOLLEN, J.; LUCE, R.. Evaluation of digital library impact and user communities by analysis of usage patterns. **D-Lib Magazine**, vol. 8, n. 6, jun., 2002.

BRODER, A. On the resemblance and containment of documents. In **SEQS: Sequences** '91, 1998.

BUTTLER, D. A short survey of document structure similarity algorithms**.** In: **International Conference on Internet Computing**, p. 3–9, 2004.

CALADO, P. P.; CRISTO, M.; MOURA, E. S.; GONÇALVES, M. A.; ZIVIANI, N.; RIBEIRO-NETO, B.. Linkage similarity measures for the classification of web documents. **Journal of the American Society for Information Science and Technology (JASIST)**, vol. 57, no. 2, p. 208-221, 2005.

CHAWATHE S. S.; GARCIA-MOLINA H. Meaningful change detection in structured **data** In: **Proceedings of the ACM SIGMOD International Conference on Management of Data.** 1997.

CHOMSKY, N. **Diálogos com Mitsou Ronat.** São Paulo: Cultrix, s.d., editado em francês pela Flammarion, 1977.

CHOMSKY, N. **Syntactic sctructures.** 3. ed., Paris: The Hague, 1969.

CROFT, W. B. **Advances in information retrieval.** London: Academic Publishers, 2000.

FLESCA, S.; MANCO, G.; MASCIARI E.; PONTIERI, L.; PUGLIESE, A. Fast detection of XML structural similarity. **IEEE Trans. Knowl. Data Eng**, v. 17, n.2, p.160–175, 2005.

FRANTZ, V.; SHAPIRO, J.; VOISKUNSKII, V. **Automated Information Retrieval: Theory and Methods.** San Diego, CA: Academic Press, 1997. 365 p.

GONZALEZ M.; LIMA V.L.S., Recuperação de Informação e Processamento da Linguagem Natural, **Minicurso**, Porto Alegre: PUC-RS, 2003.

GONÇALVES, R.; MELLO, R.S. Similaridade entre documentos semi-estruturados**.** In: **II ERBD – Escola Regional de Banco de Dados.** Passo Fundo - RS, 2006.

GREENBERG, J.. Metadata extraction and harvesting: a comparison of two automatic metadata generation applications. **Journal of Internet Cataloging**, v. 6, n.4, p. 59-82, 2004.

HARMAN, D.. Relevance feedback and other query modification techniques. In: William B. Frakes and Ricardo Baeza-Yates, editors, **Information Retrieval: Data Structures and Algorithms**, p. 241-263. Prentice Hall, 1992.

HARTER, S.P.. The Impact of Electronic Journals on Scholarly Communication: A Citation Analysis. **The Public-Access Computer Systems Review** v. 7, n. 5, 1996. Disponível on-line em http://info.lib.uh.edu/pr/v7/n5/hart7n5.html. Acesso em: 01/10/2007.

HARTER, S.; HERT, C.. Evaluation of Information Retrieval Systems: Approaches, Issues, and Methods. **Annual Review of Information Science and Technology (ARIST)**, v. 32, Medford: Martha, 1997.

HOUAISS, A. **Dicionário eletrônico Houaiss da língua portuguesa.** Rio de Janeiro: Objetiva. Versão eletrônica. 2001.

IDC. **The Expanding Digital Universe: A Forecast of Worldwide Information Growth Through 2010**, IDC, March, 2007. Disponível em: http://www.emc.com/collateral/analyst-reports/expanding-digital-idc-white-paper.pdf

INFO, Dossiê Computadores Falantes: No tempo em que as máquinas falavam. **Revista Exame Informática**, v. 128, p.82-86, Fev., 2006.

IRVIN, K.K.. **Comparing Information Retrieval Effectiveness of Different Metadata. Generation Methods**. A Master's paper for the M.S. in I.S. degree. April, 2003.

JANSSENS, F.. **Clustering of scientific fields by integrating text Mining and bibliometrics**. Katholieke Universiteit Leuven: Faculteit Ingenieurswetenschappen. Mei, 2007.

KURAMOTO, H.. Sintagmas Nominais: uma nova proposta para a Recuperação da Informação. **DataGramaZero**, v. 3, n. 1, fev. 2002.

_____. Uma abordagem alternativa para o tratamento e a recuperação da informação textual: os sintagmas nominais. **Ciência da Informação**, Brasília, v. 25, n. 2, maio/ago, p. 182-192, 1996.

KWASNIK, B.H.. The role of classification in knowledge representation and Discovery. **Library Trends**, v. 48, n. 1, Summer, p. 22-47, 1999.

LAKOFF, G.. **Women, fire, and dangerous things**: what categories reveal about the mind. Chicago: The University of Chicago Press, 1987.

LAWRENCE, S.; GILES, C.. Accessibility of Information on the Web. **Nature**, p.107-109, n. 400, 1999.

LEVY, P. **A cibercultura**. Rio de Janeiro: Ed. 34, 1999.

LIBERATO, Y.G.. **A estrutura do SN em português.** Tese de Doutorado em Letras. Belo Horizonte: UFMG, 203p. 1997.

LOURENÇO, C.A.. **Análise do Padrão Brasileiro de Metadados de Teses e Dissertações segundo o Modelo Entidade-Relacionamento.** Tese de Doutorado. Orientadora Profª. Lidia Alvarenga. ECI: UFMG, 2005.

LOPES, E.. **Fundamentos de lingüística contemporânea.** Editora Cultrix, 1999.

MARTINS, R. T.; HASEGAWA, R.; NUNES, M.G.V.. Curupira: um parser functional para o português. **NILC-TR-02-26**, Dez., 2002.

MCCALLUM, A. K.; et al.. **Automating de Construction of Internet Portals with Machine Learning Information Retrieval**, v.2, n. 3, p. 127-163, 2000.

MEADOW, C.T.; BOYCE, B.R.; KRAFT, D.H. **Text Information Retrieval Systems.** San Diego: Academic Press, 2000. 364 p.

MELNIK, S., GARCIA-MOLINA, H., AND RAHM, E. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: **ICDE**, pages 117–128. 2002.

MILSTEAD, J.; FELDMAN, S. Metadata: cataloging by any other name. **On Line Magazine**, New Haven, v.23, n.1, 1999.

MIORELLI, S.T. **ED-CER:. Extração do Sintagma Nominal. em Sentenças em Português.** Dissertação de mestrado Ciência da Computação. Porto Alegre: PUC. 2001.

MUELLER, S.P.M; PASSOS, E.J.L. As questões da comunicação científica e a ciência da informação. **Comunicação cientifica**, Brasília: Dep. de Ciência da Informação Universidade de Brasília, 2000.

NIERMAN A.; JAGADISH H.V.. ProTDB: Probabilistic data in XML. In: **Very Large Data Bases (VLDB) Conference**, Hong Kong, China, August 2002.

OTHERO, G. A. **Grammar Play: um parser sintático em Prolog para a língua portuguesa.** Dissertação de Mestrado. Porto Alegre: PUCRS, 2004.

PERINI M. A.; et al.. O Sintagma Nominal em Português: Estrutura, Significado e Função, **Revista de Estudos da Linguagem**. n. esp.. 1996.

POLANCO, X.; FRANÇOIS, C. Data clustering and cluster mapping or visualization in text processing and mining. In: International isko conference, 6., 2000, Toronto. **Proceedings…**Toronto: Ergon Verlag: Würzburg, 2000. p. 359-365.

POMBO, O.. Da Classificação dos Seres à Classificação dos Saberes, **Leituras**. Revista da Biblioteca Nacional de Lisboa, n. 2, Primavera, p. 19-33, dez., 2003. disponível no site: http://www.educ.fc.ul.pt/docentes/opombo/investigacao/opombo-classificacao.pdf Acesso em 05/12/2003.

RAMSDEN, M. J.. **An introduction to index language construction, a programed text**. London: C. Bingley, 1974.

SALTON, G.. **Automatic information organization and retrieval**. New York: McGraw-Hill, 1968.

SALTON, G e MCGILL, M. J. **Introduction to modern information retrieval.** NewYork: McGraw-Hill, 1983.

SARACEVIC, T.. Ciência da Informação: origem, evolução e relações. **PCI**, v. 1, n. 1, p. 41-62, jan/jun.1996.

SARACEVIC, T.. Relevance: a review of the literature and a framework for thinking on the notion in information science. part ii: nature and manifestations of relevance. **Journal of the american society for information science and technology**, v. 58, n. 13, p.1915–1933, 2007.

SATIJA, M.P.. Library classification:an essay in terminology. **Knowledge organization**, v. 27, n. 4, p. 221-229, 2000.

SHERA, J.H.; CLEVELAND, D.B.. History and foundations of Information Science. In: **Annual review of information science and technology**, v.12, 1977.

SILVA, M. C. P. S. e KOCH, I. V.. **Lingüística aplicada ao português: Sintaxe.** 14 ed. São Paulo: Cortez, 2007.

SILVA, F. S.. Personalização de conteúdo na TVDI através de um sistema de recomendação personalizada de programas de TV (SRPTV). **Anais...** III Fórum de Oportunidades em Televisão Digital Interativa, Poços de Caldas, 2005.

SIMEÃO, E.L.M.S.. **Comunicação extensiva e o formato do periódico científico em rede.** Brasília, 2003.

SKIENA, S.S.. **The algorithm design manual.** New York: Springer-Verlag, p. 317-318, 1997.

SOUZA, J.S.. **Classificação**: sistemas de classificação bibliográfica. 2.ed. São Paulo: Departamento Municipal de Cultura, 1950.

SOUZA, R.R.. **Uma proposta de metodologia para escolha automática de descritores utilizando sintagmas nominais.** Tese de Doutorado. Orientadora Profª. Lidia Alvarenga. UFMG, ECI, 2005.

SVENONIOUS, E.. **Classification theory**. March, 1985. 19p

TRASK, R. L.. **Dicionário de linguagem e lingüística,** São Paulo: Contexto, 2004.

TUFANO, D.. **Estudos de língua portuguesa : gramática.** 2 ed. São Paulo: Moderna, 1990.

TURING, A. M.. **Computing machinery and intelligence.** [Journal (Paginated)], 1950.

VILLAÇA, N.. **Impresso ou eletrônico? Um trajeto de leitura.** Rio de Janeiro: Mauad, 2002.

ANEXOS

I – Comparativo: Extração de SN Ogma x VISL

RESUMO 1:

SOUZA, Terezinha Batista, CATARINO, Maria Elisabete, SANTOS, Paulo Cesar dos. Metadados: catalogando dados na Internet. Transinformacao, Campinas, v. 9, n. 2, p. 93-105, maio/ago. 1997

TEXTO:
Apresenta de forma introdutória questões e conceitos fundamentais sobre metadados e a estruturação da descrição padronizada de documentos eletrônicos. Discorre sobre os elementos propostos no Dublin Core e comenta os projetos de catalogação dos recursos da Internet, CATRIONA, InterCat e CALCO.

Sintagmas nominais máximos extraídos automaticamente	
OGMA	**VISL**
1. forma introdutória 2. introdutória questões e conceitos fundamentais sobre metadados e a estruturação da descrição padronizada de documentos eletrônicos 3. os elementos propostos no dublin 4. os projetos de catalogação dos recursos da internet 5. catriona 6. intercat	1. introdutórias questões 2. conceitos fundamentais sobre metadados 3. a estruturação da descrição padronizada de documentos eletrônicos 4. os projetos de catalogação dos recursos da internet

RESUMO 2:

CUNHA, Murilo Bastos da. Biblioteca digital: bibliografia internacional anotada. Ciência da Informação, Brasília, v.26, n.2, p.195-213, maio/ago. 1997

TEXTO:
Bibliografia internacional seletiva e anotada sobre bibliotecas digitais. Aborda os seguintes aspectos: a) Visionários, principais autores que escreveram sobre a biblioteca do futuro, no período de 1945-1985; b) conceituação de biblioteca digital; c) projetos em andamento
na Alemanha, Austrália, Brasil, Canadá, Dinamarca, Espanha, Estados Unidos, Franca, Holanda, Japão, Nova Zelândia, Reino Unido, Suécia e Vaticano; d) aspectos técnicos relativos a construção de uma biblioteca digital: arquitetura do sistema, conversão de dados e escaneamento, marcação de textos, desenvolvimento de coleções, catalogação, classificação/indexação, metadados, referencia, recuperação da informação, direitos autorais e preservação da informação digital; e) principais fontes de reuniões técnicas especificas, lista de discussão, grupos e centros de estudos, cursos e treinamento.

Sintagmas nominais máximos extraídos automaticamente	
OGMA	**VISL**
1. bibliografia internacional seletiva e anotada sobre bibliotecas digitais 2. os seguintes aspectos 3. principais autores 4. o período de 1945-1985 5. a biblioteca do futuro 6. conceituação de biblioteca digital 7. projetos em andamento na alemanha 8. austrália 9. brasil 10. canadá 11. dinamarca 12. espanha , estados unidos , franca 13. holanda 14. japão , nova 15. zelândia 16. reino unido 17. suécia e vaticano 18. aspectos técnicos relativos 19. a construção de uma biblioteca digital 20. arquitetura do sistema 21. marcação de textos 22. desenvolvimento de coleções 23. catalogação 24. classificação 25. indexação 26. metadados 27. recuperação da informação , direitos autorais e preservação da informação digital 28. conversão de dados e escaneamento 29. principais fontes de reuniões técnicas 30. grupos e centros de estudos 31. cursos e treinamento 32. lista de discussão	1. bibliografia internacional seletiva e anotada sobre bibliotecas digitais 2. os seguintes aspectos 3. a biblioteca do futuro 4. o período de 1945-1985 5. conceituação de biblioteca digital 6. projetos em andamento na Alemanha 7. aspectos técnicos relativos 8. a construção de uma biblioteca digital 9. arquitetura do sistema 10. conversão de dados e escaneamento 11. marcação de textos 12. desenvolvimento de coleções 13. indexação 14. recuperação da informação 15. a informação digital 16. principais fontes de reuniões técnicas específicas 17. centros de estudos

RESUMO 3:

FAGUNDES, Maria Lucia Figueiredo, PRADO, Gilberto dos Santos. Videoteca digital : a experiência da videoteca multimeios do IA/UNICAMP. Transinformacao, Campinas, v.11, n.3, p. 293-299, set./dez. 1999

TEXTO:
Apresenta a implantação de recursos multimídia e interface Web no banco de dados desenvolvido para a coleção de vídeos da Videoteca Multimeios, pertencente ao Departamento de Multimeios do Instituto de Artes da UNICAMP. Localiza a discussão conceitual no universo das bibliotecas digitais e propõe alterações na configuração atual de seu banco de dados.

Sintagmas nominais máximos extraídos automaticamente	
OGMA	**VISL**
1. pertencente 2. o instituto de artes da unicamp 3. a o departamento de multimeios do instituto de artes da unicamp 4. a implantação de recursos multimídia e interface web no banco de dados desenvolvido para a coleção de vídeos da videoteca multimeios 5. a discussão conceitual no universo das bibliotecas digitais 6. alterações na configuração atual de seu banco de dados	1. a implantação de recursos multimídia 2. interface web 3. o banco de dados 4. coleção de vídeos da videoteca multimeios 5. o departamento de multimeios do instituto de artes da unicamp 6. discussão conceitual no universo das bibliotecas digitais 7. a configuração atual de seu banco de dados

RESUMO 4:

FAGUNDES, Maria Lucia Figueiredo, PRADO, Gilberto dos Santos. Videoteca digital : a experiência da videoteca multimeios do IA/UNICAMP. Transinformacao, Campinas, v.11, n.3, p. 293-299, set./dez. 1999

TEXTO:
Este artigo aborda a necessidade de adoção de padrões de descrição de recursos de informação eletrônica, particularmente, no âmbito da Embrapa Informática Agropecuária. O Rural Mídia foi desenvolvido utilizando o
modelo Dublin Core (DC) para descrição de seu acervo, acrescido de pequenas adaptações introduzidas diante da necessidade de adequar-se a especificidades meramente institucionais. Este modelo de metadados baseado no Dublin Core, adaptado para o Banco de Imagem, possui características que endossam a sua adoção, como a simplicidade na descrição dos recursos, entendimento semântico universal (dos elementos), escopo internacional e extensibilidade (o que permite sua adaptação as necessidades adicionais de descrição).

Sintagmas nominais máximos extraídos automaticamente	
OGMA	**VISL**
1. este artigo 2. a necessidade de adoção de padrões de descrição de recursos de informação eletrônica 3. particularmente 4. o âmbito da embrapa 5. informática agropecuária 6. o rural mídia 7. dublin 8. o modelo 9. descrição de seu acervo , acrescido de pequenas adaptações introduzidas diante de a necessidade 10. a ele a especificidades 11. meramente institucionais 12. a sua adoção 13. entendimento semântico universal 14. este modelo de metadados baseado no dublin 15. o banco de imagem 16. como a simplicidade na descrição dos recursos 17. dos elementos 18. escopo internacional e extensibilidade 19. sua adaptação 20. as necessidades adicionais de descrição	1. este artigo 2. a necessidade de adoção de padrões de descrição de recursos de 3. informação eletrônica 4. o âmbito da embrapa informática agropecuária 5. o rural mídia 6. o modelo dublin core 7. a descrição de seu acervo 8. seu acervo 9. pequenas adaptações introduzidas diante da necessidade de adequar-se a especificidades meramente institucionais 10. este modelo de metadados 11. o dublin core 12. o banco de imagem 13. características que endossam a sua adoção, como a simplicidade na 14. descrição dos recursos, entendimento semântico universal (dos elementos) 15. escopo internacional 16. o que permite sua adaptação às necessidades adicionais de descrição

RESUMO 5:

CHATAIGNIER, Maria Cecilia Pragana, SILVA, Margareth Prevot. Biblioteca digital: a experiência do Impa. Ciência da Informação, Brasília, v.30, n.3, p.7-12, set./dez. 2001

TEXTO:
Relato da experiência do Impa na informatização de sua biblioteca, utilizando o software Horizon, e na construção de um servidor de preprints (dissertações de mestrado, teses de doutorado e artigos ainda não publicados) através da participação no projeto internacional Math-Net.

Sintagmas nominais máximos extraídos automaticamente	
OGMA	**VISL**
1. a experiência do impa na informatização de sua biblioteca 2. o software 3. horizon 4. a construção de um servidor de preprints 5. relato da experiência do impa na informatização de sua biblioteca 6. dissertações de mestrado 7. teses 8. artigos aindanão publicados 9. internacional math-net 10. através da participação no projeto 11. a participação no projeto	1. a experiência 2. informatização de sua biblioteca 3. o software horizon 4. a construção de um servidor preprints 5. dissertações de mestrado, teses de doutorado e artigos ainda não publicados através da participação no projeto internacional math-net 6. através da participação no projeto internacional math-net

RESUMO 6:

MARCONDES, Carlos Henrique, SAYAO, Luis Fernando. Integração e interoperabilidade no acesso a recursos informacionais eletrônicos em C&T: a proposta da Biblioteca Digital Brasileira.. Ciência da Informação, Brasília, v.30, n.3, p.24-33, set./dez. 2001.

Descreve as opções tecnológicas e metodológicas para atingir a interoperabilidade no acesso a recursos informacionais eletrônicos, disponíveis na Internet, no âmbito do projeto da Biblioteca Digital Brasileira em Ciência e Tecnologia, desenvolvido pelo Instituto Brasileiro de Informação em Ciência e Tecnologia(IBCT). Destaca o impacto da Internet sobre as formas de publicação e comunicação em C&T e sobre os sistemas de informação e bibliotecas. São explicitados os objetivos do projeto da BDB de fomentar mecanismos de publicação pela comunidade brasileira de C&T, de textos completos diretamente na Internet, sob a forma de teses, artigos de periódicos, trabalhos em congressos, literatura "cinzenta",ampliando sua visibilidade e acessibilidade nacional e internacional, e também de possibilitar a interoperabilidade entre estes recursos informacionais brasileiros em C&T, heterogêneos e distribuídos, através de acesso unificado via um portal, sem a necessidade de o usuário navegar e consultar cada recurso individualmente.

Sintagmas nominais máximos extraídos automaticamente	
OGMA	**VISL**
1. as opções tecnológicas e metodológicas 2. a recursos 3. informacionais eletrônicos , disponíveis na internet 4. a biblioteca digital brasileira em ciência e tecnologia , desenvolvido 5. instituto brasileiro de informação em ciência e tecnologia 6. a interoperabilidade no acesso 7. o âmbito do projeto da biblioteca digital brasileira em ciência e tecnologia , desenvolvido 8. ibct 9. a internet 10. publicação e comunicação em c&t 11. os sistemas de informação e bibliotecas 12. o impacto da internet 13. a bdb 14. mecanismos de publicação por a comunidade brasileira de c&t 15. textos completos 16. a internet 17. artigos 18. trabalhos em congressos 19. literatura cinzenta 20. sua visibilidade e acessibilidade nacional 21. a interoperabilidade entre estes recursos 22. informacionais brasileiros em c&t , heterogêneos 23. a necessidade 24. cada recurso 25. a forma de teses 26. através de acesso unificado 27. explicitados os objetivos do projeto da bdb	1. as opções tecnológicas e metodológicas 2. a interoperabilidade no acesso a recursos informacionais eletrônicos 3. a internet 4. o âmbito do projeto da biblioteca digital brasileira 5. o projeto da biblioteca digital brasileira 6. a biblioteca digital brasileira 7. o ibict 8. o impacto da internet 9. as formas de publicação e comunicação 10. os sistemas de informação 11. o projeto da biblioteca digital brasileira 12. mecanismos de publicação 13. a sociedade brasileira de ciência e tecnologia 14. textos completos 15. a internet 16. a forma de teses 17. artigos de periódicos 18. trabalhos em congressos 19. literatura cinzenta 20. acessibilidade nacional e internacional 21. a interoperabilidade entre estes recursos informacionais brasileiros em c&t 22. através de acesso unificado 23. um portal 24. a necessidade do usuário 25. cada recurso

II – Enumeração dos artigos que formam o corpus ENANCIB 2005.

GT 1: Estudos Históricos e Epistemológicos da Informação

NÚMERO	TÍTULO	AUTOR
1	POR UMA CIÊNCIA FORMATIVA E INDICIÁRIA: PROPOSTA EPISTEMOLÓGICA PARA A CIÊNCIA DA INFORMAÇÃO	Eliany Alvarenga de Araújo
2	INFORMAÇÃO, MEMÓRIA E HISTÓRIA: A INSTITUIÇÃO DE UM SISTEMA DE INFORMAÇÃO NA CORTE DO RIO DE JANEIRO	Icléia Thiesen
3	FACES DA PESQUISA E DA INTERDISCIPLINARIDADE EM CIÊNCIA DA INFORMAÇÃO NO BRASIL	Renato José da Silva
4	QUESTÕES SOBRE O LOCUS ACADÊMICO-INSTITUCIONAL DA ARQUIVOLOGIA NA CIÊNCIA DA INFORMAÇÃO	Georgete Medleg Rodrigues; Angelica Alves da Cunha Marques
5	EVOLUÇÃO E TENDÊNCIAS DA CIÊNCIA DA INFORMAÇÃO, NO EXTERIOR E NO BRASIL: QUADRO COMPARATIVO A PARTIR DE PESQUISAS HISTÓRICAS E EMPÍRICAS	Lena Vania Ribeiro Pinheiro
6	AS METÁFORAS DA IDENTIDADE NAS REDES DO CONHECIMENTO EM TEMPO DE COMUNICAÇÃO GLOBALIZADA	Evelyn Goyannes Dill Orrico; Carmen Irene Correia de Oliveira
7	V ENANCIB: ANÁLISE DOS CAMINHOS DE PESQUISAS	Marlene de Oliveira; Maria Aparecida Lourenço Santana
8	LYDIA DE QUEIROZ SAMBAQUY E A CIÊNCIA DA INFORMAÇÃO NO BRASIL	Nanci Oddone
9	MEDIAÇÕES NOS ESTUDOS DE INFORMAÇÃO E COMUNICAÇÃO	Solange Puntel Mostafa
10	CONTRIBUTO PARA ENTENDER A CIÊNCIA DA INFORMAÇÃO	Marcia H. T. de Figueredo Lima

GT 2: Organização do conhecimento e representação da informação

NÚMERO	TÍTULO	AUTOR
1	ESTUDO QUALITATIVO-DESCRITIVO PARA IDENTIFICAÇÃO DE FATORES CONDICIONANTES DA PRESERVAÇÃO DIGITAL	Katia P. Thomaz
2	ORGANIZAÇÃO E REPRESENTAÇÃO DE ÁREAS DO CONHECIMENTO EM CIÊNCIA E TECNOLOGIA: PRINCÍPIOS DE AGREGAÇÃO EM GRANDES ÁREAS SEGUNDO DIFERENTES CONTEXTOS DE PRODUÇÃO E USO DE INFORMAÇÃO	Rosali Fernandez de Souza
3	A CONTRIBUIÇÃO DO MÉTODO DIPLOMÁTICO E DA INDEXAÇÃO SISTEMÁTICA DE KAISER PARA A ANÁLISE DOCUMENTAL DE CONTEÚDO DE EMENTAS JURÍDICAS: UMA EXPERIMENTAÇÃO COM PROJETOS LEGISLATIVOS	Rodrigo Rabello da Silva
4	PROPOSTA PARA UM ESQUEMA DE CLASSIFICAÇÃO DAS FONTES DE INFORMAÇÃO PARA NEGÓCIO	Antonio Braz de Oliveira e Silva; Marcus José de Oliveira Campos
5	UMA PROPOSTA DE METODOLOGIA PARA INDEXAÇÃO AUTOMÁTICA UTILIZANDO SINTAGMAS NOMINAIS	Renato Rocha Souza
6	WEB SEMÂNTICA: ASPECTOS INTERDISCIPLINARES PARA A ORGANIZAÇÃO E RECUPERAÇÃO DE INFORMAÇÕES	Rogério Aparecido Sá Ramalho; Silvana Aparecida Borsetti Gregorio Vidotti; Mariângela Spotti Lopes Fujita
7	A EFICÁCIA PROBATÓRIA DO DOCUMENTO COMO SUBSÍDIO À ORGANIZAÇÃO DA INFORMAÇÃO JURÍDICO-DIGITAL: UMA REFLEXÃO ACERCA DOS AVANÇOS TEÓRICOS DA DIPLOMÁTICA	Lúcia Maria Barbosa do Nascimento; José Augusto Chaves Guimarães
8	O ARRANJO ARQUIVÍSTICO COMO ESCRITA: UMA REFLEXÃO SOBRE A NARRATIVA EM IMAGENS A PARTIR DO FUNDO PEDRO MIRANDA NO ARQUIVO PÚBLICO E HISTÓRICO DE RIBEIRÃO PRETO	Eduardo Ismael Murguía; Tânia Cristina Registro
9	REPRESENTAÇÃO INFORMACIONAL E AS TEMÁTICAS NACIONAIS: DESAFIOS E TENDÊNCIAS PARA A ELABORAÇÃO DE LINGUAGENS DE INDEXAÇÃO	Maria Aparecida Moura
10	MODELO HIPERTEXTUAL - MHTX: UM MODELO PARA ORGANIZAÇÃO HIPERTEXTUAL DE DOCUMENTOS	Gercina Ângela Borém Oliveira Lima

GT3: Mediação, circulação e uso da informação

NÚMERO	TÍTULO	AUTOR
1	COMPORTAMENTO DE BUSCA E USO DE INFORMAÇÃO DE PESQUISADORES DAS ÁREAS DE BIOLOGIA MOLECULAR E BIOTECNOLOGIA	Isabel Merlo Crespo; Sônia Elisa Caregnato
2	TECNOLOGIAS DA INTELIGÊNCIA: USO DE SOFTWARES NA (IN)FORMAÇÃO DE SUJEITOS-APRENDENTES	Mirian de Albuquerque Aquino; Geórgia Geogletti Cordeiro Dantas
3	MEDIAÇÃO E CIRCULAÇÃO DA INFORMAÇÃO: O JOGO DISCURSIVO NA ARTE CONCEITUAL	Priscilla Arigoni Coelho; Evelyn Goyannes Dill Orrico
4	JANELAS DA CULTURA LOCAL: ABRINDO OPORTUNIDADES PARA INCLUSÃO DIGITAL	Isa Maria Freire
5	GERAÇÃO, MEDIAÇÃO E USO DE INFORMAÇÃO: UMA PROPOSTA DE MODELO TEÓRICO	Eliany Alvarenga de Araújo
6	USO DE PERIÓDIOCOS ELETRÔNICOS: UM ESTUDO SOBRE O PORTAL PERIÓDICOS CAPES NA UFMG	Luiz Cláudio Gomes Maia; Beatriz Valadares Cendon
7	UM ESTUDO SOBRE ACESSO À INFORMAÇÃO EM BIBLIOTECAS VIRTUAIS PARA A PESQUISA CIENTÍFICA	Sandra Lúcia Rebel Gomes
8	O USO DE FONTES DE INFORMAÇÃO POR EXECUTIVOS DO SETOR DE TECNOLOGIA DA INFORMAÇÃO	Jaime Sadao Yamassaki Bastos; Ricardo Rodrigues Barbosa
9	POLÍTICAS PÚBLICAS PARA O LIVRO E A LEITURA E SUA INFLUÊNCIA NA INDÚSTRIA EDITORIAL DE SALVADOR	Susane Santos Barros; Jussara Borges; Othon Jambeiro
10	O PROFISSIONAL DA INFORMAÇÃO E A MEDIAÇÃO DO ACESSO À INTERNET NA BIBLIOTECA UNIVERSITÁRIA.	Flávia Ferreira; Jussara Borges; Othon Jambeiro

GT 4: Gestão de Unidades de Informação

NÚMERO	TÍTULO	AUTOR
1	GESTÃO DA INFORMAÇÃO E DO CONHECIMENTO EM ORGANIZAÇÕES BRASILEIRAS: PROPOSTA DE MAPEAMENTO CONCEITUAL INTEGRATIVO	Rivadávia Correa Drummond de Alvarenga Neto
2	TOMADA DE DECISÃO GERENCIAL SOBRE A OFERTA DE PRODUTOS E SERVIÇOS DAS BIBLIOTECAS DE BRASÍLIA NA WEB	Sueli Angélica do Amaral
3	INFORMAÇÃO E APRENDIZAGEM ORGANIZACIONAL: ESTUDO DE CASO EM UM ÓRGÃO PÚBLICO MUNICIPAL	Elaine Silva Frois
4	ARQUITETURA DA INFORMAÇÃO PARA BIBLIOTECA DIGITAL PERSONALIZÁVEL	Liriane Soares de Araújo de Camargo
5	ANÁLISE DAS FUNCIONALIDADES DE INTRANETS E PORTAIS: PESQUISA EXPLORATÓRIA EM MÉDIAS E GRANDES ORGANIZAÇÕES BRASILEIRAS	Rodrigo Baroni de Carvalho
6	PERFIL DO CLIENTE INTERNO DO NÚCLEO DE INFORMAÇÃO BIOTECNOLÓGICA	Célia Regina Simonetti Barbalho
7	MODELO CONCEITUAL DE INTELIGÊNCIA ORGANIZACIONAL APLICADA À FUNÇÃO MANUTENÇÃO	Robson de Paula Alves; Orandi Mina Falsarella
8	USO DE FONTES DE INFORMAÇÃO PARA A INTELIGÊNCIA COMPETITIVA: UM ESTUDO DA INFLUÊNCIA DO PORTE DAS EMPRESAS SOBRE O COMPORTAMENTO INFORMACIONAL	Ricardo Rodrigues Barbosa
9	A BIBLIOTECA UNIVERSITÁRIA COMO ORGANIZAÇÃO DO CONHECIMENTO: DO MODELO CONCEITUAL ÀS PRÁTICAS	Emeide Nóbrega Duarte; Alzira Karla Araújo da Silva; Edilene Galdino dos Santos; Izabel França de Lima; Marcos Paulo F. Rodrigues; Suzana Queiroga da Costa
10	REPOSITÓRIOS INSTITUCIONAIS E A GESTÃO DO CONHECIMENTO CIENTÍFICO	Fernando César Lima Leite; Sely Maria de Souza Costa

GT5: Política Ética e Economia da Informação

NÚMERO	TÍTULO	AUTOR
1	A ECONOMIA DA INFORMAÇÃO NO BRASIL: DIMENSIONAMENTO E ESPACIALIZAÇÃO ATRAVÉS DAS OCUPAÇÕES DO CENSO DEMOGRÁFICO 2000	Marcos Franco Bueno; Paulo de Martino Jannuzzi
2	A PRESERVAÇÃO DA INFORMAÇÃO ARQUIVÍSTICA E A FORMULAÇÃO DE POLÍTICAS PÚBLICAS. A PRESERVAÇÃO DA INFORMAÇÃO ARQUIVÍSTICA E A FORMULAÇÃO DE POLÍTICAS PÚBLICAS.	Sérgio Conde de Albite Silva
3	GOVERNO ELETRÔNICO – CONCEITOS E DEBATES: INSTRUMENTO DE CAPILARIDADE DA RELAÇÃO ESTADO E SOCIEDADE NA AMÉRICA LATINA	José França Neto; Carlos Alberto Antão Siqueira; Luciana Silva Custódio
4	TÃO LONGE, TÃO PERTO: ALCANCES E LIMITAÇÕES DO E-GOV NO ÂMBITO DA GESTÃO MUNICIPAL	Adriana Mendes de Araújo; Leandra Gonçalves; Ana Maria Pereira Cardoso; Juliana do Couto Bemfica
5	INFORMAÇÃO, CONHECIMENTO E DESENVOLVIMENTO	Sarita Albagli
6	INFORMAÇÃO, ASSIMETRIA DE INFORMAÇÕES E REGULAÇÃO DO MERCADO DE SAÚDE SUPLEMENTAR.	Clóvis Ricardo Montenegro de Lima
7	A LÓGICA ECONÔMICA DA EDIÇÃO CIENTÍFICA CERTIFICADA	César Bolaño; Nair Kobashi; Raimundo Santos
8	INCLUSÃO INFORMACIONAL: ESTUDO COM INDIVÍDUOS QUE PASSARAM POR PROGRAMAS DE INCLUSÃO DIGITAL EM SALVADOR-BAHIA	Jussara Borges; Helena Pereira da Silva
9	PENSANDO AS ESTATÍSTICAS PÚBLICAS SOBRE CARREIRAS EDUCACIONAIS NA ÁREA DE CIÊNCIA E TECNOLOGIA, POR GÊNERO	Zuleica Lopes Cavalcanti de Oliveira
10	INCLUSÃO INFORMACIONAL NA PERSPECTIVA DOS TELECENTROS	Mauro Araújo Câmara

III – Lista de *stopwords* utilizada pelo OGMA

a
à
a!
a!
a!
a+fim+de
a+qual
a+respeito+de
abaixo
abaixo
abaixo+de
absolutamente
acaso
acerca+de
acima
acima+de
adiante
adiante
agora
ah
ai
ainda
além
além+de
algo
alguém
algum
ali
alve
amanhã
amém
ânimo
ante
antes
antes+de
ao
ao!
ao+invés+de
ao+redor+de
aos
aos!
apessar+de
apoiado
após
aquela
aquelas
aquele
aqueles
aqui
aqui
aquilo
arre
as
as!
as!
às!
as+quais
assaz
assim
assim
até
até+a
atrás
basta
bastante
bastante
bem
bem+como
bilhão
bilionésimo
bis
boa
bravo
c
Cá
Cada
Caramba
Catorze
Cc
ccc
cd
cedo
cem
centésimo
cêntuplo
certamente
certo
chega
ci
cibtudo
cinco
cinquenta
cinqüenta
cm
com
comigo
como+também
como+um
conforme
conosco
consigo
contigo
contra
convosco
coragem
credo
cruzes
cuja
cujas
cujo
cujos
d
da!
daquilo
das!
dc
dcc
dccc
de
de+acordo+com
debaixo+de
décimo
décuplo
defronte+de
demais
dentreo+de
dentro
depois
depois+de
depressa
desde
devagar
deveras
dez
dezenove
dezoito
dezesseis
dezessete
diante+de
do
do!
dobro
dois
dos!
doze
ducentésimo
dum!
duma!
dumas!
duns!
duodécimo
duodécuplo
dúplice
duplo
duramente
durante
duzentos
e
e+não
eh
ei
eia
ela
elas
ele
eles
em
em+cima+de
em+face+de
em+frente+a
em+frente+de
em+lugar+de
em+vez+de
entre
entretanto
essa
essas
esse
esses
esta
estas
este
estes
eu

facilmente
firme
fora
fora
fora de
força
francamente
hein
hem
hum
i
ih
ii
iii
irra
isso
isto
iv
ix
já
já
jamais
junto a
junto de
l
lá
lhe
lhes
livra
logo
logo
longe
lx
lxx
lxxx
mais
mal
mas
mas+ainda
mas+também
me
mediante
meio
meio
melhor
menos
mesmo
metade
meu
meus
mil
milésimo
milésimos
milhão
milionésimo
mim
minha
minhas
muito
muito
na
na!
nada
nada
não
não+obstane
não+obstante
não+sí
não+só
naquilo
nas!
nem
nenhum
ninguém
no!
no+caso+de
no+entanto
nonagésimo
nongentésimo
nono
nônuplo
nós
nos!
nossa
nossa
nossas
nosso
nossos
nove
novecentos
noventa
num!
numa!
numas!
nunca
nuns!
o
ó
ô
o!
o!
o!
o+qual
o+que
oba
octogésimo
óctuplo
oh
oitavo
oitenta
oito
oitocentos
olá
onde
onze
ora
ora
os
os
os!
os!
os+quais
ou
outrem
outro
outrora
oxalá
para
para+com
pela!
pelas!
pelo!
pelos!
perante
perdão
perto
perto+de
pior
pois
por
por+conseguinte
por+isso
por+trás+de
porém
porquanto
porque
portanto
porventura
pouco
pra
primeiro
próprio
psiu
puxa
quadragésimo
quadringentésimo
quádruplo
qualquer
quantas
quanto
quantos
quarenta
quarto
quase
quaternário
quatorze
quatro
quatrocentos
que
que
quê
quem
quem!
quer
quiça
qüingentésimo
quinhentos
quinquagésimo
qüinquagésimo
quinto
quíntuplo
quinze
raios
rapidamente
realmente
safa
se
segundo
segundo
seis
seiscentésimo
seiscentos
seja
sem
sempre
septingentéssimo

septuagésimo
sessenta
sete
setecentos
setenta
sétimo
setuagésimo
sétuplo
seu
seus
sexagésimo
sexcentésimo
sexto
sêxtuplo
si
sim
sob
sobre
sua
suas
tal
talvez
tanto
tão
tarde
tchau
te
terceiro
terço
teu
teus
ti
toca
todavia
todo
tomara
trás
trecentésimo
três
treze
trezentos
trigésimo
trinta
tríplice
triplo
tu
tua
tuas
tudo
ufa
ui
um
um
um
uma
umas
undécimo
undécuplo
uns
v
vamos
vário
vi
vigésimo
viii
vinte
viva
vós
vossa
vossas
vosso
vossos
x
xc
xi
xii
xiii
xiv
xix
xl
xô
xv
xvi
xvii
xviii
xx
xxi
xxx

IV – Exemplos de utilização do OGMA

e – Etiquetar texto

exemplo: ogma e texto.txt textoetiquetado.txt

ENTRADA	SAÍDA
O novo cálculo das aposentadorias resulta em valores menores do que os atuais para quem perde o benefício com menos tempo de contribuição e idade.	o/AD novo/AJ cálculo/SU de/PR as/AD aposentadorias/SU resulta/VB em/PR valores/SU menores/AJSU que/CJPL os/ADPR atuais/VBAJSU para/PR quem/PL perde/VB o/AD benefício/SU com/PR menos/AV tempo/SU de/PR contribuição/SU e/CJ idade/SU ./PN

s - Extrai os Sintagmas Nominais e grava em um arquivo

exemplo: ogma s textoetiquetado.txt relacaosn.txt

ENTRADA	SAÍDA
o/AD novo/AJ cálculo/SU de/PR as/AD aposentadorias/SU resulta/VB em/PR valores/SU menores/AJSU que/CJPL os/ADPR atuais/VBAJSU para/PR quem/PL perde/VB o/AD benefício/SU com/PR menos/AV tempo/SU de/PR contribuição/SU e/CJ idade/SU ./PN	- o novo cálculo das aposentadorias - o benefício com menos tempo de contribuição e idade - valores menores que os atuais

x – Mostra os Sintagmas Nominais do arquivo

ex: ogma x texto.txt

ENTRADA	SAÍDA
O novo cálculo das aposentadorias resulta em valores menores do que os atuais para quem perde o benefício com menos tempo de contribuição e idade.	- o novo cálculo das aposentadorias - o benefício com menos tempo de contribuição e idade - valores menores que os atuais

tt - gera tabela de termos com numero de vezes que aparecem no texto

ex: ogma tt texto.txt tabtermos.txt

ENTRADA	SAÍDA
O novo cálculo das aposentadorias resulta em valores menores do que os atuais para quem perde o benefício com menos tempo de contribuição e idade.	novo/1/0,055556 cálculo/1/0,055556 das/1/0,055556 aposentadorias/1/0,055556 resulta/1/0,055556 valores/1/0,055556 menores/1/0,055556 que/1/0,055556 atuais/1/0,055556 para/1/0,055556 quem/1/0,055556 perde/1/0,055556 benefício/1/0,055556 com/1/0,055556 menos/1/0,055556 tempo/1/0,055556 contribuição/1/0,055556 idade/1/0,055556

tts - gera tabela de termos com número de vezes que aparecem no texto (filtra stopwords)

ex: ogma tts texto.txt tabtermos.txt

ENTRADA	SAÍDA
O novo cálculo das aposentadorias resulta em valores menores do que os atuais para quem perde o benefício com menos tempo de contribuição e idade.	novo/1/0,076923 cálculo/1/0,076923 das/1/0,076923 aposentadorias/1/0,076923 resulta/1/0,076923 valores/1/0,076923 menores/1/0,076923 atuais/1/0,076923 perde/1/0,076923 benefício/1/0,076923 tempo/1/0,076923 contribuição/1/0,076923 idade/1/0,076923

ts - gera tabela de sn etiquetados com n. de vezes que aparecem

ex: ogma ts relacaosn.txt tabsnf.txt

ENTRADA	SAÍDA
O novo cálculo das aposentadorias resulta em valores menores do que os atuais para quem perde o benefício com menos tempo de contribuição e idade.	o novo cálculo das aposentadorias/1/0,333333 o benefício com menos tempo de contribuição e idade/1/0,333333 valores menores que os atuais/1/0,333333

tc - gera tabela de SN etiquetados com número de vezes que aparecem em todo o texto

ex: ogma tc relacaosn.txt tabsnf.txt

ENTRADA	SAÍDA
O novo cálculo das aposentadorias resulta em valores menores do que os atuais para quem perde o benefício com menos tempo de contribuição e idade.	o novo cálculo das aposentadorias/1/0,333333 o benefício com menos tempo de contribuição e idade/1/0,333333 valores menores que os atuais/1/0,333333

tr - gera tabela de SN pontuados

ex: ogma tr relacaosn.txt tabsnf.txt

ENTRADA	SAÍDA
- o novo cálculo das aposentadorias - o benefício com menos tempo de contribuição e idade - valores menores que os atuais	o novo cálculo das aposentadorias/1/1,100000/2 o benefício com menos tempo de contribuição e idade/1/1,400000/3 valores menores que os atuais/1/1,100000/2

i - calcula a similaridade entre duas tabelas de termos

ex: ogma i tabela1.txt tabela2.txt

it - calcula a similaridade entre dois textos comparando por termos

ex: ogma i texto1.txt texto2.txt

ir - calcula a similaridade entre dois textos comparando com SN pontuados

ex: ogma i texto1.txt texto2.txt

ic - calcula a similaridade entre dois textos comparando com os SN

ex: ogma i texto1.txt texto2.txt

V – Resultados do programa WEKA – Classificação/Naive Bayes

TERMOS/ENANCIB05

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     tt-weka.filters.unsupervised.attribute.StringToNominal-
C1-weka.filters.unsupervised.attribute.Remove-R2-
weka.filters.unsupervised.attribute.Remove-R52
Instances:    50
Attributes:   51
Test mode:    10-fold cross-validation

Time taken to build model: 0 seconds

=== Summary ===

Correctly Classified Instances          25               50      %
Incorrectly Classified Instances        25               50      %
Kappa statistic                          0.375
Mean absolute error                      0.1975
Root mean squared error                  0.4327
Relative absolute error                 61.7335 %
Root relative squared error            108.1807 %
Total Number of Instances               50

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.7       0.1        0.636     0.7       0.667    1
  0.4       0.075      0.571     0.4       0.471    2
  0.3       0.1        0.429     0.3       0.353    3
  0.6       0.2        0.429     0.6       0.5      4
  0.5       0.15       0.455     0.5       0.476    5

=== Confusion Matrix ===

 a b c d e   <-- classified as
 7 0 2 1 0 | a = 1
 1 4 0 4 1 | b = 2
 2 0 3 1 4 | c = 3
 0 2 1 6 1 | d = 4
 1 1 1 2 5 | e = 5
```

TERMOS SEM STOPWORDS/ENANCIB05

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     tts-weka.filters.unsupervised.attribute.StringToNominal-
C1-weka.filters.unsupervised.attribute.Remove-R2,53
Instances:    50
Attributes:   51

Time taken to build model: 0.02 seconds

=== Summary ===

Correctly Classified Instances          26               52      %
Incorrectly Classified Instances        24               48      %
Kappa statistic                          0.4
Mean absolute error                      0.1964
Root mean squared error                  0.4346
Relative absolute error                 61.3699 %
Root relative squared error            108.6379 %
Total Number of Instances               50

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.4       0.15       0.4       0.4       0.4      1
  0.6       0.075      0.667     0.6       0.632    2
  0.6       0.25       0.375     0.6       0.462    3
  0.8       0.075      0.727     0.8       0.762    4
  0.2       0.05       0.5       0.2       0.286    5

=== Confusion Matrix ===

 a b c d e   <-- classified as
 4 1 4 1 0 | a = 1
 2 6 0 1 1 | b = 2
 2 1 6 0 1 | c = 3
 0 0 2 8 0 | d = 4
 2 1 4 1 2 | e = 5
```

SINTAGMAS NOMINAIS/ENANCIB05

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     tc-weka.filters.unsupervised.attribute.StringToNominal-
C1-weka.filters.unsupervised.attribute.Remove-R2
Instances:    50
Attributes:   52


Time taken to build model: 0.03 seconds

=== Summary ===

Correctly Classified Instances          12               24      %
Incorrectly Classified Instances        38               76      %
Kappa statistic                          0.05
Mean absolute error                      0.2984
Root mean squared error                  0.5301
Relative absolute error                 93.258  %
Root relative squared error            132.5289 %
Total Number of Instances               50

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.2       0.175      0.222     0.2       0.211    1
  0.1       0.2        0.111     0.1       0.105    2
  0.2       0.2        0.2       0.2       0.2      3
  0.3       0.15       0.333     0.3       0.316    4
  0.4       0.225      0.308     0.4       0.348    5

=== Confusion Matrix ===

 a b c d e   <-- classified as
 2 2 3 1 2 | a = 1
 4 1 0 1 4 | b = 2
 0 3 2 3 2 | c = 3
 1 1 4 3 1 | d = 4
 2 2 1 1 4 | e = 5
```

SINTAGMAS NOMINAIS PONTUADOS/ENANCIB05

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     tr-weka.filters.unsupervised.attribute.Remove-R53-
weka.filters.unsupervised.attribute.StringToNominal-C1-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    50
Attributes:   51

Time taken to build model: 0 seconds

=== Summary ===

Correctly Classified Instances          23               46      %
Incorrectly Classified Instances        27               54      %
Kappa statistic                          0.325
Mean absolute error                      0.2185
Root mean squared error                  0.4583
Relative absolute error                 68.2714 %
Root relative squared error            114.5864 %
Total Number of Instances               50

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.7       0.075      0.7       0.7       0.7      1
  0.5       0.075      0.625     0.5       0.556    2
  0.1       0.15       0.143     0.1       0.118    3
  0.4       0.125      0.444     0.4       0.421    4
  0.6       0.25       0.375     0.6       0.462    5

=== Confusion Matrix ===

 a b c d e   <-- classified as
 7 0 1 0 2 | a = 1
 1 5 1 0 3 | b = 2
 1 2 1 3 3 | c = 3
 1 0 3 4 2 | d = 4
 0 1 1 2 6 | e = 5
```

TERMOS/JORNAIS04

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     q-tt-
weka.filters.unsupervised.attribute.StringToNominal-C1-
weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    10-fold cross-validation
Time taken to build model: 0.16 seconds

=== Summary ===

Correctly Classified Instances         106               66.25   %
Incorrectly Classified Instances        54               33.75   %
Kappa statistic                          0.55
Mean absolute error                      0.1687
Root mean squared error                  0.4063
Relative absolute error                 44.9841 %
Root relative squared error             93.8305 %
Total Number of Instances              160

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.625     0.192      0.521     0.625     0.568    1
  0.625     0.075      0.735     0.625     0.676    2
  0.925     0.108      0.74      0.925     0.822    3
  0.475     0.075      0.679     0.475     0.559    4

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 25  4  5  6 |  a = 1
 11 25  1  3 |  b = 2
  3  0 37  0 |  c = 3
  9  5  7 19 |  d = 4
```

TERMOS SEM STOPWORDS/JORNAIS04

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     q-tts-
weka.filters.unsupervised.attribute.StringToNominal-C1-
weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    10-fold cross-validation

Time taken to build model: 0.03 seconds

=== Summary ===

Correctly Classified Instances         147               91.875  %
Incorrectly Classified Instances        13                8.125  %
Kappa statistic                          0.8917
Mean absolute error                      0.0404
Root mean squared error                  0.1978
Relative absolute error                 10.7801 %
Root relative squared error             45.6836 %
Total Number of Instances              160

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.85      0.033      0.895     0.85      0.872    1
  0.975     0.025      0.929     0.975     0.951    2
  0.975     0.025      0.929     0.975     0.951    3
  0.875     0.025      0.921     0.875     0.897    4

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 34  3  0  3 |  a = 1
  1 39  0  0 |  b = 2
  1  0 39  0 |  c = 3
  2  0  3 35 |  d = 4
```

SINTAGMAS NOMINAIS/JORNAIS04

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     q-tc-
weka.filters.unsupervised.attribute.StringToNominal-C1-
weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    10-fold cross-validation

Time taken to build model: 0.03 seconds

=== Summary ===

Correctly Classified Instances         147               91.875  %
Incorrectly Classified Instances        13                8.125  %
Kappa statistic                          0.8917
Mean absolute error                      0.0445
Root mean squared error                  0.2049
Relative absolute error                 11.8669 %
Root relative squared error             47.3202 %
Total Number of Instances              160

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.825     0.017      0.943     0.825     0.88     1
  0.925     0.017      0.949     0.925     0.937    2
  0.975     0.033      0.907     0.975     0.94     3
  0.95      0.042      0.884     0.95      0.916    4

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 33  2  1  4 |  a = 1
  0 37  2  1 |  b = 2
  1  0 39  0 |  c = 3
  1  0  1 38 |  d = 4
```

SINTAGMAS NOMINAIS PONTUADOS/JORNAIS04

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     q-tr-
weka.filters.unsupervised.attribute.StringToNominal-C1-
weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    10-fold cross-validation

Time taken to build model: 0.03 seconds

=== Summary ===

Correctly Classified Instances         132               82.5    %
Incorrectly Classified Instances        28               17.5    %
Kappa statistic                          0.7667
Mean absolute error                      0.0878
Root mean squared error                  0.2887
Relative absolute error                 23.4213 %
Root relative squared error             66.6644 %
Total Number of Instances              160

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.75      0.058      0.811     0.75      0.779    1
  0.75      0.05       0.833     0.75      0.789    2
  0.975     0.058      0.848     0.975     0.907    3
  0.825     0.067      0.805     0.825     0.815    4

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 30  5  0  5 |  a = 1
  4 30  3  3 |  b = 2
  1  0 39  0 |  c = 3
  2  1  4 33 |  d = 4
```

SINTAGMAS NOMINAIS ANINHADOS/JORNAIS04

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     q-trn-weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 1: Prior probability = 0.25
Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         136               85      %
Incorrectly Classified Instances        24               15      %
Kappa statistic                          0.8
Mean absolute error                      0.0726
Root mean squared error                  0.2657
Relative absolute error                 19.3732 %
Root relative squared error             61.3545 %
Total Number of Instances              160

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.85      0.117      0.708     0.85      0.773    1
  0.85      0.008      0.971     0.85      0.907    2
  0.9       0.042      0.878     0.9       0.889    3
  0.8       0.033      0.889     0.8       0.842    4

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 34  0  3  3 |  a = 1
  5 34  0  1 |  b = 2
  4  0 36  0 |  c = 3
  5  1  2 32 |  d = 4
```

SINTAGMAS NOMINAIS ANINHADOS PONTUADOS/JORNAIS04

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     q-tra-weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 1: Prior probability = 0.25
Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         137               85.625  %
Incorrectly Classified Instances        23               14.375  %
Kappa statistic                          0.8083
Mean absolute error                      0.073
Root mean squared error                  0.2628
Relative absolute error                 19.4653 %
Root relative squared error             60.6951 %
Total Number of Instances              160

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.825     0.083      0.767     0.825     0.795    1
  0.85      0.05       0.85      0.85      0.85     2
  0.9       0.017      0.947     0.9       0.923    3
  0.85      0.042      0.872     0.85      0.861    4

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 33  4  0  3 |  a = 1
  3 34  1  2 |  b = 2
  4  0 36  0 |  c = 3
  3  2  1 34 |  d = 4
```

VI – Resultados do programa WEKA – Agrupamento/SimpleKMeans

TERMOS/ENANCIB05

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 5 -S 10
Relation:     tt-weka.filters.unsupervised.attribute.StringToNominal-
C1-weka.filters.unsupervised.attribute.Remove-R2
Instances:    50
Attributes:   52

kMeans
======

Number of iterations: 4

Within cluster sum of squared errors: 34.420771215381976

Clustered Instances

0       6 ( 12%)
1      11 ( 22%)
2       7 ( 14%)
3      20 ( 40%)
4       6 ( 12%)


Class attribute: grupo
Classes to Clusters:

 0 1 2 3 4  <-- assigned to cluster
 0 1 0 7 2 | 1
 2 5 2 0 1 | 2
 2 1 0 7 0 | 3
 0 4 1 2 3 | 4
 2 0 4 4 0 | 5

Cluster 0 <-- 3
Cluster 1 <-- 2
Cluster 2 <-- 5
Cluster 3 <-- 1
Cluster 4 <-- 4

Incorrectly clustered instances :   29.0   58      %
```

TERMOS SEM STOPWORDS/ENANCIB05

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 5 -S 10
Relation:     tts-weka.filters.unsupervised.attribute.StringToNominal-
C1-weka.filters.unsupervised.attribute.Remove-R2,53
Instances:    50
Attributes:   51


Test mode:    Classes to clusters evaluation on training data
=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 5
Within cluster sum of squared errors: 36.5906257742584


Clustered Instances

0      11 ( 22%)
1      14 ( 28%)
2       9 ( 18%)
3      12 ( 24%)
4       4 (  8%)


Class attribute: grupo
Classes to Clusters:

 0 1 2 3 4  <-- assigned to cluster
 3 2 1 4 0 | 1
 0 6 2 2 0 | 2
 5 2 1 2 0 | 3
 1 3 2 0 4 | 4
 2 1 3 4 0 | 5

Cluster 0 <-- 3
Cluster 1 <-- 2
Cluster 2 <-- 5
Cluster 3 <-- 1
Cluster 4 <-- 4

Incorrectly clustered instances :   28.0   56      %
```

SINTAGMAS NOMINAIS/ENANCIB05

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 5 -S 10
Relation:     tc-weka.filters.unsupervised.attribute.StringToNominal-
C1-weka.filters.unsupervised.attribute.Remove-R2
Instances:    50
Attributes:   52

Test mode:    Classes to clusters evaluation on training data

=== Model and evaluation on training set ===

kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 37.34862520548471


Clustered Instances

0       9 ( 18%)
1       8 ( 16%)
2      13 ( 26%)
3      14 ( 28%)
4       6 ( 12%)


Class attribute: grupo
Classes to Clusters:

 0 1 2 3 4  <-- assigned to cluster
 0 3 3 3 1 | 1
 4 0 3 1 2 | 2
 1 1 3 3 2 | 3
 1 2 2 4 1 | 4
 3 2 2 3 0 | 5

Cluster 0 <-- 2
Cluster 1 <-- 1
Cluster 2 <-- 5
Cluster 3 <-- 4
Cluster 4 <-- 3

Incorrectly clustered instances :   35.0   70      %
```

SINTAGMAS NOMINAIS PONTUADOS/ENANCIB05

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 5 -S 10
Relation:     tr-weka.filters.unsupervised.attribute.Remove-R53-
weka.filters.unsupervised.attribute.StringToNominal-C1-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    50
Attributes:   51
Test mode:    Classes to clusters evaluation on training data

=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 36.45431942699116

Clustered Instances

0       9 ( 18%)
1      14 ( 28%)
2       8 ( 16%)
3      15 ( 30%)
4       4 (  8%)


Class attribute: grupo
Classes to Clusters:

 0 1 2 3 4  <-- assigned to cluster
 7 1 1 1 0 | 1
 1 3 1 3 2 | 2
 1 3 2 3 1 | 3
 0 4 2 3 1 | 4
 0 3 2 5 0 | 5

Cluster 0 <-- 1
Cluster 1 <-- 4
Cluster 2 <-- 3
Cluster 3 <-- 5
Cluster 4 <-- 2

Incorrectly clustered instances :   30.0   60      %
```

TERMOS/JORNAIS04

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 4 -S 10
Relation:     q-tt-
weka.filters.unsupervised.attribute.StringToNominal-C1-
weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    Classes to clusters evaluation on training data
=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 19
Within cluster sum of squared errors: 162.54369559657385

Clustered Instances

0       25 ( 16%)
1       41 ( 26%)
2       42 ( 26%)
3       52 ( 33%)


Class attribute: gt
Classes to Clusters:

  0  1  2  3  <-- assigned to cluster
  6 15  2 17 | 1
  6 11  0 23 | 2
  4  3 33  0 | 3
  9 12  7 12 | 4

Cluster 0 <-- 4
Cluster 1 <-- 1
Cluster 2 <-- 3
Cluster 3 <-- 2

Incorrectly clustered instances :   80.0   50      %
```

TERMOS SEM STOPWORDS/JORNAIS04

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 4 -S 10
Relation:     q-tts-
weka.filters.unsupervised.attribute.StringToNominal-C1-
weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    Classes to clusters evaluation on training data
=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 6
Within cluster sum of squared errors: 156.8196373288044

Clustered Instances

0       39 ( 24%)
1       38 ( 24%)
2       18 ( 11%)
3       65 ( 41%)


Class attribute: gt
Classes to Clusters:

  0  1  2  3  <-- assigned to cluster
  1  1 14 24 | 1
 38  0  1  1 | 2
  0 37  0  3 | 3
  0  0  3 37 | 4

Cluster 0 <-- 2
Cluster 1 <-- 3
Cluster 2 <-- 1
Cluster 3 <-- 4

Incorrectly clustered instances :   34.0   21.25   %
```

SINTAGMAS NOMINAIS/JORNAIS04

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 4 -S 10
Relation:     q-tc-
weka.filters.unsupervised.attribute.StringToNominal-C1-
weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    Classes to clusters evaluation on training data
=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 7
Within cluster sum of squared errors: 159.860516930811


Clustered Instances

0       57 ( 36%)
1       52 ( 33%)
2       16 ( 10%)
3       35 ( 22%)


Class attribute: gt
Classes to Clusters:

  0  1  2  3  <-- assigned to cluster
  0 31  5  4 | 1
 27  3  5  5 | 2
 30  4  3  3 | 3
  0 14  3 23 | 4

Cluster 0 <-- 3
Cluster 1 <-- 1
Cluster 2 <-- 2
Cluster 3 <-- 4

Incorrectly clustered instances :   71.0   44.375  %
```

SINTAGMAS NOMINAIS PONTUADOS/JORNAIS04

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 4 -S 10
Relation:     q-tr-
weka.filters.unsupervised.attribute.StringToNominal-C1-
weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    Classes to clusters evaluation on training data
=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 157.74296025684458

Clustered Instances

0       49 ( 31%)
1       73 ( 46%)
2       14 (  9%)
3       24 ( 15%)


Class attribute: gt
Classes to Clusters:

  0  1  2  3  <-- assigned to cluster
  2 28  3  7 | 1
 26  7  4  3 | 2
 20 12  3  5 | 3
  1 26  4  9 | 4

Cluster 0 <-- 2
Cluster 1 <-- 1
Cluster 2 <-- 3
Cluster 3 <-- 4

Incorrectly clustered instances :   94.0   58.75   %
```

SINTAGMAS NOMINAIS ANINHADOS/JORNAIS04

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 4 -S 10
Relation:     q-trn-weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    Classes to clusters evaluation on training data
=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 7
Within cluster sum of squared errors: 183.10294702257517


Clustered Instances

0       53 ( 33%)
1       40 ( 25%)
2       15 (  9%)
3       52 ( 33%)


Class attribute: gt
Classes to Clusters:

  0  1  2  3  <-- assigned to cluster
  6  5 11 18 | 1
 37  0  1  2 | 2
  1 34  0  5 | 3
  9  1  3 27 | 4

Cluster 0 <-- 2
Cluster 1 <-- 3
Cluster 2 <-- 1
Cluster 3 <-- 4

Incorrectly clustered instances :   51.0   31.875  %
```

SINTAGMAS NOMINAIS ANINHADOS PONTUADOS/JORNAIS04

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 4 -S 10
Relation:     q-tra-weka.filters.unsupervised.attribute.Remove-R163-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    160
Attributes:   161
              [list of attributes omitted]
Test mode:    Classes to clusters evaluation on training data
=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 8
Within cluster sum of squared errors: 176.77158490696897


Clustered Instances

0       47 ( 29%)
1       41 ( 26%)
2       16 ( 10%)
3       56 ( 35%)


Class attribute: gt
Classes to Clusters:

  0  1  2  3  <-- assigned to cluster
  6  6 16 12 | 1
 39  0  0  1 | 2
  1 35  0  4 | 3
  1  0  0 39 | 4

Cluster 0 <-- 2
Cluster 1 <-- 3
Cluster 2 <-- 1
Cluster 3 <-- 4

Incorrectly clustered instances :   31.0   19.375  %
```

VII – Número descritores extraídos de cada documento por método

Corpus: **ENANCIB05**

N. Doc	GT	TT	TTS	TC	TR
1	*1*	*1620*	*1527*	*1029*	*1040*
2	*1*	*1735*	*1644*	*1065*	*1095*
3	*1*	*1659*	*1561*	*1047*	*1073*
4	*1*	*1381*	*1291*	*1041*	*1049*
5	*1*	*1614*	*1542*	*1207*	*1236*
6	*1*	*1752*	*1654*	*1129*	*1151*
7	*1*	*1344*	*1280*	*1019*	*1023*
8	*1*	*1916*	*1817*	*1121*	*1133*
9	*1*	*1328*	*1248*	*835*	*844*
10	*1*	*1553*	*1463*	*903*	*927*
C	*1*	*5378*	*5230*	*4794*	*4881*
1	*2*	*2293*	*2208*	*1716*	*1744*
2	*2*	*1227*	*1157*	*947*	*962*
3	*2*	*1489*	*1418*	*1007*	*1018*
4	*2*	*1758*	*1684*	*1179*	*1203*
5	*2*	*1386*	*1317*	*997*	*1009*
6	*2*	*1485*	*1412*	*981*	*991*
7	*2*	*1748*	*1668*	*1213*	*1242*
8	*2*	*1794*	*1689*	*1209*	*1239*
9	*2*	*1373*	*1311*	*935*	*949*
10	*2*	*1403*	*1325*	*1034*	*1049*
C	*2*	*5548*	*5414*	*5320*	*5403*
1	*3*	*1468*	*1384*	*1045*	*1063*
2	*3*	*1577*	*1491*	*996*	*1005*
3	*3*	*1207*	*1134*	*797*	*802*
4	*3*	*1598*	*1516*	*1053*	*1072*
5	*3*	*1137*	*1065*	*762*	*773*
6	*3*	*1371*	*1288*	*951*	*957*
7	*3*	*1589*	*1493*	*894*	*914*
8	*3*	*1347*	*1265*	*1010*	*1021*
9	*3*	*1786*	*1689*	*1119*	*1135*
10	*3*	*1160*	*1096*	*751*	*766*
C	*3*	*4673*	*4542*	*4152*	*4212*
1	*4*	*1397*	*1324*	*894*	*907*
2	*4*	*1413*	*1324*	*1018*	*1034*
3	*4*	*1484*	*1406*	*1049*	*1063*
4	*4*	*1067*	*1002*	*803*	*811*
5	*4*	*1546*	*1478*	*1156*	*1167*
6	*4*	*1626*	*1538*	*1109*	*1128*
7	*4*	*1749*	*1666*	*1255*	*1279*
8	*4*	*1057*	*983*	*790*	*806*
9	*4*	*1341*	*1273*	*953*	*963*
10	*4*	*1542*	*1463*	*1081*	*1102*
C	*4*	*4534*	*4398*	*4442*	*4497*
1	*5*	*1639*	*1569*	*1306*	*1316*
2	*5*	*1714*	*1620*	*1055*	*1082*
3	*5*	*1405*	*1322*	*1043*	*1062*
4	*5*	*1446*	*1358*	*981*	*994*
5	*5*	*1316*	*1255*	*826*	*849*
6	*5*	*1604*	*1526*	*1040*	*1055*
7	*5*	*1502*	*1417*	*968*	*1002*
8	*5*	*1526*	*1438*	*1009*	*1022*
9	*5*	*883*	*819*	*518*	*534*
10	*5*	*1799*	*1714*	*1218*	*1245*
C	*5*	*5205*	*5066*	*4788*	*4863*

Corpus: **JORNAIS04**

				Método			
N. DOC	**TEMA**	**TT**	**TTS**	**TC**	**TR**	**TCA**	**TRA**
1	*Informática*	867	805	423	437	677	677
2	*Informática*	239	209	98	100	153	153
3	*Informática*	349	301	194	197	267	267
4	*Informática*	284	247	116	118	167	167
5	*Informática*	751	686	350	360	568	568
6	*Informática*	162	143	57	57	111	111
7	*Informática*	380	337	178	179	268	268
8	*Informática*	518	470	255	260	388	388
9	*Informática*	321	288	160	162	214	214
10	*Informática*	123	104	49	50	74	74
11	*Informática*	123	106	48	50	84	84
12	*Informática*	651	589	346	350	525	525
13	*Informática*	326	283	169	170	231	231
14	*Informática*	223	196	94	95	134	134
15	*Informática*	243	216	106	108	167	167
16	*Informática*	662	597	349	355	528	528
17	*Informática*	348	308	142	142	223	223
18	*Informática*	236	215	96	97	168	168
19	*Informática*	290	263	125	126	199	199
20	*Informática*	212	185	83	84	129	129
21	*Informática*	230	201	106	108	174	174
22	*Informática*	280	244	138	141	205	205
23	*Informática*	383	335	193	195	257	257
24	*Informática*	328	288	138	140	219	219
25	*Informática*	759	697	438	445	629	629
26	*Informática*	265	236	128	133	190	190
27	*Informática*	347	305	152	157	251	251
28	*Informática*	493	451	234	240	391	391
29	*Informática*	271	239	125	127	191	191
30	*Informática*	363	331	182	186	289	289
31	*Informática*	401	354	216	221	289	289
32	*Informática*	356	317	169	172	250	250
33	*Informática*	841	775	517	523	778	778
34	*Informática*	196	184	81	85	161	161
35	*Informática*	256	228	104	105	173	173
36	*Informática*	344	295	146	146	189	189
37	*Informática*	269	243	119	122	224	224
38	*Informática*	490	452	249	254	384	384
39	*Informática*	259	232	103	106	156	156
40	*Informática*	743	685	409	418	639	639
1	*Mundo*	492	438	190	193	299	299

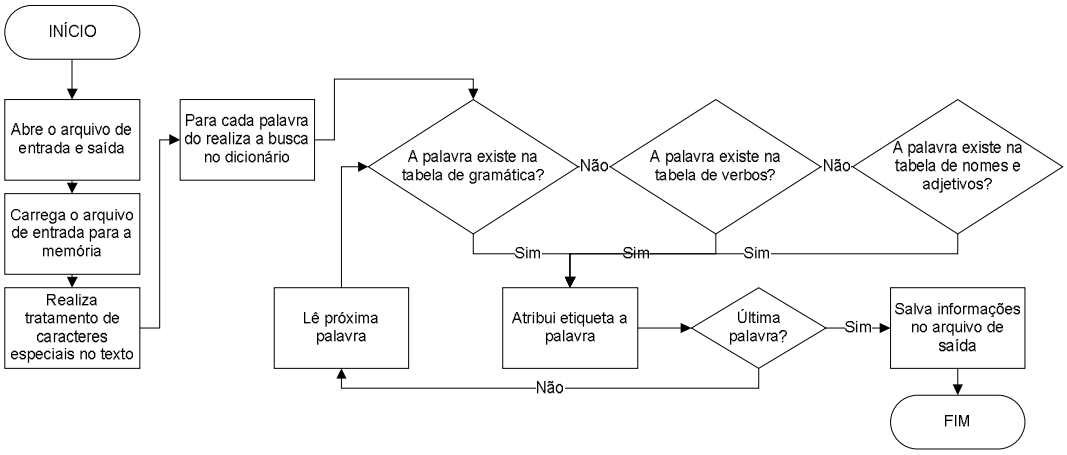
		Método					
N. DOC	**TEMA**	**TT**	**TTS**	**TC**	**TR**	**TCA**	**TRA**
2	*Mundo*	514	466	215	218	342	342
3	*Mundo*	326	283	137	142	224	224
4	*Mundo*	201	180	77	79	130	130
5	*Mundo*	243	219	108	109	159	159
6	*Mundo*	197	172	86	87	140	140
7	*Mundo*	214	186	97	98	166	166
8	*Mundo*	233	205	89	91	146	146
9	*Mundo*	228	197	100	101	156	156
10	*Mundo*	295	257	123	129	213	213
11	*Mundo*	279	242	117	118	190	190
12	*Mundo*	589	528	251	256	417	417
13	*Mundo*	264	237	97	99	170	170
14	*Mundo*	290	261	131	133	230	230
15	*Mundo*	597	539	269	273	413	413
16	*Mundo*	417	365	141	144	222	222
17	*Mundo*	230	199	102	104	167	167
18	*Mundo*	296	262	141	142	202	202
19	*Mundo*	297	265	128	131	202	202
20	*Mundo*	259	231	110	112	165	165
21	*Mundo*	281	255	115	119	198	198
22	*Mundo*	387	344	157	159	230	230
23	*Mundo*	255	231	120	123	189	189
24	*Mundo*	257	227	110	111	189	189
25	*Mundo*	295	263	120	122	189	189
26	*Mundo*	304	269	135	138	208	208
27	*Mundo*	312	277	136	138	226	226
28	*Mundo*	308	258	121	122	202	202
29	*Mundo*	349	317	185	191	264	264
30	*Mundo*	321	280	139	144	248	248
31	*Mundo*	284	255	119	126	212	212
32	*Mundo*	278	234	114	116	173	173
33	*Mundo*	253	222	109	109	169	169
34	*Mundo*	268	242	134	137	215	215
35	*Mundo*	303	261	121	124	195	195
36	*Mundo*	240	217	109	113	163	163
37	*Mundo*	330	293	143	150	231	231
38	*Mundo*	280	247	129	135	229	229
39	*Mundo*	335	298	149	150	215	215
40	*Mundo*	336	293	148	150	223	223
1	*Turismo*	482	444	271	271	407	407
2	*Turismo*	350	318	206	209	316	316
3	*Turismo*	412	383	251	261	382	382
4	*Turismo*	423	391	237	243	384	384

		Método					
N. DOC	**TEMA**	**TT**	**TTS**	**TC**	**TR**	**TCA**	**TRA**
5	*Turismo*	451	423	234	240	376	376
6	*Turismo*	427	386	248	252	388	388
7	*Turismo*	446	412	239	250	387	387
8	*Turismo*	442	403	231	235	359	359
9	*Turismo*	450	403	224	227	348	348
10	*Turismo*	464	435	232	235	369	369
11	*Turismo*	431	398	243	244	358	358
12	*Turismo*	429	391	230	233	351	351
13	*Turismo*	443	404	224	229	330	330
14	*Turismo*	435	403	252	257	393	393
15	*Turismo*	440	418	321	326	427	427
16	*Turismo*	439	403	215	216	341	341
17	*Turismo*	405	379	257	262	363	363
18	*Turismo*	439	394	220	221	316	316
19	*Turismo*	422	385	220	228	342	342
20	*Turismo*	378	340	205	208	306	306
21	*Turismo*	417	379	221	222	335	335
22	*Turismo*	425	388	245	250	364	364
23	*Turismo*	375	337	204	205	302	302
24	*Turismo*	405	362	194	195	294	294
25	*Turismo*	409	381	202	206	337	337
26	*Turismo*	371	327	193	197	266	266
27	*Turismo*	334	312	203	208	283	283
28	*Turismo*	411	368	189	198	301	301
29	*Turismo*	397	367	221	224	337	337
30	*Turismo*	397	365	219	224	319	319
31	*Turismo*	405	368	182	183	262	262
32	*Turismo*	350	322	191	194	289	289
33	*Turismo*	359	313	191	195	275	275
34	*Turismo*	294	260	170	173	254	254
35	*Turismo*	387	339	156	157	209	209
36	*Turismo*	383	361	206	211	338	338
37	*Turismo*	332	303	200	201	278	278
38	*Turismo*	365	325	174	178	284	284
39	*Turismo*	350	323	179	181	263	263
40	*Turismo*	345	313	186	191	267	267
1	*Veículo*	187	169	75	77	123	123
2	*Veículo*	227	187	80	80	107	107
3	*Veículo*	203	182	86	89	141	141
4	*Veículo*	203	177	83	84	145	145
5	*Veículo*	211	186	86	88	140	140
6	*Veículo*	221	186	78	78	119	119
7	*Veículo*	218	192	88	88	152	152

		Método					
N. DOC	**TEMA**	**TT**	**TTS**	**TC**	**TR**	**TCA**	**TRA**
8	*Veículo*	193	162	78	78	134	134
9	*Veículo*	209	186	84	87	124	124
10	*Veículo*	215	189	86	88	156	156
11	*Veículo*	208	186	89	90	145	145
12	*Veículo*	202	177	94	97	153	153
13	*Veículo*	196	175	97	98	149	149
14	*Veículo*	217	187	102	104	149	149
15	*Veículo*	212	180	84	85	113	113
16	*Veículo*	228	185	76	79	106	106
17	*Veículo*	202	176	84	86	146	146
18	*Veículo*	249	207	90	92	141	141
19	*Veículo*	209	183	80	83	155	155
20	*Veículo*	224	196	108	113	164	164
21	*Veículo*	191	175	116	118	164	164
22	*Veículo*	233	204	92	95	150	150
23	*Veículo*	225	202	85	87	152	152
24	*Veículo*	233	205	110	110	186	186
25	*Veículo*	242	224	96	96	185	185
26	*Veículo*	230	204	104	106	175	175
27	*Veículo*	254	234	102	104	177	177
28	*Veículo*	267	234	127	131	197	197
29	*Veículo*	245	217	107	108	168	168
30	*Veículo*	166	149	109	111	154	154
31	*Veículo*	266	232	105	109	175	175
32	*Veículo*	272	238	104	105	175	175
33	*Veículo*	252	221	100	103	171	171
34	*Veículo*	286	261	138	139	220	220
35	*Veículo*	206	184	119	120	165	165
36	*Veículo*	260	227	106	108	192	192
37	*Veículo*	252	223	107	108	186	186
38	*Veículo*	297	263	140	143	217	217
39	*Veículo*	273	242	109	114	174	174
40	*Veículo*	272	234	108	111	174	174

VIII – Algoritmos do OGMA

Algoritmo para etiquetar o documento.



Códificação em C#

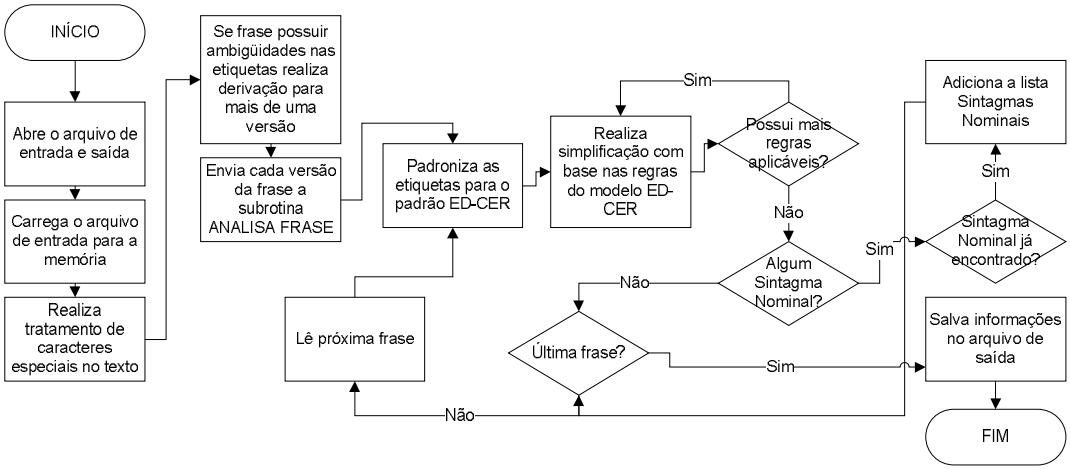
```
 OleDbConnection oleDb = new System.Data.OleDb.OleDbConnection();
 oleDb.ConnectionString  =  "Provider=Microsoft.Jet.OLEDB.4.0;Data  Source=ogma.mdb;Jet
OLEDB:Database Password=ogma.lcm;";
 oleDb.Open();
 TextReader tr = new StreamReader(fin, Encoding.GetEncoding("ISO-8859-1"));
 //FileInfo trs = new FileInfo(args[1]);
 //StreamWriter Tex = trs.CreateText();
 StreamWriter Tex = new StreamWriter(fout, false, Encoding.GetEncoding("ISO-8859-1"));

 Console.WriteLine("Etiquetando o arquivo " + fin);
 Console.WriteLine("Escrevendo no arquivo "+fout);

 string text = null;

 int ContP = 0;


 while ((text = tr.ReadLine()) != null)
 {
        //char[]      delimiterChars      =      {'      ',      ',',      '.',      ':',
       '\t','!','?','/','<','>','(',')'};
         char[] delimiterChars = { ' ', '\t' };
         text = " " + text.ToLower();

         // inicia o tratamento do texto

         text = text.Replace(" do que ", " que ");
         text = text.Replace(" ao ", " a o ");
         text = text.Replace(" aos ", " a os ");
         text = text.Replace(" pela ", " por a ");
         text = text.Replace(" pelas ", " por a ");
         text = text.Replace(" pelos ", " por o ");
         text = text.Replace(" pelo ", " por e ");

         text = text.Replace(" neste ", " em este ");
         text = text.Replace(" nisto ", " em isto ");

         text = text.Replace(" nums ", " em ums ");
         text = text.Replace(" numas ", " em umas ");
         text = text.Replace(" num ", " em um ");
         text = text.Replace(" numa ", " em uma ");
```

```
text = text.Replace(" dums ", " de ums ");
text = text.Replace(" dumas ", " de umas ");
text = text.Replace(" dum ", " de um ");
text = text.Replace(" duma ", " de uma ");
text = text.Replace(" nos ", " em os ");
text = text.Replace(" dos ", " de os ");
text = text.Replace(" do ", " de o ");
text = text.Replace(" das ", " de as ");
text = text.Replace(" da ", " de a ");
text = text.Replace(" nas ", " em as ");
text = text.Replace(" no ", " em o ");
text = text.Replace(" na ", " em a ");
text = text.Replace(" ás ", " aos as ");

//conjunto de palavras
text = text.Replace(" a fim de ", " a+fim+de ");
text = text.Replace(" a que ", " a+que ");
text = text.Replace(" a qual ", " a+qual ");
text = text.Replace(" a respeito de ", " a+respeito+de ");
text = text.Replace(" abaixo de ", " abaixo+de ");
text = text.Replace(" acerca de ", " acerca+de ");
text = text.Replace(" acima de ", " acima+de ");
text = text.Replace(" além de ", " além+de ");
text = text.Replace(" antes de ", " antes+de ");
text = text.Replace(" ao invés de ", " ao+invés+de ");
text = text.Replace(" ao redor de ", " ao+redor+de ");
text = text.Replace(" apessar de ", " apessar+de ");
text = text.Replace(" as quais ", " as+quais ");
text = text.Replace(" até a ", " até+a ");
text = text.Replace(" bem como ", " bem+como ");
text = text.Replace(" como também ", " como+também ");
text = text.Replace(" como um ", " como+um ");
text = text.Replace(" de acordo com ", " de+acordo+com ");
text = text.Replace(" debaixo de ", " debaixo+de ");
text = text.Replace(" defronte de ", " defronte+de ");
text = text.Replace(" dentreo de ", " dentreo+de ");
text = text.Replace(" depois de ", " depois+de ");
text = text.Replace(" diante de ", " diante+de ");
text = text.Replace(" e não ", " e+não ");
text = text.Replace(" em cima de ", " em+cima+de ");
text = text.Replace(" em face de ", " em+face+de ");
text = text.Replace(" em frente a ", " em+frente+a ");
text = text.Replace(" em frente de ", " em+frente+de ");
text = text.Replace(" em lugar de ", " em+lugar+de ");
text = text.Replace(" em vez de ", " em+vez+de ");
text = text.Replace(" mas ainda ", " mas+ainda ");
text = text.Replace(" mas também ", " mas+também ");
text = text.Replace(" não obstane ", " não+obstane ");
text = text.Replace(" não obstante ", " não+obstante ");
text = text.Replace(" não sí ", " não+sí ");
text = text.Replace(" não só ", " não+só ");
text = text.Replace(" no caso de ", " no+caso+de ");
text = text.Replace(" no entanto ", " no+entanto ");
text = text.Replace(" o qual ", " o+qual ");
text = text.Replace(" o que ", " o+que ");
text = text.Replace(" os quais ", " os+quais ");
text = text.Replace(" para com ", " para+com ");
text = text.Replace(" perto de ", " perto+de ");
text = text.Replace(" por conseguinte ", " por+conseguinte ");
text = text.Replace(" por isso ", " por+isso ");
text = text.Replace(" por trás de ", " por+trás+de ");



//text = text.Replace('-', ' ');
text = text.Replace("/", " ");
text = text.Replace("@", " ");
text = text.Replace("\'", " ");
text = text.Replace("\"", " ");
text = text.Replace('\u0093', ' ');
text = text.Replace('\u0094', ' ');
text = text.Replace("\'", " ");
text = text.Replace(",", " ,/VG ");
text = text.Replace("]", " ]/PN ");
text = text.Replace("[", " [/PN ");
text = text.Replace(")", " )/PN ");
text = text.Replace("(", " (/PN ");
```

```
        text = text.Replace(">", " >/PN ");
        text = text.Replace("<", " </PN ");
        text = text.Replace("=", " =/PN ");
        text = text.Replace(".", " ./PN ");
        text = text.Replace("!", " !/PN ");
        text = text.Replace(":", " :/PN ");
        text = text.Replace(";", " ;/PN ");
        text = text.Replace("?", " ?/PN ");

        text = text.Replace("-se ", "/VB a ele ");
        text = text.Replace("-lhe ", "/VB a ele ");
        text = text.Replace("-la ", "/VB a ela ");
        text = text.Replace("-lo ", "/VB a ele ");
        text = text.Replace("-las ", "/VB a elas ");
        text = text.Replace("-los ", "/VB a eles ");
        text = text.Replace("-las ", "/VB a elas ");
        text = text.Replace("-los ", "/VB a eles ");


        string[] words = text.Split(delimiterChars);
        string Palavra;
        string PalavraOriginal;
        string Etiqueta = "";

        foreach (string s in words)
        {
            ContP++;
            Etiqueta = "";
            Palavra = s;
            PalavraOriginal = Palavra;


            if (s != "") {
Tex.Write(s);
if (!s.Contains("/") && (s!=Environment.NewLine)) Tex.Write("/");
            }


            //número
            try
            {
if (Convert.ToInt32(s) > 0) Tex.Write("NA");
            }
            catch { }


            // palavras simples
            Palavra = Palavra.Replace("\"", " ");
            //Console.WriteLine("[" + Palavra + "]");
            if (Etiqueta=="") Etiqueta = BuscaPalavra(Palavra, "Gramatica", oleDb);


            if (Etiqueta == "")
            {
// procura na lista de verbos
Etiqueta = Etiqueta + BuscaPalavra(Palavra, "Verbos", oleDb);


// procura na lista de nomes
Etiqueta = Etiqueta + BuscaPalavra(Palavra, "Nomes", oleDb);

if ((Palavra.Contains('-')) && (Etiqueta==""))
{
    Palavra = Palavra.Substring(Palavra.LastIndexOf('-') + 1);
    Etiqueta = Etiqueta + BuscaPalavra(Palavra, "Nomes", oleDb);
}

if (Etiqueta == "")
{
    if (Palavra.EndsWith("íssimo")) Etiqueta = "AJ";
    if (Palavra.EndsWith("íssima")) Etiqueta = "AJ";
    if (Palavra.EndsWith("oso")) Etiqueta = "AJ";
    if (Palavra.EndsWith("rimo")) Etiqueta = "AJ";
    if (Palavra.EndsWith("inho")) Etiqueta = "AJ";
    if (Palavra.EndsWith("inha")) Etiqueta = "AJ";

}
```

```
  //nomes proprios
  //if (PalavraOriginal.Length > 2)
  //          if   (PalavraOriginal   ==   PalavraOriginal.ToUpper().Substring(0,   1)   +
PalavraOriginal.ToLower().Substring(1, PalavraOriginal.Length - 1)) Etiqueta = "NP";

            }


            // Se não encontro a classe classifica como Nome proprio
            if ((Etiqueta == "") && (Palavra.Length>2) && (!Palavra.EndsWith("VB")) &&
(!Palavra.EndsWith("PN")) && (!Palavra.EndsWith("VG")))
  Etiqueta = "SU";

            //if (s != "")
            if (!Palavra.Contains("PN"))
  Tex.Write(Etiqueta + " ");
            else
  Tex.WriteLine(Etiqueta);
          }
          //Tex.WriteLine();
 }
 Console.WriteLine("Número de palavras analisadas {0}", ContP);

 oleDb.Close();
 tr.Close();
 Tex.Close();
```

Algoritmo para extrair os sintagmas nominais.



Códificação em C#

```
public string AnalisaFrase(string frase)
{
    string rst;
    string palavra;
    rst = "";
    // Normaliza as tags para o padrão ED-CER
    frase = frase.Replace("/AD", "/AR");
    frase = frase.Replace("/AI", "/AR");
    frase = frase.Replace("/VP", "/AJ");
    frase = frase.Replace("/NC", "/NU");
    frase = frase.Replace("/NM", "/NU");
    frase = frase.Replace("/NO", "/NU");
    frase = frase.Replace("/NR", "/NU");
    //frase = frase.Replace("/VG", "/CO");
    frase = frase.Replace("/CJ", "/CO");
    frase = frase.Replace("/PL", "/LG");


    //ADAPTAÇÂO DO MODO SELETOR DO METODO ED-CER
    //Inicia analise
    frase = frase.Replace("/LG", "/ct");
    frase = frase.Replace("/VB", "/ct");
    frase = frase.Replace("/PN", "/ct");

    int alterado = 1;
    while (alterado == 1)
    {
        alterado = 0;
        frase.Trim();
        string[] palavras = frase.Split(' ');
        int npalavras = palavras.GetLength(0);
        int ppalavras = 0;
        string esquerda = "";
        while (ppalavras < npalavras)
        {

            palavra = palavras[ppalavras];
            if (palavra != "")
            {
                string[] PC0 = palavras[ppalavras].Split('/');
                string[] PC1 = { "", "" };
                string[] PC2 = { "", "" };
                string[] PC3 = { "", "" };
```

```
                        esquerda = esquerda + " " + PC0[0];

                        if (ppalavras + 1 < npalavras) PC1 = palavras[ppalavras +
1].Split('/');
                        if (ppalavras + 2 < npalavras) PC2 = palavras[ppalavras +
2].Split('/');
                        if (ppalavras + 3 < npalavras) PC3 = palavras[ppalavras +
3].Split('/');
                        if ((PC0[1] == "AV") && (PC1[1] == "ct"))
                        {
                            esquerda = esquerda + "_" + PC1[0] + "/ct";
                            ppalavras++;
                        }
                        else
                            if ((PC0[1] == "CO") && (PC1[1] == "ct"))
                            {
                                esquerda = esquerda + "_" + PC1[0] + "/ct";
                                ppalavras++;
                            }
                            else
                                if ((PC0[1] == "AR") && (PC1[1] == "ct"))
                                {
                                    esquerda = esquerda + "_" + PC1[0] + "/ct";
                                    ppalavras++;
                                }
                                else
                                    if ((PC0[1] == "NU") && (PC1[1] == "ct"))
                                    {
                                        esquerda = esquerda + "_" + PC1[0] + "/ct";
                                        ppalavras++;
                                    }
                                    else
                                        if ((PC0[1] == "ct") && (PC1[1] == "PR"))
                                        {
                                            esquerda = esquerda + "_" + PC1[0] + "/ct";
                                            ppalavras++;
                                        }
                                        else
                                            if ((PC0[1] == "ct") && (PC1[1] == "CO"))
                                            {
                                                esquerda = esquerda + "_" + PC1[0] +
"/ct";
                                                ppalavras++;
                                            }
                                            else
                                                esquerda = esquerda + "/" + PC0[1];
                    }
                    ppalavras++;
                }
                if (frase != esquerda) alterado = 1;
                frase = esquerda;
            }

            //ADAPTAÇÂO DO MODO ANALISADOR DO METODO ED-CER
            // Inicia gramatica

            frase = frase.Replace("/AR","/de");
            frase = frase.Replace("/PD","/de");
            frase = frase.Replace("/PI","/de");
            frase = frase.Replace("/AJ","/MD");
            frase = frase.Replace("/NU","/MD");
            frase = frase.Replace("/PS","/MD");
            frase = frase.Replace("/CO","/co");
            frase = frase.Replace("/PR","/pr");
            frase = frase.Replace("/SU","/re");
            frase = frase.Replace("/PP","/de"); // Modificação no ED-CER antes PP -> re
            frase = frase.Replace("/NP","/re");

            //Incia analise
            frase = frase.Replace("/re", "/NS");
            alterado = 1;
            while (alterado == 1)
            {
                alterado = 0;
                frase.Trim();
                string[] palavras = frase.Split(' ');
```

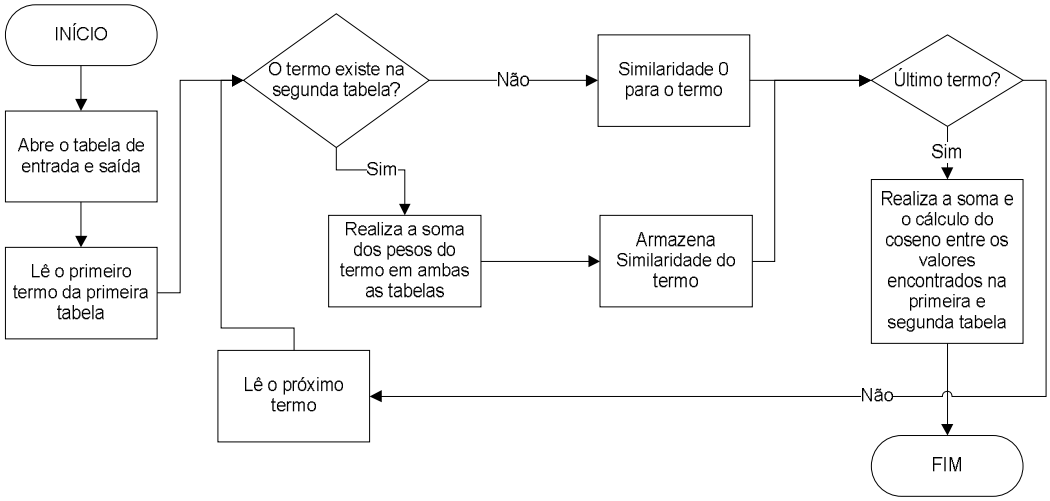
```
                int npalavras = palavras.GetLength(0);
                int ppalavras = 0;
                string esquerda = "";
                while (ppalavras < npalavras)
                {
                    palavra = palavras[ppalavras];


                    if (palavra != "")
                    {
                        string[] PC0 = palavras[ppalavras].Split('/');
                        string[] PC1 = {"",""};
                        string[] PC2 = {"",""};
                        string[] PC3 = {"",""};

                        esquerda = esquerda + " " + PC0[0];

                        if           (ppalavras+1<npalavras)           PC1           =
palavras[ppalavras+1].Split('/');
                        if           (ppalavras+2<npalavras)           PC2           =
palavras[ppalavras+2].Split('/');
                        if           (ppalavras+3<npalavras)           PC3           =
palavras[ppalavras+3].Split('/');
                        if ((PC0[1] == "AV") && (PC1[1] == "AV"))
                        {
                            esquerda = esquerda + PC1[0] + "/AV";
                            ppalavras++;
                        } else
                        if ((PC0[1] == "AV") && (PC1[1] == "MD"))
                        {
                            esquerda = esquerda  + "_" + PC1[0] + "/MD";
                            ppalavras++;
                        } else
                        if ((PC0[1] == "MD") && (PC1[1] == "co") && (PC2[1] == "MD"))
                        {
                            esquerda = esquerda + "_" + PC1[0] + "_" + PC2[0] + "/MD";
                            ppalavras++;
                            ppalavras++;
                        } else
                        if ((PC0[1] == "NS") && (PC1[1] == "MD"))
                        {
                            esquerda = esquerda + "_" + PC1[0] + "/NS";
                            ppalavras++;
                        } else
                        if ((PC0[1] == "MD") && (PC1[1] == "NS"))
                        {
                            esquerda = esquerda + "_" + PC1[0] + "/NS";
                            ppalavras++;
                        } else
                        if ((PC0[1] == "NS") && (PC1[1] == "pr") && (PC2[1] == "NS"))
                        {
                            esquerda = esquerda + "_" + PC1[0] + "_" + PC2[0] + "/NS";
                            ppalavras++;
                            ppalavras++;
                        } else
                        if ((PC0[1] == "NS") && (PC1[1] == "pr") && (PC2[1] == "de") &&
(PC3[1] == "NS"))
                        {
                            esquerda = esquerda + "_" + PC1[0] + "_" + PC2[0] + "_" +
PC3[0] + "/NS";
                            ppalavras++;
                            ppalavras++;
                            ppalavras++;
                        } else
                        if ((PC0[1] == "NS") && (PC1[1] == "VG") && (PC2[1] == "MD"))
                        {
                                        esquerda = esquerda + "_" + PC1[0] + "_" +
PC2[0] + "/NS";
                                        ppalavras++;
                                        ppalavras++;
                        } else
                            if ((PC0[1] == "NS") && (PC1[1] == "VG") && (PC2[1] == "de")
&& (PC3[1] == "MD"))
                            {
                                esquerda = esquerda + "_" + PC1[0] + "_" + PC2[0] + "_"
+ PC3[0] + "/NS";
                                ppalavras++;
```

```
                                ppalavras++;
                                ppalavras++;
                            }
                            else
                            if ((PC0[1] == "NS") && (PC1[1] == "co") && (PC2[1] ==
"NS"))
                                        {
                                            esquerda = esquerda + "_" + PC1[0] + "_" +
PC2[0] + "/NS";
                                            ppalavras++;
                                            ppalavras++;
                                        } else
                                        if ((PC0[1] == "NS") && (PC1[1] == "co") &&
(PC2[1] == "de") && (PC3[1] == "NS"))
                                        {
                                            esquerda = esquerda + "_" + PC1[0] + "_" +
PC2[0] + "_" + PC3[0] + "/NS";
                                            ppalavras++;
                                            ppalavras++;
                                            ppalavras++;
                                        } else

                            if ((PC0[1] == "AV") && (PC1[1] == "NS"))
                            {
                                esquerda = esquerda + "_" + PC1[0] + "/NS";
                                ppalavras++;
                            } else
                            if ((PC0[1] == "de") && (PC1[1] == "NS"))
                            {
                                esquerda = esquerda + "_" + PC1[0] + "/NS";
                                ppalavras++;
                            }
                            else
                            {
                                esquerda = esquerda + "/" + PC0[1];

                            }
                        }
                        ppalavras++;
                    }
                    if (frase != esquerda) alterado = 1;
                    frase = esquerda;
                }

                frase = frase.Replace("/NS", "/SN");

                rst = frase;
                return rst;
            }


       public string[] ExtraiSN(string fin, string fout)
       {
           TextReader tr = new StreamReader(fin, Encoding.GetEncoding("ISO-8859-1"));
           //FileInfo trs = new FileInfo(args[1]);
           //StreamWriter Tex = trs.CreateText();
           StreamWriter Tex = new StreamWriter(fout, false, Encoding.GetEncoding("ISO-8859-
1"));
           Console.WriteLine("Extraindo sintagmas do arquivo " + fin);
           Console.WriteLine("------------------------------------------------------------
");

           string frase;
           string[] rst = {""};

           while ((frase = tr.ReadLine()) != null)
           {
          //Console.WriteLine(frase);
          char[] delimiterChars = { ' ', '\t' };
          //Deriva todas as combinações da frase no caso de ambiguidades
          frase.Trim();
          string[] words = frase.Split(delimiterChars);
          IList<string> deriva = new List<string>();
          int n = 0;
          int nd = 0;
          foreach (string s in words)
```

```
  {
 if ((s != "") && (s.Contains('/')))
 {
string[] PC = s.Split('/');
if (deriva.Count < 2500)
{

    n = PC[1].Length / 2;

    int cd = deriva.Count;
    while (n > 1)
    {
   int cdd = cd;
   while (cdd > 0)
   {
  deriva.Add(deriva[cdd - 1]);
  cdd--;
   }
   n = n - 1;
    }
    //Console.WriteLine(deriva.Count);


    n = PC[1].Length / 2;
    nd = deriva.Count;
    if (n != 0) nd = nd / n;
    if (nd == 0) nd = 1;

    int contd = 0;
    for (int x = 0; x < nd; x++)
    {
   for (int tag = 0; tag < n; tag++)
   {
  while (deriva.Count - 1 < contd) deriva.Add("");

  deriva[contd] = deriva[contd] + " " + PC[0] + "/" + PC[1].Substring(tag * 2, 2);
  contd++;
   }
    }

}
else
{
    nd = deriva.Count;
    for (int x = 0; x < nd; x++)
    {
   if (PC[1].Length > 1)
  deriva[x] = deriva[x] + " " + PC[0] + "/" + PC[1].Substring(0, 2);
   else
  deriva[x] = deriva[x] + " " + PC[0] + "/";

    }

}

 }
  }
  // Analisa o resultado de cada frase
  string resultado = "";
  string SN = "";

  IList<string> resultados = new List<string>();
  IList<string> resultadosfinal = new List<string>();
  bool JaCadastrado = false;
  foreach (string s in deriva)
  {
 //Console.WriteLine(s);
 resultado = AnalisaFrase(s);
 string[] palavras = resultado.Split(' ');

 foreach (string palavra in palavras)
 {
string[] PC = palavra.Split('/');
if (PC.Count()>1) if (PC[1] == "SN") {

    // Volta com contrações básicas
    PC[0] = PC[0].Replace("_de_os_", "_dos_");
```

```
     PC[0] = PC[0].Replace("_de_o_", "_do_");
     PC[0] = PC[0].Replace("_de_as_", "_das_");
     PC[0] = PC[0].Replace("_de_a_", "_da_");
     PC[0] = PC[0].Replace("_em_as_", "_nas_");
     PC[0] = PC[0].Replace("_em_o_", "_no_");
     PC[0] = PC[0].Replace("_em_a_", "_na_");


     SN = " "+PC[0].Replace("_"," ").Replace("+"," ")+" ";
     if (!(resultados.Contains(SN)))
     {
    resultados.Add(SN);
     }
 };
  }

   }

   // remove SN duplicados e escolhe os maiores.
   resultadosfinal.Clear();
   foreach (string SNR in resultados)
   {
  JaCadastrado = false;
  int pos = 0;
  while (pos < resultadosfinal.Count)
  {
 string SNRV = resultadosfinal[pos];
 if (SNRV == SNR) JaCadastrado = true;
 // o SN é menor e já esta cadastrado
 if (SNRV.IndexOf(SNR) >= 0)
 {
     JaCadastrado = true;
 }
 // o SN é maior que o do cadastrado
 if (SNR.IndexOf(SNRV) >= 0)
 {
     resultadosfinal[pos] = "";
 }
 pos++;
  }
  if  (!JaCadastrado)
 resultadosfinal.Add(SNR);
   }

   // grava o resultado no arquivo ou envia para a tela.
   foreach (string SNR in resultadosfinal)
   {
  if (SNR != "")
  if (fout=="temp$$.txt")
  {
 Console.WriteLine(SNR.Trim());
  }
  else
  {
 Tex.WriteLine(SNR.Trim());
 //rst..add(SNR.Trim());
  }
   }


    }
    Tex.Close();
    tr.Close();

    return rst;
}
```

Algoritmo para cálculo da similaridade



Códificação em C#

```
public void Similaridade(string ftaba, string ftabb, string fsim)
{
    IList<string> arquivoa = new List<string>();
    IList<string> arquivoaF = new List<string>();
    IList<string> arquivob = new List<string>();
    IList<string> arquivobF = new List<string>();
    IList<string> ambos = new List<string>();

    Console.WriteLine("Comparando o arquivo " + ftaba + " com " + ftabb);

    TextReader TRarqA = new StreamReader(ftaba, Encoding.GetEncoding("ISO-8859-1"));
    string text = null;
    while ((text = TRarqA.ReadLine()) != null)
    {
   if (text.Length>1) arquivoa.Add(text);
    }
    TRarqA.Close();

    TextReader TRarqB = new StreamReader(ftabb, Encoding.GetEncoding("ISO-8859-1"));
    text = null;
    while ((text = TRarqB.ReadLine()) != null)
    {
   if (text.Length > 1) arquivob.Add(text);
    }
    TRarqB.Close();


    string[] sna = {""};
    string[] snb = { "" };
    string pesob = "";
    int posa = 0;
    int posb = 0;
    bool jambos;
    while (posa < arquivoa.Count)
    {
   sna = arquivoa[posa].Split('/');
   posb = 0;
   jambos = false;
   while (posb < arquivob.Count)
   {
       snb = arquivob[posb].Split('/');
```

```
          if ((sna[0] == snb[0]) && (sna[0].Length > 1))
          {
         //ambos.Add(sna[0] + "/" + sna[2] + "/" + snb[2]);
         jambos = true;
         pesob = snb[2];
          }
          posb++;
      }
      if ((!jambos) && (sna[0].Length>1)) arquivoaF.Add(arquivoa[posa]);
      if ((jambos) &&  (sna[0].Length>1)) ambos.Add(sna[0] + "/" + sna[2] + "/" +
pesob);
      posa++;
       }
       while (posb < arquivob.Count)
       {
      snb = arquivob[posa].Split('/');
      posa = 0;
      jambos = false;
      while (posa < arquivoa.Count)
      {
          sna = arquivoa[posa].Split('/');
          if ((sna[0] == snb[0]) && (sna[0].Length > 1))
          {
         jambos = true;
          }
          posa++;
      }
      if ((!jambos) && (snb[0].Length>1)) arquivobF.Add(arquivob[posa]);
      posb++;
       }

       arquivoa = arquivoaF;
       arquivob = arquivobF;

       double da = 0;
       double db = 0;
       double num = 0;

       foreach (string s in arquivoa)
       {
      double pont = 0;
      sna = s.Split('/');
      pont = double.Parse(sna[2]);
      pont = Math.Pow(pont, 2);
      da = da + pont;
       }

       foreach (string s in arquivob)
       {
      double pont = 0;
      snb = s.Split('/');
      pont = double.Parse(snb[2]);
      pont = Math.Pow(pont, 2);
      db = db + pont;
       }

       foreach (string s in ambos)
       {
      double ponta = 0;
      double pontb = 0;
      snb = s.Split('/');
      ponta = double.Parse(snb[1]);
      pontb = double.Parse(snb[2]);
      num = num + (ponta * pontb);
      ponta = Math.Pow(ponta, 2);
      da = da + ponta;
      pontb = Math.Pow(pontb, 2);
      db = db + pontb;
       }

       double sim = 0;

       sim = num / (Math.Sqrt(da) * Math.Sqrt(db));

       Console.Write("Similaridade (cos): ");
       Console.WriteLine(String.Format("{0:0.000000}", sim));
```

```
       if (fsim != "")
       {
      Console.WriteLine("Escrevendo resultado no arquivo " + fsim);
      StreamWriter Tex = new StreamWriter(fsim, true, Encoding.GetEncoding("ISO-8859-
1"));
      Tex.WriteLine(ftaba+"/"+ftabb+"/"+String.Format("{0:0.000000}", sim));
      Tex.Close();
       }
   }
```