

**UNIVERSIDADE FEDERAL DE MINAS GERAIS  
ESCOLA DE CIÊNCIA DA INFORMAÇÃO**

**AGNALDO LOPES MARTINS**

**O USO DO SINTAGMA NOMINAL NA RECUPERAÇÃO DE DOCUMENTOS:  
PROPOSTA DE UM MECANISMO AUTOMÁTICO PARA CLASSIFICAÇÃO  
TEMÁTICA DE TEXTOS DIGITAIS**

**BELO HORIZONTE  
2014**

Agnaldo Lopes Martins

**O USO DO SINTAGMA NOMINAL NA RECUPERAÇÃO DE DOCUMENTOS:**

Proposta de um mecanismo automático para classificação  
temática de textos digitais

Tese apresentada ao Programa de Pós-graduação em  
Ciência da Informação da Universidade Federal de Minas  
Gerais, como requisito parcial para obtenção do título de  
doutor em Ciência da Informação.

Área de concentração: Produção, Organização e  
Utilização da Informação.

Linha de Pesquisa: Organização e Uso da Informação

Orientador: Prof. Dr. Renato Rocha Souza

Belo Horizonte  
2014

M386u Martins, Agnaldo Lopes.

O uso do sintagma nominal na recuperação de documentos [manuscrito] : proposta de um mecanismo automático para classificação temática de textos digitais / Agnaldo Lopes Martins. – 2014.

192 f. : il., enc.

Orientador: Renato Rocha Souza.

Tese (doutorado) – Universidade Federal de Minas Gerais, Escola de Ciência da Informação.

Referências: f. 118-123.

Anexos: f. 124-192.

1. Ciência da informação – Teses. 2. Sistemas de recuperação da informação – Teses. 3. Processamento da linguagem natural (Computação) – Teses. 4. Indexação automática – Teses. I. Título. II. Souza, Renato Rocha. III. Universidade Federal de Minas Gerais, Escola de Ciência da Informação.

CDU: 025.4.03



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 **AGNALDO LOPES MARTINS**, matrícula: 2010656088

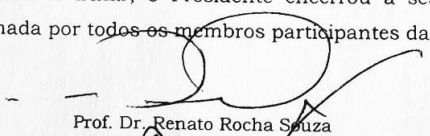
As 13:00 horas do dia 18 de agosto de 2014, reuniu-se na Escola de Ciência da Informação da UFMG a Comissão Examinadora aprovada *ad referendum* pelo Colegiado do Programa de Pós-Graduação em Ciência da Informação em 17/07/2014, para julgar, em exame final, o trabalho intitulado **O uso do sintagma nominal na recuperação de documentos: proposta de um mecanismo automático para classificação temática de textos digitais**, 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. 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 arguição 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. Flávio Codeco Coelho	APROVADO
Prof. Dr. Luiz Cláudio Gomes Maia	APROVADO
Prof. Dr. Manoel Palhares Moreira	APROVADO
Profa. Dra. Heliana Ribeiro de Mello	APROVADO
Profa. Dra. Maria Aparecida Moura	APROVADO
Profa. Dra. Renata Maria Abrantes Baracho Porto	APROVADO

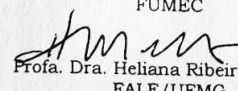
Pelas indicações, o candidato foi considerado APROVADO.

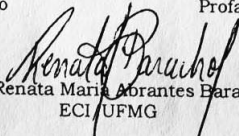
O resultado final foi comunicado publicamente ao candidato pelo Presidente da Comissão. Nada mais havendo a tratar, o Presidente encerrou a sessão, da qual foi lavrada a presente ATA que será assinada por todos os membros participantes da Comissão Examinadora.

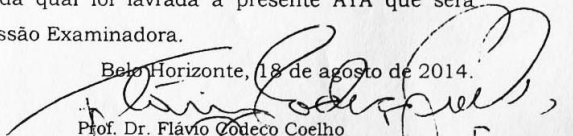
Belo Horizonte, 18 de agosto de 2014.

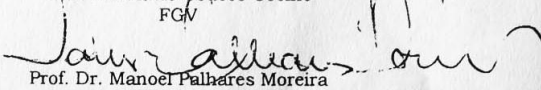
  
Prof. Dr. Renato Rocha Souza  
FGV/RJ

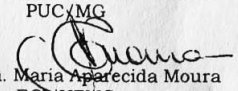
  
Prof. Dr. Luiz Cláudio Gomes Maia  
FUMEC

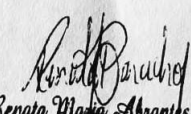
  
Profa. Dra. Heliana Ribeiro de Mello  
FALE/UFMG

  
Profa. Dra. Renata Maria Abrantes Baracho Porto  
ECI/UFMG

  
Prof. Dr. Flávio Codeco Coelho  
FGV

  
Prof. Dr. Manoel Palhares Moreira  
PUC/MG

  
Profa. Dra. Maria Aparecida Moura  
ECI/UFMG

  
Profa. Dra. Renata Maria Abrantes Baracho Porto  
Coordenadora do Programa de Pós-Graduação  
em Ciência da Informação

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

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**

**FOLHA DE APROVAÇÃO**

"O USO DO SINTAGMA NOMINAL NA RECUPERAÇÃO DE DOCUMENTOS:  
PROPOSTA DE UM MECANISMO AUTOMÁTICO PARA CLASSIFICAÇÃO TEMÁTICA  
DE TEXTOS DIGITAIS"

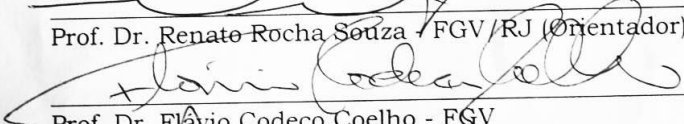
Agnaldo Lopes Martins

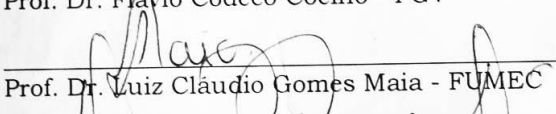
Tese submetida à Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação em Ciência da Informação da Universidade Federal de Minas Gerais, como parte dos requisitos à obtenção do título de "**doutor em Ciência da Informação**", linha de pesquisa "**Organização e Uso da Informação**".

Tese aprovada em: 18 de agosto de 2014.

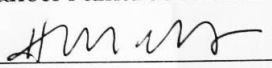
Por:

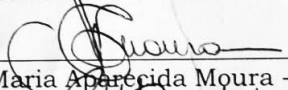
  
Prof. Dr. Renato Rocha Souza - FGV/RJ (Orientador)

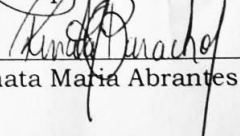
  
Prof. Dr. Flávio Codeço Coelho - FGV

  
Prof. Dr. Luiz Cláudio Gomes Maia - FUMEC

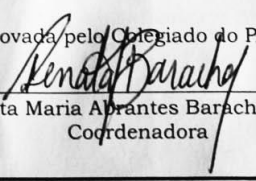
  
Prof. Dr. Manoel Palhares Moreira - PUC/MG

  
Profa. Dra. Heliana Ribeiro de Mello - FALE/UFMG

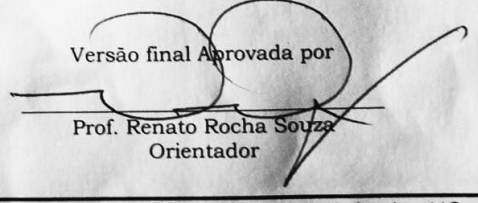
  
Profa. Dra. Maria Aparecida Moura - ECI/UFMG

  
Profa. Dra. Renata Maria Abrantes Baracho Porto - ECI/UFMG

Aprovada pelo Colegiado do PPGCI

  
Profa. Renata Maria Abrantes Baracho Porto  
Coordenadora

Versão final aprovada por

  
Prof. Renato Rocha Souza  
Orientador

*Para minha esposa Míria por aturar meus incontáveis defeitos, por ter me dado filhos maravilhosos e por ser parte da história que agora quero lhes contar.*

*Aos meus três filhos com quem tive que dividir todo esse tempo dedicado ao doutorado. Quero agora tentar devolver um pouco do tempo que privei de vocês.*

*Aos meus pais e irmãos, pois Deus foi bondoso comigo ao me colocar em uma família muito especial.*

## AGRADECIMENTOS

### ***Agradecimento especial***

*Ao meu jovem orientador, Renato, por ter acreditado em mim desde o primeiro dia em que nos conhecemos. Tenho algumas pessoas importantes na vida e você é uma delas. Esteja certo de que alguns se tornam mestres, outros, como você, nascem mestres.*

*Mais uma vez Teus desígnios me amparam.*

Quero agradecer aos colegas da TOP, especialmente ao Gilson que sempre me apoiou.

Aos amigos próximos, que aturam os meus defeitos.

Aos colegas do mestrado e doutorado, pelas discussões por vezes acaloradas.

À ECI. É impressionante o carinho de todos os funcionários, só estando lá pra saber.

À banca de qualificação, por ter me mostrado que no caminho haveria pedras.

Aos mestres digo: continuarei sendo uma sombra atenta a qualquer movimento.

Aos meus alunos. Completei o mapa, fica para vocês a missão de encontrar o tesouro.

Ao Hiram, que foi embora sem avisar, talvez para me proteger, como sempre fez.

*Certamente meus pais sabiam o que estavam fazendo quando me deram de  
presente  
um violão,  
uma enciclopédia,  
e uma máquina de escrever.  
Mas deixaram que eu compreendesse este gesto sozinho, mais tarde.*



*“Na verdade, nada nos parece mais natural, óbvio e indiscutível que as classificações dos entes, dos fatos e dos acontecimentos que constituem os quadros mentais em que estamos inseridos. Elas constituem os pontos estáveis que nos impedem de rodopiar sem solo, perdidos no desconforto do inominável, da ausência de "idades" ou "geografias". Só elas nos permitem orientar-nos no mundo à 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”.*

*Olga Pombo*

*"Os animais dividem-se em a) pertencentes ao imperador, b) embalsamados, c) amestrados, d) leões, e) sereias, f) fabulosos, g) cães soltos, h) incluídos nesta lista, i) que se agitam como loucos, j) inumeráveis, k) desenhados com um pincel finíssimo de pelo de camelo, l) etc., m) que acabam de partir o jarão, n) que de longe parecem moscas”.*

*Jorge Luís Borges*

## RESUMO

Esta tese objetivou avaliar o uso do sintagma nominal como fonte de dados para um sistema automático de classificação de documentos textuais armazenados no formato digital. Foram utilizadas diversas ferramentas tecnológicas que transformaram artigos científicos em uma lista de sintagmas nominais que foram utilizados para treinamento de um sistema classificador baseado em treinamento supervisionado. Dentre as ferramentas utilizadas o software *Palavras* foi o responsável pela identificação e remoção dos sintagmas nominais dos corporas utilizados. Para treinamento da máquina classificadora foi utilizado o aplicativo SVMLight. A metodologia foi desenvolvida em duas etapas; na primeira foi realizado um teste qualitativo na comparação entre os documentos do corpus; e na segunda etapa foi realizado o treinamento utilizando SVM com um número maior de documentos. Ao final, vários testes foram realizados sendo possível demonstrar que a metodologia proposta foi capaz de classificar documentos com alta precisão.

**Palavras-chave:** Sintagmas nominais. Processamento da linguagem natural. Classificação de documentos. Sistemas de recuperação da informação.

## **ABSTRACT**

This thesis aimed to evaluate the use of the noun phrase as a data source for an automatic classification of text documents stored in digital format. Various technological tools that have transformed scientific articles in a list of noun phrases that have been used for a classifier system based on supervised learning training. Among the tools used the words were responsible for the identification and removal of noun phrases of corporas. For training the classifier machine the application SVMLight was used. The methodology was developed in two stages; the first qualitative test was performed when comparing the documents of the corpus; and in the second stage SVM training was conducted using a larger number of documents. At the end, several tests were performed and it is possible to demonstrate that the proposed methodology was able to classify documents with high precision.

**Keywords:** Noun phrases. Natural processing language. Document classification. Information Retrieval Systems.

## LISTA DE ILUSTRAÇÕES

Figura 1. Conteúdos da Ciência da Informação pela ASIS	18
Figura 2. Um modelo de SRI	25
Figura 3. Modelos de recuperação da informação	25
Figura 4. Documentos para treinamento e para teste	31
Figura 5. Visualização hierárquica dos SN de uma frase em português	33
Figura 6. Pipeline de um sistema para PNL genérico	35
Figura 7. Ilustração de uma separação entre duas classes	40
Figura 8. Medida de discrepância entre a saída desejada e a obtida	40
Figura 9. Um exemplo de SN em níveis	42
Figura 10. Um exemplo de sintagma em forma arbórea	44
Figura 11. Exemplo de um problema entre o referente e o classificador	46
Figura 12. Conjunto representando $A=\{i+c\}$	49
Figura 13. Conjunto representando $A=\{B1+B2\}+C$	52
Figura 14. Diagrama com etapas do tratamento de prospecção	64
Figura 15. Trecho em PDF de um dos documentos utilizado	65
Figura 16. Lista de sintagmas importados para o Excel	72
Figura 17. Tela do software Oracle Virtual Box	96
Figura 18. Arquivos processados pelo software Palavras	97
Figura 19. Tela do software clasdoc criado pelo autor	99
Figura 20. Saída do processamento pelo software SVMLight	109
Figura 21. Resultado final do processamento pelo software SVMLight	109

## LISTA DE TABELAS

Tabela 1. Lista dos documentos utilizados na primeira etapa	67
Tabela 2. Total de sintagmas extraídos dos documentos	70
Tabela 3. Quantidade de repetições de determinados sintagmas	74
Tabela 4. Lista dos sintagmas que mais repetem	74
Tabela 5. Lista dos sintagmas que mais repetem em doc3	75
Tabela 6. Fórmula para calcular a semelhança entre os documentos	77
Tabela 7. Comparando Doc1 e Doc2	78
Tabela 8. Comparando Doc1 e Doc3	78
Tabela 9. Comparando Doc2 e Doc3	78
Tabela 10. Comparação entre todos os documentos	79
Tabela 11. Comparação após remoção dos quantificadores	80
Tabela 12. Percentual de melhora com stemming	81
Tabela 13. Percentual de melhora com stemming	81
Tabela 14. Percentual de melhora com convergência de sinônimos	83
Tabela 15. Origem dos documentos utilizados	85
Tabela 16. Comparação direta doc 1 do treino e doc 1 da prova	86
Tabela 17. Comparação entre teste e contraprova (Doc2 e DocP2)	87
Tabela 18. Comparação entre teste e contraprova (Doc3 e DocP3)	88
Tabela 19. Comparação entre teste e contraprova (Doc1 e DocP1)	88
Tabela 20. Comparação entre (Doc2 e DocP2)	89
Tabela 21. Comparação entre (Doc3 e DocP3)	89
Tabela 22. Total de documentos utilizados para treinamento	94
Tabela 23. Total de documentos utilizados para teste	94
Tabela 24. Número de sintagmas por documento de treinamento	107
Tabela 25. Número de sintagmas por documento de treinamento	107
Tabela 26. Número de sintagmas por documento de treinamento	108
Tabela 27. Número de sintagmas por documento de teste	108

## LISTA DE GRÁFICOS

Gráfico 1. Total de sintagmas nos três primeiros documentos	72
Gráfico 2. Total de SN após a normalização nos três documentos	73
Gráfico 3. Total de SN nos três primeiros documentos comparando com os sintagmas únicos	73
Gráfico 4. Distribuição gráfica dos sintagmas para o documento 3	76
Gráfico 5. Comparando os sintagmas da Engenharia com os sintagmas das outras duas temáticas	104
Gráfico 6. Comparando os sintagmas da História com os sintagmas das outras duas temáticas	105
Gráfico 7. Comparando os sintagmas da Letras com os sintagmas das outras duas temáticas	105
Gráfico 8. Avaliação da classificação na área de Engenharia	110
Gráfico 9. Avaliação da classificação na área de História	111
Gráfico 10. Avaliação da classificação na área de Letras	112

## LISTA DE ABREVIATURAS E SIGLAS

ASCII	American Standard Code
CI	Ciência da Informação
CG	Constraint Grammar
HTML	Hyperlink Text Markup Language
IA	Inteligência Artificial
KDT	Knowledge Discovery from Text
LC	Linguística computacional
PDF	Portable Document Format
PLN	Processamento de Linguagem Natural
PPGCI	Programa de Pós Graduação em Ciência da Informação
RBC	Raciocínio Baseado em Casos
RDF	Resource Description Framework
RI	Recuperação da Informação
RNA	Rede Neural Artificial
SAdj	Sintagma Adjetival
SAdv	Sintagma Adverbial
SN	Sintagma Nominal
SP	Sintagma Preposicional
SRI	Sistema de Recuperação da Informação
SV	Sintagma Verbal
SVM	Support Vector Machine
TI	Tecnologias de Informação
VBA	Visual Basic for Application
VISL	Visual Interactive Syntax Learning
WEKA	Waikato Environment for Knowledge Analysis

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>17</b>
1.1 Delimitação do problema .....	19
1.2 Objetivo geral .....	21
1.3 Objetivos específicos .....	21
<b>2 FUNDAMENTOS CONCEITUAIS .....</b>	<b>23</b>
2.1 Sistemas de recuperação da informação .....	24
2.2 Classificação de documentos .....	26
2.3 Linguística.....	31
2.4 Processamento da linguagem natural .....	34
2.5 Aprendizagem de máquina e Support Vector Machines (SVMs) .....	37
<b>3 O SINTAGMA NOMINAL.....</b>	<b>42</b>
3.1 A sintaxe do sintagma nominal.....	44
3.2 A identificação do referente.....	47
3.3 A identificação do classificador.....	50
3.4 Subclassificador e qualificador .....	51
3.5 O recortador e o quantificador .....	53
3.6 Influências na produção do texto escrito .....	55
3.6.1 A influência do estilo literário do autor .....	56
3.6.2 A influência do sexo do autor .....	57
3.6.3 A influência da oralidade na escrita .....	58
3.6.4 A influência do suporte digital .....	58
3.6.5 A influência do uso de certas palavras no decorrer do tempo.....	59
3.6.6 A influência do gênero textual.....	60
3.6.7 A influência de fatores sociais, emocionais e de formação do autor .....	61
<b>4 EXPERIMENTO DE PROSPECÇÃO.....</b>	<b>62</b>
4.1 Primeira etapa: Comparação direta entre os documentos.....	62
4.1.1 Escolha dos documentos.....	64
4.1.2 Preparação do Corpus .....	65
4.1.3 Converter os documentos para texto puro .....	66
4.1.4 Filtragem preliminar do conteúdo .....	67
4.1.5 Extração dos sintagmas nominais .....	68
4.1.6 Normalização da lista gerada.....	70
4.1.7 Sintagmas únicos e ordenados.....	71



4.2	Aplicação do método preliminar .....	71
4.2.1	Comparação direta entre sintagmas dos documentos .....	77
4.2.2	Comparação após remoção dos quantificadores .....	79
4.2.3	Comparação após o processo de <i>stemming</i> .....	80
4.2.4	Comparação após a convergência de sinônimos.....	82
4.2.5	Contraprova .....	84
4.2.6	Comparações da contraprova.....	86
4.2.6.1	Comparação do Doc1 do treinamento e doc1 da contraprova.....	86
4.2.6.2	Comparação do Doc2 treinamento com DocP2 da contraprova.....	87
4.2.6.3	Comparação direta Doc3 treinamento com DocP3 da contraprova .....	88
4.2.6.4	Contraprova após <i>stemming</i> do Doc1 de treino com DocP1 da contraprova ..	89
4.2.6.5	Contraprova após <i>stemming</i> do Doc2 treino com o DocP2 da contraprova ....	89
4.2.6.6	Contraprova após <i>stemming</i> do Doc3 treino com o DocP3 da contraprova ....	89
4.2.7	Avaliação dos resultados da prospecção .....	90
<b>5</b>	<b>METODOLOGIA CONSOLIDADA.....</b>	<b>92</b>
5.1	Preparação dos corpora.....	93
5.1.1	Etapa 1a: Conversão de PDF para texto puro.....	95
5.1.2	Etapa 1b: PLN e extração dos sintagmas .....	96
5.1.3	Etapa 1c: Tratamento da lista final.....	98
5.1.4	Etapa 2a: Extração dos quantificadores e qualificadores .....	99
5.1.5	Etapa 2b: Conversão do sintagma em seu <i>Stemming</i> .....	100
5.1.6	Etapa 2c: Separação dos <i>stemmings</i> únicos.....	101
5.2	Treinamento do sistema de classificação .....	102
5.2.1	Etapa 3a: Criação do arquivo de treinamento .....	102
5.2.2	Etapa 3b: Treinamento e testes de classificação automática.....	106
<b>6</b>	<b>CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....</b>	<b>113</b>
	<b>REFERÊNCIAS.....</b>	<b>118</b>
	<b>ANEXOS.....</b>	<b>124</b>
	ANEXO A – TEXTOS COMPLETOS.....	124
	ANEXO B - SAÍDAS NO PROCESSAMENTO DE PALAVRAS .....	167
	ANEXO C - MACROS NO EXCEL.....	179
	ANEXO D - LISTA DE SINTAGMAS APÓS A EXTRAÇÃO DOS QUANTIFICADORES DO DOCUMENTO 1 .....	184
	ANEXO E - SINTAGMAS OU SENTENÇAS QUE FORAM REMOVIDAS MANUALMENTE ....	185

ANEXO F - BIBLIOTECA DE COMANDOS DO CLASDOC.....	186
ANEXO G - BIBLIOTECA DE COMANDOS DO PALAVRAS .....	187
ANEXO H - CÓDIGO FONTE DO PROCESSAMENTO PELO SISTEMA CLASSDOC.COM .....	188

## 1 INTRODUÇÃO

De um modo geral todas as mentes são capazes de classificar coisas, quer seja pelas suas semelhanças, quer seja pelas suas diferenças. Classificamos pessoas, coisas, mares, estrelas, pensamentos, amores, notícias, conhecimento e tudo o que conhecemos. Para conhecer, classificamos.

Algumas áreas do conhecimento lidam especificamente com a classificação. Para estas, a classificação é antes de tudo um processo, uma forma de organização para uma posterior recuperação da coisa em si, ou da informação presente nela.

Antes de ser apenas mais um problema de pesquisa, a classificação fundamenta-se também como uma etapa importante no tratamento do volume de documentos que podem ser encontrados no ambiente digital, tanto em acervos controlados, quanto em documentos avulsos dispersos pelas redes de computadores.

Dentre as várias ferramentas que podem ser utilizadas, este enfoque utilizando a linguística e a computação tornou-se importante na medida em que o desenvolvimento de ferramentas e métodos para organização de informações e posterior recuperação é uma área em constante evolução, uma vez que grandes volumes de informação exigem processos tecnológicos de recuperação cada vez mais sofisticados (BAEZA-YATES; RIBEIRO-NETO, 1999).

Os impactos dessa demanda social pela busca por informação podem ser sentidos em diversos segmentos e são potencialmente inspiradoras para a Ciência da Informação (CI), onde novos modelos para recuperação da informação emergem sob uma perspectiva cada vez mais interdisciplinar.

Saracevic (1992) identifica este fenômeno como um imperativo tecnológico aplicado ao campo da CI que pressupõe o desenvolvimento de uma crescente gama de produtos e serviços de informação. Isto acelera o surgimento de novas tecnologias, resultando em diferentes transformações na sociedade e repercutindo em todas as suas esferas.

Saracevic (1996, p. 43) destaca ainda que “problemas informacionais existem há muito tempo e sempre estiveram presentes, mas sua importância real ou percebida mudou e essa mudança foi responsável pelo surgimento da CI [...]”.

Nesta área o trabalho a ser realizado pela CI se apresenta interdisciplinar por natureza, uma vez que para a efetiva realização do processo de disponibilização da informação torna-se necessária uma interação entre diversos campos, o que está de acordo com a visão de Pinheiro (1997) que destaca a questão transformadora em que vive o campo, exigindo um conhecimento em novas disciplinas e subáreas, modificando a própria CI e também proporcionando resultados para as quais ela não poderia dar soluções sozinha.

Esta percepção se alinha ao pensamento de Borko (1968) que coloca a CI como a disciplina que investiga as propriedades e o comportamento da informação, as forças que governam seu fluxo e os meios de processá-la para otimizar sua acessibilidade e seu uso.

Enfatizando ainda mais a natureza interdisciplinar da CI, os dois esquemas, propostos pela American Society for Information Science (ASIS) e pela Information Science Abstracts (ISA), destacam a natureza interdisciplinar da CI. O esquema proposto pela ISA pode ser visto na figura 1.

Figura 1 - Conteúdos da Ciência da Informação pela ASIS

Active and operation -Business and Management- - Educational Activities - Socioeconomic Activities	Building and Facilities	Communications Media	Document Types Availability, Access Content, Purpose
Fields and Disciplines	Hardware, Equipment and Systems	Knowledge Information Knowledge and Information Organization Devices	Natural Functions and Events
Network	Organization	Personal and Informal Groups	Physical Media
Product and Service Providers	Qualities Human Qualities	Research and Analytical Methods	Sociocultural Aspects

Fonte: ZINS (2006, p. 457).

Duas áreas que estão surgindo como fontes de ferramentas a serem utilizadas pela CI são a linguística computacional (LC) e a inteligência artificial (IA), esta última ligada ao desenvolvimento e ao uso de algoritmos que permitem um uso

mais refinado dos recursos computacionais a fim a aumentar a eficácia dos mecanismos de recuperação da informação (RI).

Nestes novos modelos o campo da LC surge como uma candidata a apoiar as diversas etapas da preparação de documentos para recuperação. A LC é a área de conhecimento que explora as relações entre linguística e computação, tornando possível a construção de sistemas com capacidade de reconhecer e produzir informação a partir de textos apresentados em linguagem natural (VIEIRA, 2014).

Uma das áreas de pesquisa da LC é a linguística baseada em corpus, que se utiliza do agrupamento de documentos de texto que, após tornarem-se legíveis para a máquina, podem rapidamente serem pesquisados a fim de obter informações sobre seu conteúdo, trabalho este que envolve tanto um processamento morfosintático e semântico, como também pragmático, etapas fortemente apoiadas pela IA.

Apoiando-se neste contexto onde a CI se aproveita de ferramentas de outras áreas como a computação e a linguística, esta tese propõe a classificação automática de documentos em classes predefinidas (classificação de documentos), proposta que será detalhada a seguir.

### **1.1 Delimitação do problema**

Dentre as pesquisas atualmente desenvolvidas em conjunto pela computação, a CI e a linguística, podemos destacar: reconhecimento de informações semânticas em documentos; desenvolvimento de metalinguagens; criação de novas formas de apresentação dos documentos; desenho de interfaces inteligentes e intuitivas; possibilidade de customização dos sistemas de recuperação da informação; e o uso de sintagmas nominais no lugar de palavras chaves apenas, como comumente utilizado em Sistema de Recuperação da Informação (SRI).

Esta última hipótese baseia-se na viabilidade dos sintagmas como descritores, confirmada por Kuramoto (1996) em sua tese de doutorado. A escolha automática de descritores já foi investigada por alguns pesquisadores, Souza (2005) entre eles. A utilização dos sintagmas para descoberta automática de clusters de documentos também já foi alvo de pesquisas, como a realizada por Maia (2010). Os sintagmas também já foram utilizados de forma a montar uma Estrutura Hierárquica

Temática (EHT) para recuperação da informação, como apresentado por Marco Gonzalez (2014).

A proposta desta tese surge então em um momento oportuno, uma vez que o uso de palavras chaves em sistemas de recuperação da informação já foi alvo de inúmeras pesquisas. Propõe-se um estudo mais aprofundado sobre o sintagma e como ele pode ser utilizado na classificação de documentos, apoiando-se para esta tarefa no uso de Processamento de Linguagem Natural (PLN).

Como dito, em geral as pesquisas existentes baseiam-se em palavras soltas no texto, portanto suscetíveis a uma baixa qualidade da informação processada por uma máquina. A principal hipótese desta tese é a de que é possível elevar a qualidade dos sistemas de classificação de documentos a partir de uma melhor compreensão dos elementos básicos que compõem o corpus para treinamento do sistema.

Apesar de McDonald & Yazdani (1990 apud OTHERO, 2009, pg.79) terem uma frase interessante que diz que: “o grande problema em PLN é [...] que nós ainda não compreendemos completamente o funcionamento da linguagem humana”. Newmeyer (1980 apud OTHERO, 2009, pg.87) destaca que: “aprendemos mais sobre a natureza da linguagem nos últimos 25 anos do que nos 2.500 anos anteriores”, o que gerou avanços em áreas ligadas ao PLN.

Como o texto armazenado em meio digital é o objeto básico tratado pelo PLN, a linguística e a computação tornaram-se fundamentais para que o computador possa tornar-se um instrumento capaz de identificar minimamente os fenômenos da língua, para criar taxonomias que sejam úteis na interação entre computadores e seres humanos.

Visto em um nível mais alto, o PLN tem como objetivo a extração de informações linguísticas que possam favorecer uma ampla gama de tarefas nos sistemas de informações automatizados, tais como a criação automática de catálogos, a sumarização de textos, a recuperação de informações, respostas a perguntas a partir da linguagem natural (sistemas de diálogo), geração de relatórios automáticos, criação de categorias em textos e tradução automática, entre outros.

Um dos grandes trabalhos da PLN é transformar uma sentença de entrada, potencialmente ambígua, em uma forma não ambígua que possa ser usada internamente por um sistema de computador. Estas representações internas variam, é claro, de uma aplicação a outra. A transposição de uma frase potencialmente

ambígua para uma representação interna é conhecida como *parsing* (OBERMEIER, 1987).

Obviamente que isso exige um grande esforço intelectual e computacional, pois não devemos imaginar que a estrutura da língua é simplesmente um amontoado de palavras unidas umas às outras ou sem qualquer regra. Prova disto é a existência de um constituinte, entre a palavra e a frase, chamado sintagma.

Para a língua inglesa existem diversos trabalhos como o de Voutilainen (1993), que criou uma ferramenta para extração de sintagmas chamada NpTool<sup>1</sup>, com regras sintáticas para toda a gramática inglesa. Podemos destacar ainda os trabalhos de Zhai e Evans. Neste último, são utilizadas estatísticas de corpus e heurísticas linguísticas para a extração de sintagmas nominais (SNs), estes trabalhos foram citados por Santos (2005).

Para o português existe o trabalho de Miorelli (2001) que propõe um método, o ED-CER, para a extração de SNs em sentenças em português. Esse método é constituído por duas fases: na primeira fase são selecionados candidatos a SNs; na segunda, um analisador sintático verifica quais candidatos são realmente SNs, tomando como base uma gramática do SN construída a partir da abordagem de Perini (1996).

Uma longa discussão sobre a extração de sintagmas nominais será apresentada no capítulo 3. Ela será importante para a compreensão dos limites atuais da automação dos trabalhos de recuperação dos sintagmas em textos.

## **1.2 Objetivo geral**

Propor um mecanismo para classificação automática de documentos digitais, utilizando os sintagmas nominais como fonte de dados para treinamento de um sistema de classificação supervisionado.

## **1.3 Objetivos específicos**

- Realizar a extração automática de sintagmas em documentos científicos utilizando ferramenta computacional.

---

<sup>1</sup> Acessível no endereço: <<http://www2.lingsoft.fi/doc/nptool/intro/>>.

- Propor as Máquinas de Vetores de Suporte (SVM) como algoritmos de classificação que podem ser treinadas para identificar estruturas léxico-sintáticas.
- Propor um pré-processamento a ser realizado no sintagma, a fim de aumentar o nível de similaridade estrutural entre os sintagmas presentes em documentos de uma mesma área do conhecimento.

\*\*\*

A estrutura da tese é a seguinte:

Nessa introdução foi apresentado um panorama geral da tese, seguido de uma delimitação do problema e dos objetivos gerais e específicos.

No segundo capítulo, denominado **Fundamentos Conceituais**, serão explicitados os principais conceitos relativos ao conjunto teórico-conceitual que permitirá o embasamento da metodologia proposta, fundamentos que irão do mais geral ao mais específico.

No terceiro capítulo será apresentado o **referencial teórico sobre o sintagma**. Apesar de ser parte do fundamento conceitual, optou-se por separar o sintagma em um capítulo à parte, tendo em vista a importância que ele tomou para o desenvolvimento da tese.

No quarto capítulo, intitulado **Experimento de Prospeção**, será apresentado o modelo prático que irá embasar toda a etapa de classificação automática dos documentos. Neste capítulo serão desenvolvidas, passo a passo, as etapas para extração dos sintagmas, bem como o seu uso na recuperação de documentos.

No quinto capítulo, intitulado **Metodologia Consolidada**, será aplicada a metodologia e serão apresentados os resultados obtidos a partir da proposta da tese na classificação automática de um corpus de documentos textuais extraídos de artigos científicos em três áreas do conhecimento.

No sexto capítulo, denominado **Considerações Finais e Trabalhos Futuros**, serão feitas as considerações a respeito dos resultados encontrados e algumas propostas de continuidade da pesquisa.



## 2 FUNDAMENTOS CONCEITUAIS

É inegável a influência social dos processos informacionais na sociedade contemporânea, ao ponto de ser considerada como a *sociedade da informação*, talvez para distingui-la da sociedade industrial de outrora.

Vale destacar a posição de Saracevic, que aponta três características gerais que constituem a CI: interdisciplinaridade, ligação inexorável com a tecnologia de informação e, por último, uma participação ativa e deliberada na evolução da sociedade da informação (SARACEVIC, 1996).

Além dos seus próprios fundamentos conceituais relacionados ao tratamento da informação, que foi o objetivo deste doutorado, a CI também observa o seu objeto sob outros olhares, tais como: a recuperação da informação, a representação da informação, o processamento da informação, os serviços da informação, a comunicação da informação, as tecnologias de informação e comunicação, a produção e recepção da informação, os canais de comunicação, o uso da informação, os estudos da cognição, os estudos dos usuários, as aplicações da inteligência artificial, os estudos ligados à aprendizagem em meio virtual e diversos outros.

Esses aspectos se alinham ao pensamento de Nehmy, para quem esta nova ciência está ligada tanto a uma produção acadêmica quanto ao desenvolvimento de produtos tecnológicos e de técnicas, principalmente na criação de *thesaurus*, na modelagem de bancos de dados, na transposição de documentos textuais e na criação de mecanismos que possam classificar/catalogar documentos impressos ou digitais (NEHMY, 1996).

Neste capítulo serão apresentados os principais fundamentos teóricos necessários à compreensão desta tese de natureza totalmente interdisciplinar, assim como a própria CI.

A seção será dividida em: sistemas de recuperação da informação; classificação de documentos; linguística; PLN e SVM. Sobre a linguística serão apresentados apenas os fundamentos necessários a uma compreensão sobre como os sintagmas nominais podem ser identificados de forma automática em um documento. Obviamente será dado um enfoque maior aos sintagmas nominais, objetos de estudo desta tese, e que haverá um capítulo à parte para melhor detalhamento.

## 2.1 Sistemas de recuperação da informação

As primeiras preocupações com a recuperação da informação já foram identificadas no século II a.C., quando Galeno publicou o *De libris propriis líber*, um catálogo de suas próprias obras visando distinguir as de sua autoria daquelas escritas por outros autores e atribuídas indevidamente a ele (ARAÚJO, 1994).

Lancaster (2004) coloca a recuperação informacional como a atividade mais importante dos centros de informação. Recuperar a informação é um processo que faz parte de um sistema de informação, que seria o centro onde a informação estaria disponível aos usuários.

Segundo Souza (2005), um dos problemas centrais da recuperação de informações em SRIs é a predição de quais documentos são relevantes e quais devem ser descartados. Essa tarefa de "escolha" em sistemas automatizados é executada por algum tipo de algoritmo que, baseado em heurísticas previamente definidas, decide quais documentos a serem recuperados são relevantes e os ordena a partir dos critérios estabelecidos.

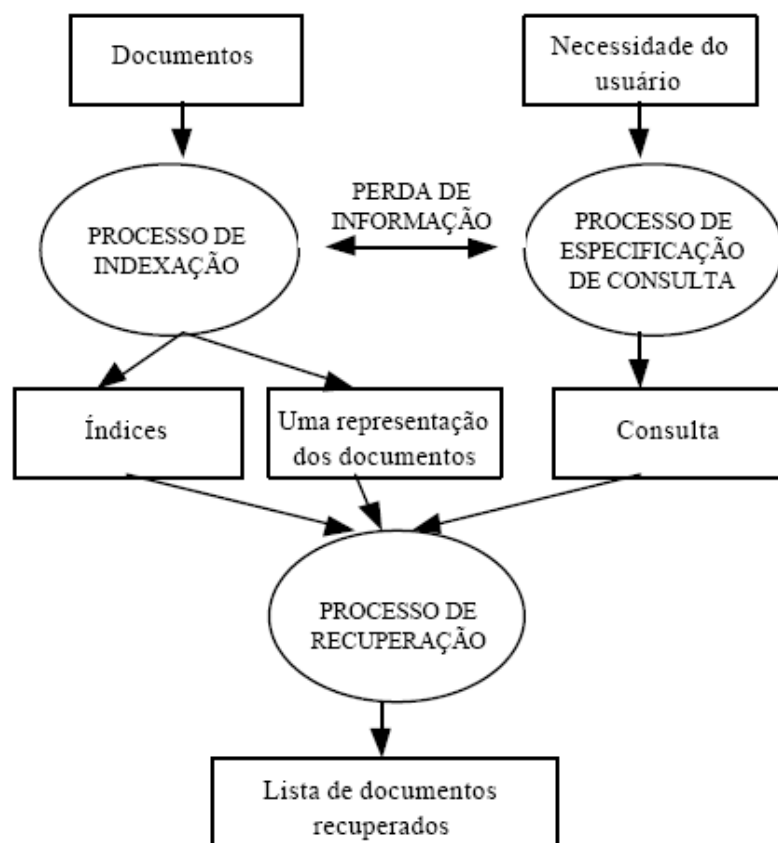
Na visão de Baeza-Yates e Ribeiro-Neto (1999) esse modelo de recuperação da informação também pode ser caracterizado como uma função que determina uma pontuação para a relação de um determinado documento com uma determinada consulta.

Na tentativa de avaliar o desempenho de sistemas de recuperação de informações, Araújo (1994) destaca que um dos grandes problemas é que o grau de relevância não é uma propriedade dicotômica, pois ocorre em graus. Além disso, os SRIs precisam trabalhar com certa imprecisão, principalmente por parte do usuário, pois ambiguidades e indeterminismos não são aspectos cobertos pela lógica de Boole (composta pelos operadores lógicos E, OU, NÃO).

Lancaster (2004) ressalta que a qualidade da estratégia de busca e o vocabulário utilizado pelo usuário são fatores importantes para os processos de busca e recuperação da informação. Segundo ele, é muito importante ter uma boa estratégia de busca, mas se o documento não estiver adequadamente indexado, não obteremos um resultado satisfatório.

Um modelo para recuperação da informação pode ser visto na figura 2:

Figura 2 – Um modelo de SRI

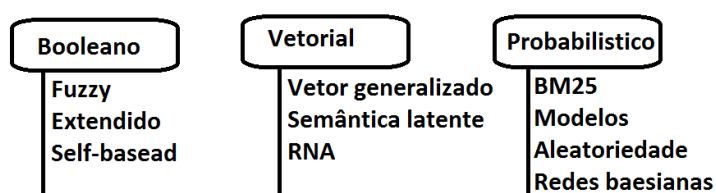


Fonte: Adaptação de GEY (1992), pg. 36.

A avaliação de um SRI pode ser conseguida através de análises de variáveis de revocação e de precisão, onde “precisão” é a fração dos documentos já examinados que são relevantes, e “revocação” é a fração dos documentos relevantes observada dentre os documentos examinados.

Dentre os principais modelos de sistemas para recuperação da informação destacam-se os apresentados na figura 3 e descritos a seguir:

Figura 3 – Modelos de recuperação da informação



Fonte: Adaptado de: BAEZA-YATES; RIBEIRO-NETO, 1999.

**Modelo booleano:** neste caso, ao ser apresentada uma consulta  $Q$  e um conjunto de documentos considerados relevantes para  $Q$ , o índice atribuído aos documentos deve indicar qual documento é mais importante, estabelecendo uma ordem de relevância. Esses índices são calculados com base na comparação entre a consulta e os documentos.

Uma maneira direta de implementar o modelo booleano seria assumir a existência de uma lista invertida em que cada entrada correspondesse a um termo de indexação. Ademais, uma entrada nesta lista irá apontar para um ou mais documentos onde o termo ocorre. O conjunto de documentos recuperados pode ser obtido pela interseção das listas invertidas de documentos que aparecem na consulta. Assim, somente documentos cujos termos de indexação satisfazem a consulta booleana são recuperados.

**Modelo vetorial:** o modelo de espaço vetorial, ou simplesmente modelo vetorial, representa documentos e consultas como vetores de termos. Os documentos devolvidos como resultado de uma consulta são representados similarmente, ou seja, o vetor resultante de uma consulta é produzido através de um cálculo de similaridade no vetor de espaço  $n$ -dimensional.

**Modelo probabilístico:** o modelo probabilístico descreve os documentos considerando pesos binários que representam a presença ou ausência de termos. O vetor resultante gerado pelo modelo tem como base o cálculo da probabilidade de que um documento seja relevante para uma consulta.

## 2.2 Classificação de documentos

De uma maneira geral, classificar é estar consciente de algo, representando-o e exercendo sobre o objeto uma discriminação em relação aos demais objetos possíveis.

Enquanto processo mental, a classificação é o meio pelo qual o ser humano pode distinguir as coisas por meio da identificação de suas diferenças ou semelhanças, estabelecendo relações e agrupando-as em classes de acordo com as relações encontradas. Além disso, a classificação pode ser realizada por uma dentre várias características dos objetos, o que pode gerar diferentes arranjos para eles, de acordo com as razões existentes, ao escolher uma dentre várias

possibilidades de classificação. A classificação torna-se, pois, a arte de colocar diversas coisas desordenadas em um todo ordenado (TRISTÃO, 2004).

Em seu uso prático nas diversas áreas do conhecimento, a classificação pode gerar, como resultado, três importantes produtos: 1) o esquema de classificação; 2) o esquema de notação; e 3) o índice para facilitar a recuperação do objeto. Por sua vez, o esquema de classificação pode ser subdividido em classes, que em geral se apresentam como um modelo que propõe uma visão de um nível mais geral a outro mais específico.

Jacob e Shaw (1998, p. 155) apresentam a classificação como sendo “um processo cognitivo de dividir as experiências do mundo em grupos de entidades ou categorias, para construir uma ordem física [...] facilitando o armazenamento e a posterior recuperação da informação”. Os humanos fazem isso o tempo todo e de forma automática (JACOB; SHAW, 1998).

O ofício de classificar não é algo recente. Alguns filósofos na idade antiga se ocupavam em estabelecer grandes agrupamentos segundo conceitos. Na obra *Organon*, Aristóteles (1987) concebeu a categorização como um processo mental e estabeleceu dez categorias, a saber: substância, qualidade, quantidade, relação, lugar, tempo, situação, posse, ação, sofrimento ou passividade.

Já Francis Bacon, em 1214, na obra *De Augmentis Scientiarum* sugeriu um esquema de classificação que foi considerado claro e satisfatório para abarcar todo o estudo do conhecimento humano. O sistema proposto por Bacon foi um dos mais influentes nas diversas outras formas posteriores de classificar o conhecimento.

Segundo Kaula (2011), os principais esquemas Escolásticos foram os propostos por Locke; Comte; Coleridge; Spencer; Bain; Pearson; e E. C. Richardson.

Outras categorias de esquemas de classificação foram desenvolvidas, como o de Linnaeus, a classificação botânica de Jussieu, Hooker e Bentham e ficaram conhecidos como esquemas para se classificar as ciências.

Muitos esquemas se baseiam em uma notação, usados principalmente para organizar coleções físicas de objetos, como livros, por exemplo. Dentre eles temos: a classificação de Harris; Schwartz; a classificação de Melvil Dewey; a classificação expansiva proposta por Cutter; a classificação racional de Perkins, Hartwig, Fletcher, Bonazzi e Rowell; a classificação ajustável de Brown; a classificação científica Internacional de 1901; a classificação proposta pela biblioteca do congresso americano; a classificação dos dois pontos proposta por Ranganathan em 1933; a

classificação de Bliss em 1935; e a de Rider em 1961. Além dos supracitados, surgiram também alguns esquemas de classificação aplicáveis a assuntos especializados.

A partir dos esquemas propostos é possível chegar a uma definição importante: os filósofos se propuseram a criar esquemas para classificar o conhecimento. Os cientistas, por outro lado, criaram mecanismos de classificação para organizar bibliotecas e acervos, esquemas baseados na classificação por assunto ou conceito.

Uma disciplina que se ocupa primordialmente em classificar documentos é a biblioteconomia. Dentre as metodologias criadas pela biblioteconomia se destacam a CDD e a CDU. Do ponto de vista de finalidade, as classificações CDD e CDU são classificações documentárias, utilizadas para organizar documentos em bibliotecas, com a finalidade de recuperar a informação contida nestes documentos, sendo que a CDD surgiu necessariamente para ser utilizada em bibliotecas, já a CDU surgiu para o uso bibliográfico.

A CDD foi criada por Melvin Dewey, com base na classificação de Harris, foi a primeira classificação bibliográfica de que se tem conhecimento, pois utiliza números arábicos. Atualmente é a classificação mais utilizada no mundo, editado em várias línguas, mas suas línguas oficiais são o espanhol e o inglês. Na sua primeira edição, ela iniciou com 42 páginas e recebeu o nome em inglês de *A Classification and subject index for cataloging and arranging the book and pamphlets of a library*, e passou a se chamar de Classificação Decimal Dewey a partir da décima sexta edição.

Já a classificação CDU foi criada pelos belgas Paul Otlet e Henri de La Fontaine em 1892, baseada na CDD, tendo sido publicada em vários idiomas, sendo o Inglês o seu idioma principal.

A CDU faz parte de uma autorização de Dewey para a expansão da CDD, na sua primeira edição teve o nome de *Manuel de Repertoire Bibliographique Universel*, e possuía 33.000 entradas. Antigamente era conhecida como classificação de Bruxelas, mas na segunda edição passou a se chamar, em francês, de *Classification Decimale Universelle*. Ela está sendo traduzida em muitas línguas, sendo que no Brasil, o IBICT publicou a primeira Edição-Padrão Internacional em língua Portuguesa em 1997.

Segundo Lidia Alvarenga o que na realidade se classifica em uma biblioteca tradicional ou digital não são os documentos, mas os conceitos contidos nesses documentos. Deve-se ressaltar que não somente os textos são formados de conceitos, mas também as imagens fixas e em movimento e os sons, fato que vem dificultar sobremaneira os projetos de classificação automática de documentos na web. (ALVARENGA, 2001).

Podemos entender por teoria do conceito o conjunto de reflexões pertinentes à complexa região epistemológica interdisciplinar que compreende o ato de representação, comunicação e preservação de objetos e pensamentos e cujo conhecimento integra os campos da linguagem. O conceito e a compreensão do que seja conceito é tema crucial, por pertencer à essência do trabalho de tratamento e organização da informação, compreendendo os processos de análise de assunto, classificação e recuperação da informação. (ALVARENGA, 2001)

Os documentos no meio digital colocam em cheque a tradicional classificação manual realizada pela biblioteconomia que se baseia em assuntos, partindo do mais geral para o mais específico. A explosão documental trouxe consigo uma série de problemas para a recuperação da informação, dentre eles uma possível falta de uma classificação sistemática dos documentos disponibilizados e a consequente não recuperação do documento. Portanto, esquemas automáticos de classificação passaram a ser alvo de pesquisas em inúmeras áreas.

Várias são as pesquisas que pretendem melhorar a performance dos algoritmos de classificação automática de documentos. Entretanto, são poucas as que se concentram na busca por estruturas que permitam uma melhor representação dos dados que serão utilizados por estes algoritmos. Como foi mencionado anteriormente, estes são o objetivo e a contribuição almejados por esta tese: utilizar a promissora estrutura dos sintagmas nominais em conjunto com os conhecimentos atuais sobre aprendizagem de máquina, mais especificamente utilizando o algoritmo Support Vector Machines (SVM), como proposto por Vapnik (1995), que pode ser utilizado nos mais diversos trabalhos de classificação em geral. Ele foi utilizado nesta tese na tentativa de criar um ambiente que seja mais eficiente na classificação automática de documentos a partir do seu conteúdo, contribuindo desta maneira para áreas recentes de pesquisa, tais como a web semântica.

Algoritmos de computadores são constituídos por uma sequência de passos desenvolvidos em alguma linguagem computacional, que permite aos computadores

resolver problemas que antes seriam resolvidos por um humano. Recentes avanços na computação permitiram o desenvolvimento de algoritmos que pudessem simular processos mentais. Esses algoritmos são conhecidos por formarem a base para o aprendizado de máquina na Inteligência artificial.

Nesse ponto vale a pena destacar que os sistemas baseados em conhecimento podem ser vistos como “conectores semânticos”, recebendo informações de diversas origens e sendo capazes de analisá-las, interpretá-las, identificando a sua relevância e estando aptos a direcionar soluções de acordo com interesses dinâmicos, ao contrário das formas tradicionais de computação (MARTINS, 2012).

O conjunto de sintagmas extraídos de um corpus será a base para a metodologia de classificação automática dos documentos proposta nesta tese. A parte algorítmica de classificação automática proposta está baseada no trabalho de Manning et al. (2008), que propõem uma maneira geral de classificação digital da informação que pretende descrever os documentos através de classes em que:  $d \in \mathcal{D}$ , onde  $\mathcal{D}$  é um espaço de documentos e  $d$  é um documento no acervo digital. Estes espaços de documentos estariam contidos em classes,  $C=\{c_1, c_2 \dots c_n\}$ , que comporiam um conjunto de rótulos predefinidos. Para representar essa associação entre documento e classe teremos:

$$\langle d, c \rangle = \langle \text{Brasília a capital projetada por Niemeyer, Brasil} \rangle$$

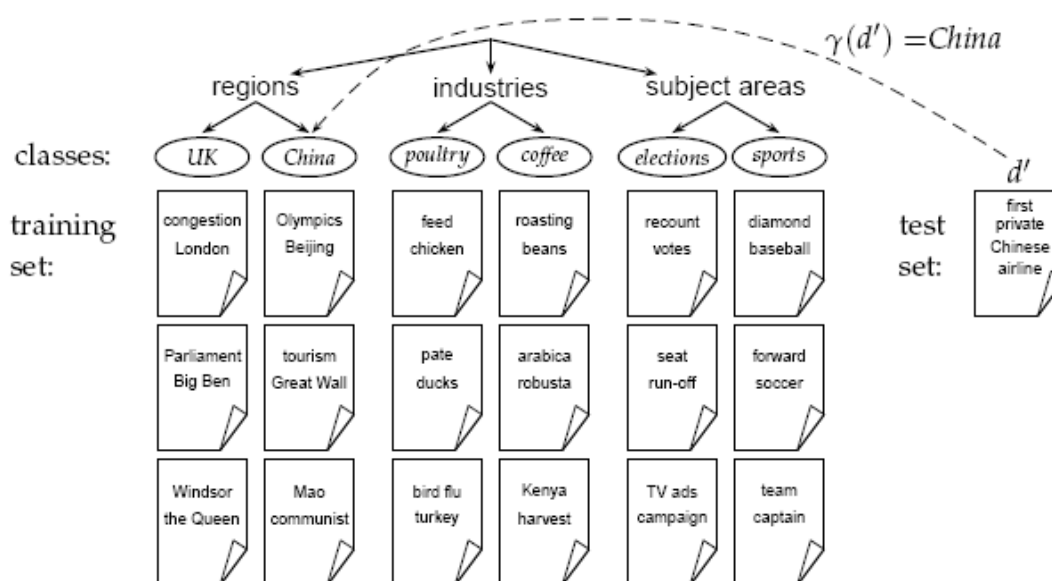
O papel do usuário supervisor, que treinaria o sistema para que ele pudesse aprender a classificar, seria o de determinar a função de associação  $\gamma$  entre os documentos e as classes.

$$\gamma = \mathcal{D} \rightarrow C$$

Esse tipo de associação que depende de um supervisor é denominado aprendizado *supervisionado*, pois o algoritmo irá depender de um professor para lhe apresentar as classes e os documentos de exemplo para cada uma delas, como é mostrado na figura 4, que traz uma estrutura hierárquica de classificação. No primeiro nível temos as classes  $C: \{\text{regions (UK, China), industries (poultry, coffee), subject areas (elections, sports)}\}$ , e no terceiro nível os documentos propriamente ditos.



Figura 4 – Documentos para treinamento e para teste



Fonte: Manning et al. (2008).

O principal objetivo ao treinar um software para classificar documentos é que ele seja capaz de classificar um documento desconhecido, submetido posteriormente, a partir dos documentos separados por classes. Exemplos de algoritmos de aprendizado para classificação são descritos por Lopes (2004): árvores de decisão, conjuntos de regras, classificadores baseados em exemplos, classificadores probabilísticos, Support Vector Machines e Redes Neurais Artificiais.

## 2.3 Linguística

A grande maioria dos dados de que o homem dispõe está sob a forma textual, portanto torna-se importante pensar sobre “a palavra”, mesmo que recoberta por inúmeras dificuldades<sup>2</sup>.

Há, aparentemente, uma instância entre sentido e intelecto, que transforma dado em palavra. O intelecto *stricto sensu* é uma

<sup>2</sup> Resulta da celebre sentença “o significado de uma palavra é seu uso na linguagem” na qual o ponto de vista de Wittgenstein permite uma abertura à análise dos problemas da linguística a partir de um corpus e do processamento da linguagem natural. A descrição de uma língua, segundo o filósofo, estará sempre condicionada à parcialidade imposta pelas infindáveis possibilidades de lances dos jogos de linguagem e, portanto, também passíveis de uma análise estatística.

tecelagem que usa palavras como fios. O intelecto *lato sensu* tem uma antessala na qual uma fiação que transforma algodão bruto (dados dos sentidos) em fios (palavras). (FLUSSER, 1963).

Na prática, a transmissão da informação utilizando a linguagem natural chega até as pessoas de forma organizada por intermédio dos seus sentidos, agrupadas de acordo com regras que formam frases e atuando como um elemento fundamental na comunicação social. Portanto, uma coisa a se destacar é que o documento escrito não encerra totalmente as possibilidades da língua, pois a língua só se realiza plenamente no conjunto entre falante e ouvinte. Não sendo apenas uma sucessão de palavras é, antes de tudo, uma estrutura de valores e formas, que são construções sociais e determinam as condições, desde a produção até a transferência da informação.

Formalmente a linguagem compreende os elementos fonológico (ligado à pronúncia), morfológico (relacionado à forma), sintático (relacionado às sentenças) e semântico (relacionado ao sentido de determinada estrutura).

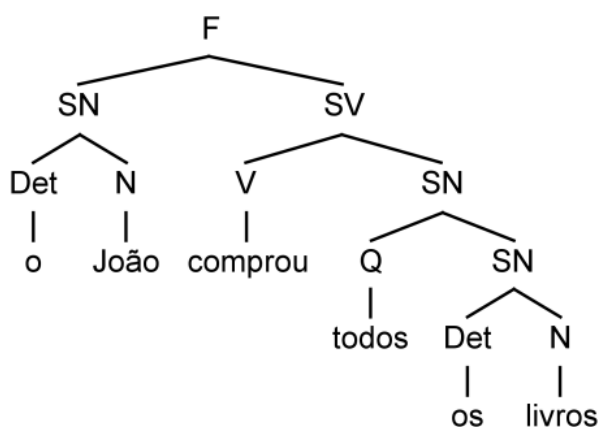
A importância de todas essas áreas ligadas ao estudo da linguagem fica evidente quando os sistemas de recuperação da informação precisam tratar grandes volumes de documentos, uma vez que muitos trabalhos deixam de ser manuais e passam a ser realizados de forma automática ou semiautomática. Abaixo são apresentadas a morfologia e a sintaxe, indispensáveis à pesquisa apresentada.

Falando sobre a *morfologia*, que trata especificamente da estrutura e da formação das palavras, estas estruturas são conhecidas como morfemas, que podem ser independentes como na palavra *concebido*, ou dependentes como no caso dos sufixos (ex.: o “pré” da palavra *preconcebido*). O morfema então pode ser compreendido como sendo a menor unidade gramatical, uma vez que ele pode ser utilizado na composição de outras palavras, modificando ou colocando-se em oposição à palavra inicial.

A morfologia também se ocupa da classificação sintática das palavras, estudo esse denominado pelos linguistas como partes do discurso (*part-of-speech*), compostos por classes como substantivos, advérbios, preposições, adjetivos etc. Segundo Vieira (2001), as palavras de determinada categoria podem ser usadas como base de um determinado grupo (ou sintagma). Tais palavras são chamadas de núcleo e identificam o tipo de objeto ou conceito que o sintagma descreve. Por exemplo: os sintagmas nominais possuem como núcleo um substantivo (ou nome);

nas frases “o cachorro”, “o cachorro raivoso” e “o cachorro raivoso do canil”, temos sintagmas nominais que descrevem o mesmo tipo de objeto. Da mesma forma, os sintagmas adjetivais *faminto*, *muito faminto*, *faminto como um cavalo*, descrevem um mesmo tipo de qualidade (VIEIRA, 2011). A Figura 5 apresenta um esquema simplificado com a extração dos sintagmas na frase: *O João comprou todos os livros*.

Figura 5 – Exemplo de uma visualização hierárquica dos sintagmas nominais extraídos de uma frase em português



Fonte: SOUZA (2005)

No caso de um sistema automatizado, os aspectos morfológicos são tratados por um programa de computador conhecido genericamente como etiquetador de categorias gramaticais (*POS taggers*). Como resultado do processamento de um determinado texto, o software seria capaz de devolver o mesmo texto no qual todas as palavras teriam sido etiquetadas com sua categoria gramatical.

Outro aspecto a ser considerado é a sintaxe. A avaliação sintática de uma sentença permite classificá-la como um todo, processamento esse realizado após uma análise morfológica. Assim como as palavras, as frases seguem determinadas regras de formação, algo importante para que sejamos capazes de reconhecer até mesmo frases que nunca escutamos antes.

A partir da identificação da estrutura de uma frase é possível dizer, por exemplo, qual ação está sendo realizada pelo sujeito, a concordância, o gênero, o número ou o grau entre as palavras.

Sistemas informatizados capazes de realizar o processamento da sintaxe são conhecidos como analisadores sintáticos ou *parsers*, em inglês. Estes *softwares* identificam estruturas válidas a partir de um léxico definido para uma determinada língua (SOUZA, 2005).

Vale ressaltar que os estudos realizados sobre o sintagma nesta tese estão baseados na língua portuguesa escrita no Brasil. Ao comparar as diversas línguas é facilmente perceptível a diferença na estruturação dos sintagmas.

## 2.4 Processamento da linguagem natural

O processamento da linguagem natural ou PLN teve início como uma disciplina autônoma em meados da década de 50 e incorporou rapidamente ferramentas e técnicas da inteligência artificial, da ciência da computação e da linguística propriamente dita, tendo como objetivo a compreensão da linguagem humana em computadores, através de algoritmos criados para esse fim.

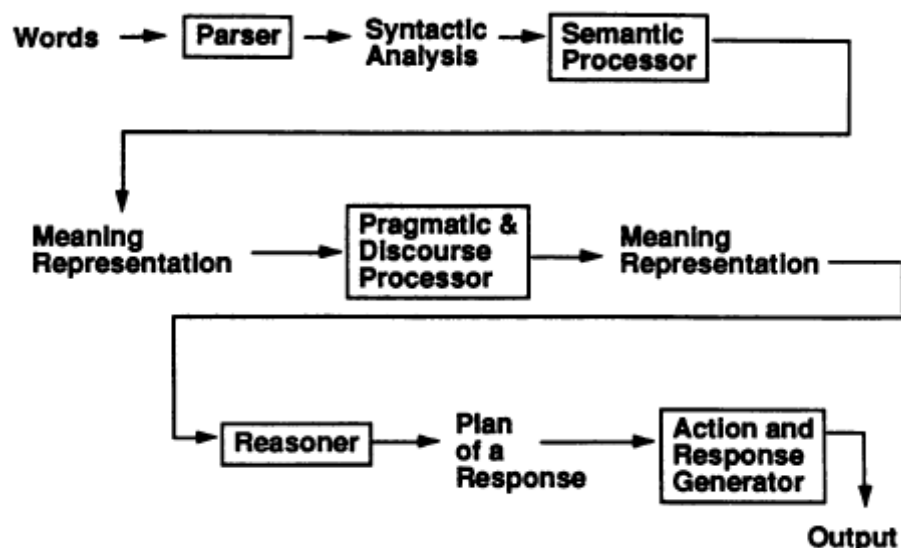
No entanto, apenas recentemente as pesquisas saíram dos laboratórios e passaram a compor sistemas de informações automatizados como tradutores, sistemas de perguntas e respostas, comandos de voz em automóveis e dispositivos móveis.

Atualmente existem inúmeras visões<sup>3</sup> relativas ao campo do PLN, o que decorre do fato de existirem inúmeros produtos que podem ser gerados ou que podem em algum momento fazer uso dele. Genericamente, qualquer modelo possui uma ou várias das etapas apresentadas na Figura 6.

---

<sup>3</sup> ALLEN, J. **Natural Language Understanding**. Menlo Park: Benjamin/Cummings, 1987.  
 BATES, M.; WEISCHEDEL, R. M. (ed.). **Challenges in Natural Language Processing**. Cambridge: Cambridge Univ. Press, 1993.  
 GAZDAR, G.; MELLISH, C. **Natural Language Processing in {LISP}**. Reading: Addison-Wesley, 1989.  
 SMITH, G. W. **Computers and Human Language**. Oxford: Oxford Univ. Press, 1991.  
 WINOGRAD, T. **Language as a Cognitive Process**. Reading: Addison- Wesley, 1983.

Figura 6 – Pipeline de um sistema para PLN genérico



Fonte: BATES (1993).

A maioria dos sistemas de PLN tem algum tipo de pré-processador que faz análise morfológica, realiza consultas a dicionários, realiza substituições lexicais (para normalizar as abreviaturas, por exemplo) e faz atribuição do discurso. A ordem em que estes processos são realizados, as técnicas utilizadas para realizá-las e os formatos do resultado são algo altamente idiossincrático (BATES, 1993).

Conforme Covington (1997), a pesquisa em PLN está voltada, essencialmente, a três aspectos da comunicação em linguagem natural, a saber: fonologia, morfologia e sintaxe, e semântica e pragmática.

No que diz respeito à identificação do sintagma nominal utilizando os mecanismos de PLN, várias são as contribuições de instituições e de pesquisadores individuais. Dentre os principais trabalhos podemos destacar: Miorelli (2001), Othero (2004), Santos (2005), Arcoverde (2007), Costa (2007), David (2007) e Lopes (2009).

Miorelli (2001), no Departamento de Ciência da Computação da Pontifícia Universidade Católica do Rio Grande do Sul, construiu um método chamado ED-CER para extração de SNs aplicando as regras estruturais de sintagmas nominais de Perini (1996). Com esse trabalho, a autora objetivou a formalização de um método para extração de SNs em língua portuguesa, aplicando ferramentas do PLN no intuito de possibilitar uma maior funcionalidade dos SRIs. Assim, o trabalho tem como meta encontrar palavras-chave ou expressões-chave para representar os

conteúdos em formato digital dos resumos das dissertações do Programa de Pós-Graduação em Ciência da Computação da Pontifícia Universidade Católica do Rio Grande do Sul, que constituem o corpus da pesquisa (SILVA, 2014).

Othero (2004a) desenvolveu um *parser*, na área de Letras, que faz análise de sentenças do português, o *Grammar Play*. O tipo de frase analisada por essa ferramenta é a declarativa que contenha um único verbo e que não esteja na forma interrogativa, salvo quando estiver estruturada por meio dos pronomes de interrogação “QU” (que, qual) (SILVA, 2014).

Santos (2005), na área de Ciência da Computação, apresentou uma nova abordagem de molde de regras para o TBL (Aprendizado Baseado em Transformações): o termo atômico com restrição (TA com restrição), que consiste na verificação de uma possível dependência entre a preposição a ser classificada e o verbo precedente. Dessa maneira, geram-se regras específicas para observar a relação entre essa preposição e o verbo antecedente. É importante que isso seja observado porque muitos SNs, estando inseridos dentro de um sintagma preposicional, estão na função de objeto e são precedidos por uma preposição (SILVA, 2014).

Arcoverde (2007), com o objetivo de representar textos, constrói um modelo híbrido que utiliza dois tipos de conhecimento, o linguístico e o estatístico. Esses tipos de conhecimento, aplicados a sistemas de RI, possibilitam um processo de pós-filtragem de informação. Na construção do modelo, o autor usa a Categorização de Textos integrada a um sistema de RI, em que recursos de PLN proporcionam experiências na busca de informações relevantes ao usuário (SILVA, 2014).

Costa (2007), com seu projeto na área de Ciência da Computação, apresenta uma gramática computacional para o português, chamada de LXGram30, em desenvolvimento na Universidade de Lisboa. Os objetivos do LXGram são analisar frases, no intuito de produzir uma descrição formal do seu significado, e gerar frases a partir de representações desse significado. O autor foca na modelagem e na implementação da sintaxe e da semântica de SN da língua portuguesa (SILVA, 2014).

David (2007) desenvolve um programa de computador, no Programa de Pós-Graduação em Linguística da Universidade Federal do Ceará, que atribui a estas expressões sua estrutura de constituintes e sua representação, por meio de colchetes rotulados e de árvores, com o objetivo de analisar expressões nominais da

língua portuguesa. Outro objetivo expresso pela autora é testar a hipótese de que sintagmas determinantes têm como núcleo pronomes pessoais.

Lopes (2011), do Programa de Pós Graduação em Ciência da Computação da Pontifícia Universidade Católica do Rio Grande do Sul, tem o objetivo de extrair automaticamente conceitos para um domínio caracterizado por um corpus em língua portuguesa. Para tanto, a autora define um método de extração de termos candidatos a conceitos a partir de um corpus marcado linguisticamente, ordena os termos extraídos segundo sua relevância, identifica, dentre os termos extraídos, quais devem ser considerados conceitos do domínio, e constrói um conjunto de recursos linguísticos a partir desses conceitos, facilitando a sua compreensão, manipulação e visualização.

## **2.5 Aprendizagem de máquina e Support Vector Machines (SVMs)**

As técnicas de aprendizagem de máquina (AM), dentre elas as SVMs, empregam um princípio de inferência denominado indução, em que são obtidas conclusões genéricas a partir de um conjunto particular de exemplos. O aprendizado indutivo pode ser dividido em dois tipos principais: supervisionado e não supervisionado. No aprendizado supervisionado há um professor externo, que apresenta o conhecimento do ambiente por meio de conjuntos de exemplos na forma: entrada, saída desejada. O algoritmo de AM extrai a representação do conhecimento a partir desses exemplos. O objetivo é que a representação gerada seja capaz de produzir saídas corretas para novas entradas não apresentadas previamente.

Um requisito importante para as técnicas de AM é que elas sejam capazes de lidar com dados imperfeitos, denominados “ruídos”. Muitos conjuntos de dados apresentam esse tipo de caso, sendo alguns erros comuns à presença de dados com rótulos e/ou atributos incorretos. A técnica de AM deve idealmente ser robusta a ruídos presentes nos dados, procurando não fixar a obtenção dos classificadores sobre esse tipo de caso. Deve-se também minimizar a influência de *outliers* no processo de indução. Os *outliers* são exemplos muito distintos dos presentes no conjunto de dados. Esses dados podem ser ruídos ou casos muito particulares, raramente presentes no domínio.

*Support Vector Machine* é uma máquina de aprendizagem baseada na minimização do risco estrutural. Seu processo de aprendizado é do tipo supervisionado e ela pode ser usada na resolução de problemas de classificação e de regressão. Em um contexto de classificação binária, por exemplo, a ideia principal da SVM é construir um hiperplano como superfície de separação ótima entre exemplos positivos e exemplos negativos (HAYKIN, 1999). A acurácia das SVMs na tarefa de reconhecimento de padrões é comprovada por inúmeros trabalhos na literatura (VAPNIK, 1995).

As SVMs são baseadas no princípio da utilização de um algoritmo de aprendizagem que realiza a maximização da margem entre as classes de dados do problema. Dessa forma, as SVMs obtêm boa capacidade de generalização.

As SVMs são baseadas no tipo de aprendizado chamado supervisionado, ou aprendizado com professor (especialista no conhecimento a ser aprendido pela máquina). Esse aprendizado supervisionado consiste de três componentes (VAPNIK, 1992; 1998 apud HAYKIN, 1999): 1) Ambiente: conjunto de vetores de entrada  $x$ ; 2) Professor: o professor fornece uma resposta para cada vetor de entrada  $x$  recebido de acordo com uma função  $f(x)$  desconhecida; 3) Máquina: ou algoritmo de aprendizagem, que é capaz de implementar funções de mapeamento de entrada-saída da forma  $y = f(x, r)$  onde  $y$  é a resposta produzida pela máquina e  $r$  é um conjunto de parâmetros usados como pesos aos valores do vetor  $x$ . (COUTINHO, 2007)

Os dados do conjunto de treinamento devem ser estatisticamente representativos, para que a máquina possa reconhecer possíveis padrões posteriores não apresentados a ela inicialmente. Esta é uma propriedade conhecida como Generalização (VAPNIK, 1995). Além desta grande amostra de dados, é necessário também que as funções  $d = F(x, r)$  tenham comportamento determinístico, ou seja, para um certo conjunto de entrada  $x$  e um conjunto de parâmetros  $r$ , a saída deve ser sempre a mesma (COUTINHO, 2007).

As Support Vector Machine foram propostas inicialmente como ferramenta de classificação binária. Porém muitos dos problemas reais possuem características multiclasse. Para que fosse possível a utilização da SVM nesse tipo de aplicação, foram propostos alguns procedimentos para estender a SVM binária.



As principais abordagens utilizam como base a decomposição de um problema multiclasse com  $k$  classes,  $k > 2$ , em  $k$  problemas binários, destacando os métodos de decomposição Um-Contra-Todos e Todos-Contra-Todos.

A proposta inicial da SVM era a aplicação em problemas de classificação, sendo o principal enfoque dessa técnica. Ela ainda pode ser aplicada a problemas de regressão (HAYKIN, 1999).

Devido ao fato de sua eficiência em problemas de alta dimensionalidade, a SVM vem obtendo grande sucesso em aplicações de visão computacional, que buscam extrair informações a partir de imagens. Também são aplicadas em bioinformática (MA; HUANG, 2008) e em classificação textual (TONG; KOLLER, 2000).

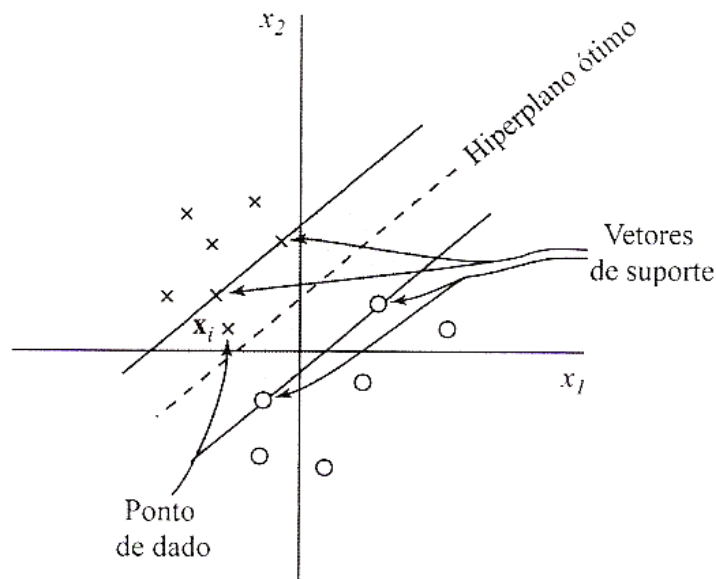
Vapnik (1995) define que uma máquina de aprendizado, após a observação de vários pares de entrada  $X$  e de saída  $Y$ , deve ser capaz de: 1) imitar o comportamento do sistema, gerando, a partir de dados de entrada próximos a  $X$ , saídas próximas a  $Y$  (esta tarefa é conhecida como classificação, pois o domínio de  $X$  é dividido de acordo com o número de saídas  $Y$  possíveis no sistema); 2) descobrir a função que governa o sistema. Nesse caso, o número de saídas  $Y$  é infinito e esta tarefa é conhecida como regressão.

Para que uma máquina seja capaz de realizar uma das tarefas acima, ela deve ser submetida a um processo de aprendizagem, também conhecido como processo de treinamento. Existem diversas maneiras de desenvolver um programa que torne o computador capaz de aprender, as SVM se baseiam na teoria do aprendizado estatístico.

Como dito anteriormente, o objetivo da máquina de aprendizado é escolher uma função  $f(x,r)$  capaz de mapear a relação entre  $x$  e  $y$ , onde  $r$  são os parâmetros desta relação. As funções usadas para aprender este mapeamento são conhecidas como funções indicadoras, em problemas de classificação e de funções de aproximação, e em problemas de regressão (VAPNIK, 1999).

O hiperplano de separação encontrado por uma SVM, mostrado na Figura 7, é ótimo, pois só há um ponto que minimiza a função de custo quadrática existente no problema de otimização característico da SVM.

Figura 7 – Ilustração de uma superfície ótima de separação entre duas classes



Fonte: CARVALHO, 2005.

Para escolher a função que melhor se ajuste ao conjunto de treinamento é necessária uma medida de discrepância  $L(y, f(x, r))$ , que indica a diferença da saída desejada  $d$  da saída obtida  $y$  (CARVALHO, 2005). Para problemas de classificação binária, com somente duas classes, são usadas funções de discrepância como:

Figura 8 – Medida de discrepância entre a saída desejada e a obtida

$$L(y, f(x, r)) = \begin{cases} 0, & \text{se } y = f(x, r) \\ 1, & \text{se } y \neq f(x, r) \end{cases}$$

Fonte: CARVALHO, 2005.

Neste trabalho foi usada uma implementação de Support Vector Machine chamada SVM-Multiclass<sup>4</sup>, desenvolvida por Thorsten Joachims, professor do Departamento de Ciência da Computação da Universidade de Cornell. Essa

<sup>4</sup> Disponível no site <[http://www.cs.cornell.edu/People/tj/svm%5Flight/svm\\_multiclass.html](http://www.cs.cornell.edu/People/tj/svm%5Flight/svm_multiclass.html)>.

implementação foi escolhida porque trabalha com problemas de classificação com uma ou mais classes.

A SVM-Multiclass utiliza como entrada em ambos os módulos, de treinamento e de teste, um arquivo de texto onde cada linha corresponde a um vetor de dados no seguinte formato:

*<classe> <característica>:<valor> <característica>:<valor>...*

Onde a classe é um número usado para identificar cada palavra; a característica corresponde ao valor 1 ou 0, representando a presença ou a ausência daquele sintagma no documento em questão.

A execução dos módulos da SVM-Multiclass é feita através de linhas de comando, para o módulo de treinamento esta linha é escrita da seguinte forma:

*svm\_multiclass\_learn -c <C> <arquivo de entrada> <arquivo de saída>*

A linha de comando para a execução do módulo de classificação é a seguinte:

*svm\_multiclass\_classify <arq. de entrada 1> <arq. de entrada 2> <arq. de saída>*

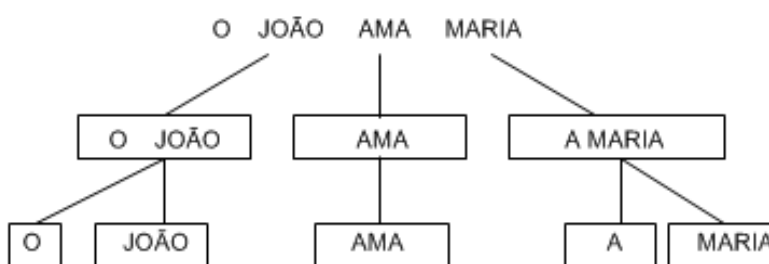
Onde arquivo de entrada 1 é o arquivo com os vetores que serão classificados, arquivo de entrada 2 é o arquivo de saída do módulo de treinamento e arquivo de saída é um arquivo onde cada linha representa um vetor do arquivo de classificação e contém o número da classe à qual a SVM considera que o vetor pertence, ou seja, o resultado da classificação para o conjunto de teste.

Durante a seção “Metodologia Consolidada” esse processo será melhor detalhado.

### 3 O SINTAGMA NOMINAL

Ao estudarmos a estrutura sintática de uma sentença, podemos observar que ela atende a determinadas regras. Não podemos analisar uma sentença como sendo apenas um amontoado de palavras, pois é justamente a posição dessas palavras dentro da sentença que irá lhe conferir algum sentido. Há, entre os níveis de palavra e de sentença, uma organização denominada *sintagma* (OTHERO, 2009, p. 21). Vejamos o exemplo de Souza:

Figura 9 – Um exemplo de SN em níveis



Fonte: SOUZA (2005)

Ou seja, a sentença acima não foi escrita sem nenhum critério. Para ter alguma significação ela atende a regras sintáticas rígidas, sob pena de perder totalmente o sentido para o ouvinte. Essa organização se dá ao nível sintagmático.

Ao tentar compreender o sintagma, estamos concordando com Sag, Wasow e Bender (OTHERO, 2009, p. 16) ao afirmarem que:

Desenvolver tecnologias realmente robustas que lidem com a linguagem natural requer análises detalhadas e cuidadosas sobre a estrutura gramatical da língua e como ela influencia o significado. Atalhos que se baseiem em heurísticas, tentativas de adivinhação ou simples usos de templates irão inevitavelmente levar a erros (OTHERO, 2009, pg.67).

O SN é um conjunto de elementos que constituem uma unidade significativa dentro da oração e que mantêm a dependência e a ordem entre seus constituintes. Eles se organizam em torno de um elemento fundamental denominado núcleo do sintagma, que pode ele mesmo ser um sintagma.

Perini et al. (1996) definem o SN como uma classe gramatical que tem um comportamento de sujeito da oração, de objeto direto e, se precedido de um predicado, de adjunto adnominal ou de objeto direto, o que concorda com a tese de Liberato (1997) quando diz que o SN representa os referentes ou os conceitos de uma oração.

O tipo de sintagma tratado nesta tese será o sintagma nominal. Entretanto, existem os sintagmas verbais (SV), os sintagmas adjetivais (SA) e os sintagmas preposicionados (SP), que são normalmente formados por uma preposição + sintagma nominal.

Geralmente na estrutura de uma oração, em sua forma base, aparecem como constituintes obrigatórios o SN e o SV, como nos exemplos:

*[Os garotos] {empinavam papagaios de papel}*

SN                      SV

*[Nós]                      {assistimos a uma conferência sobre tóxicos}.*

SN                      SV

Segundo Perini, o SN pode aparecer de acordo com algumas estruturas básicas, são elas (SOUZA, 2005):

SN = O                      O SN pode ser uma oração;

SN = N                      O SN pode ser um nome;

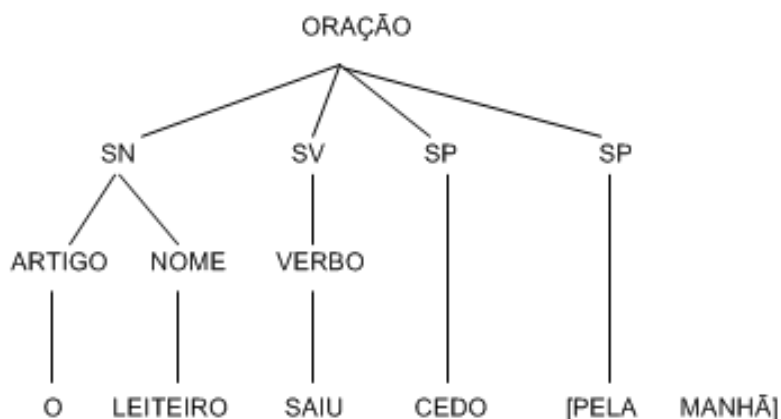
SN = Det+N                      É um determinante mais um nome;

SN = SN+O                      É um sintagma nominal mais uma oração.

Apesar de regras simples como as mostradas acima, Kuramoto (1996) conseguiu identificar 323 estruturas diferentes para SN ao analisar manualmente um corpus contendo 15 documentos. Isso porque Kuramoto explorou as diversas possibilidades de combinação entre os elementos da oração e a possibilidade de aninhamentos entre sintagmas, ou seja, os sintagmas podem possuir níveis hierárquicos, estando uns dentro de outros.

Vejamos abaixo a decomposição de um sintagma utilizando um diagrama arbóreo.

Figura 10 – Um exemplo de sintagma em forma arbórea



Fonte: Othero, 2009.

Essa maneira de “enxergar” e analisar a sentença e seus constituintes é conhecida como *Phrase Structure Grammar*, ou PSG.

Uma outra teoria seria denominada X-barras:

A teoria X-barras é um dos pilares do modelo de princípios e parâmetros da linguística gerativa. Trata-se de uma concepção restritiva da gramática de estrutura sintagmática, que permite análises psicolinguisticamente mais realistas e computacionalmente mais elegantes do que as praticadas anteriormente no âmbito da gramática gerativa. Desse modo, essa teoria tem sido empregada também na linguística computacional, em análises baseadas em formalismos gramaticais de natureza não transformacional que operam com a unificação de traços (ALENCAR apud OTHERO, 2009, pg. 82).

### 3.1 A sintaxe do sintagma nominal

Esta seção pretende trazer uma melhor compreensão sobre o sintagma nominal e o que ele referencia, do ponto de vista de Liberato. Não se trata de esgotar o assunto, mas essa base permitirá calcular o valor da semelhança estrutural do SN, conforme proposto no capítulo sobre metodologia.

O objetivo do trabalho proposto por Liberato (1997) é desenvolver uma análise do sintagma nominal com motivação semântica, mas considerando também

aspectos pragmáticos e funcionais. O trabalho do autor supracitado permite certo grau de generalização no reconhecimento dos sintagmas sem, contudo, tentar ser universal.

Talvez seja importante destacar aqui que os atos de comunicação entre falantes e ouvintes são influenciados diretamente por inúmeros fatores, dentre eles: sociais, psicológicos e de conhecimento da língua. Na verdade, qualquer mensagem entre os envolvidos é determinada por uma série de combinações de fatores complexos, nos quais a gramática responderia apenas por uma parte do problema. Podemos perceber isto na sentença número 1, abaixo destacada:

*1. A porta estava fechada!*

Ela pode simplesmente significar uma observação feita por alguém ou, dependendo de como a frase foi dita pelo falante, pode significar um imperativo para que a última pessoa que passou pela porta deixe-a novamente como estava antes.

Em outros casos, frases gramaticalmente diferentes podem ter a mesma referência:

*2. O filho de Carlos está com febre.*

*3. Meu sobrinho está doente.*

Em determinado uso real na comunicação entre dois falantes, tanto a oração 2 quanto a 3 podem se referir à mesma pessoa. Através destes simples exemplos já é possível perceber que a identificação do referente no sintagma passa a ser de fundamental importância para se definir a situação da efetiva comunicação.

Portanto, o que temos na formação básica do sintagma é a entidade ou referente, em conjunto com elementos que dão as pistas necessárias para a identificação do referente pretendido (LIBERATO, 1997). Vejamos alguns exemplos:

<b>SN</b>	<b>Referente</b>	<b>Recortador</b>
Ciência da informação	Ciência	da informação
Campo científico	Campo	Científico
Cientistas da informação	Cientistas	da informação
Informação científica	Informação	Científica
Evento de comunicação	Evento	de comunicação
Grupo de cientistas	Grupo	de Cientistas

Nos exemplos acima, os recortadores têm papel de restringir o referente, dando pistas mais adequadas para a identificação do referente sobre o qual estamos pretendendo dizer algo na mensagem. Ou seja, a presença e a posição dos elementos dentro de um sintagma nominal têm justificativas funcionais.

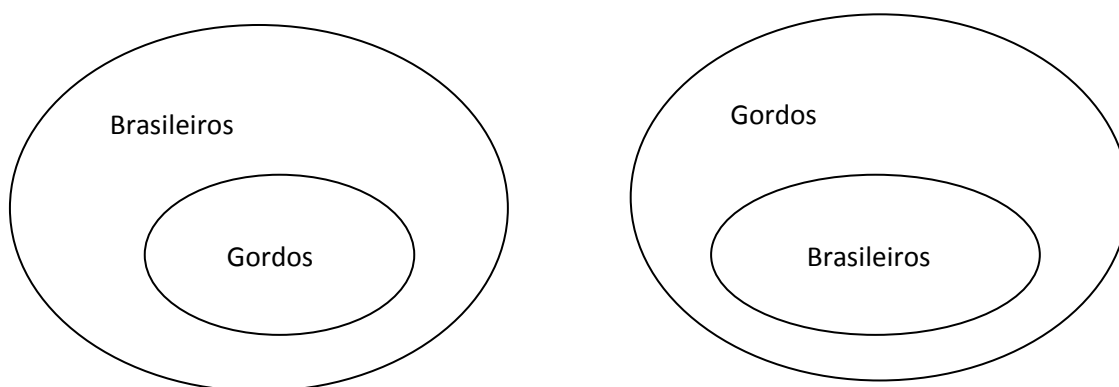
A identificação do referente em um SN passa necessariamente pela procura de uma palavra com sentido de substantivo. Ao ser identificada, ela será o classificador presente na sentença, pois cada sintagma nominal tem apenas um classificador. Ao encontrar dois ou mais classificadores dentro do SN, o ouvinte pode buscar fora desse SN algo que possa fazê-lo decidir por um deles. Um predicado, por exemplo, pode dar uma boa pista (LIBERATO, 1997). O exemplo na oração 4 ilustra bem uma possível dúvida na escolha do classificador:

4. *O cachorro do meu vizinho estacionou na minha garagem*

Nesse caso, tanto o cachorro quanto o vizinho poderiam, em um primeiro momento, ser candidatos a classificadores. Porém, como estacionar não é algo que pode ser realizado por um cachorro, fica fácil dizer que o vizinho é o referente da sentença. Ainda em 4, *cachorro* poderia ser identificado como um qualificador para vizinho.

5. *Os homens brasileiros gordos*

Figura 11 – Exemplo de um problema entre o referente e o classificador



Fonte: Liberato, 1997

A identificação correta do classificador na oração 5 dependeria do contexto no qual o sintagma está inserido. Portanto, é uma informação que está fora do sintagma. Vejamos o problema pelas sentenças 6 e 7:

6. *Os homens gordos brasileiros deveriam comer menos gordura.*



7. *Os homens brasileiros gordos são piores ainda, pois se alimentam inadequadamente.*

Em 6 temos um universo de homens gordos e estamos nos referindo apenas a uma parcela deles, que são os brasileiros. Já em 7 o universo de homens são todos os brasileiros e estamos nos referindo a uma parte deles, que são os gordos. Em ambos os casos, tanto *em homens gordos brasileiros* presentes em 6 quanto em *homens brasileiros gordos*, não é possível identificar diretamente o classificador.

### 3.2A identificação do referente

Parte da tarefa que realizamos, ao construir um significado a partir de um enunciado, está na identificação dos referentes representados pelo SN, sendo uma tarefa importante para a efetiva comunicação. Entretanto, os constituintes da língua não carregam o significado, pois ele deve ser construído pelo ouvinte através das pistas deixadas pelo sintagma e pelo seu conhecimento prévio (LIBERATO, 1997).

Segundo Daniel Andler (1992 apud PERINI, 2003), a busca por uma explicação unificada para o problema de identificação do referente é legítima, mas longínqua. Isso pode ser percebido de forma mais direta em Muller (1993, p. 273 apud PERINI, 2003, p. 19): “Este mundo, que contém os objetos os quais falamos, pode ser o ‘mundo real’ ou qualquer outro mundo sobre o qual somos capazes de falar, qualquer ‘mundo possível’”.

Austin (1962) demonstrou que a forma e a função entre o ato e o enunciado não são rígidas. Por exemplo, é possível dar ordens através de perguntas ou fazer perguntas usando formas declarativas. Além disso, podemos fazer afirmações que não envolvem uma entidade, por exemplo:

8. *Está quente.*

9. *O tempo está quente.*

Tanto em 8 quanto em 9 o tempo pode ser o referente, mas ele não está explícito em 8 e, nesse caso, irá depender de um contexto.

Em uma situação mais trivial pode-se ter:

10. *Edgard está incomunicável.*

Nesse caso, é particularmente fácil perceber *Edgard* como sendo o referente.

Entretanto, alguns autores como Chafe (1976) assumem que o referente não é uma entidade particular, mas a ideia que temos em nossa mente sobre tal entidade. Esse destaque feito por Chafe é importante para perceber que a forma como o autor produz um determinado texto leva em consideração o conhecimento presumido do ouvinte e, portanto, impacta diretamente na estrutura do sintagma, sendo produzido no momento da escrita.

Em particular, e para fins de realização de processamento via computador, o referente deverá ser uma entidade, e não um conceito ou ideia que se pode ter sobre um referente. Isso pode ser feito de algumas formas: **Forma 1 - Busca direta pelo sujeito:** o referente pode ser um nome próprio, como em 10 – *Edgard*; **Forma 2 - Busca por um SN descritivo:** usando a expressão linguística que descreve a entidade, como no exemplo “*o marido da minha vizinha fala como ninguém*”. Nesse caso “*o marido da minha vizinha*” seria a entidade. Entretanto, a entidade pode ainda não permanecer muito clara: no exemplo “*um filho da minha vizinha*”, no qual existem vários “*filhos da minha vizinha*” e o autor está se referindo a um deles, mas não se pode dizer qual.

Lyons (1977 apud LIBERATO, 1997) destaca que as atribuições feitas pelo falante podem ser verdadeiras ou não, porém precisam ser adequadas, como no exemplo 11.

*11. O atual rei do Brasil*

Já se sabe que atualmente não existe um rei no Brasil, mas isso não inviabiliza a identificação da entidade “*O atual rei*” através do SN descritivo.

Fauconnier (1994 apud PERINI, 2003) destaca a existência de referentes que podem ser entidades ou apenas ter papéis.

*12. O presidente foi eleito.*

*13. O apartamento de Carlos está fechado.*

Em 12 temos o papel de presidente, uma vez que ele será trocado a cada quatro anos. Já em 13 teremos uma entidade, pois o apartamento será sempre o de Carlos.

O referente também pode ter um uso predicativo:

*14. O namorado de Nina é o professor de matemática.*

Em 14 temos o “é o” como uma forma de estabelecer a identidade entre dois referentes, pois o falante estaria colocando para o ouvinte que o namorado de Nina

e o professor de matemática são, na verdade, o mesmo referente. Já em 15, o professor de matemática estaria sendo utilizado com uma função predicativa.

*15. O namorado de Nina é professor de matemática.*

Podemos ter também entidades que são abstratas.

*16. As feias que me perdoem, mas beleza é fundamental.*

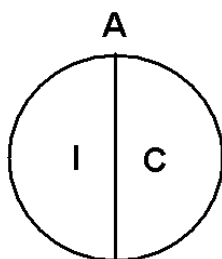
Em 16, beleza passa a ser uma entidade abstrata que irá representar as mulheres belas. Mas isso só poderá ser percebido em um contexto geral, pois, retirada do contexto, a entidade abstrata perderá o sentido.

Em outro exemplo podemos ter:

*17. Vou contratar um advogado competente.*

O falante poderia estar se referindo a um papel, no qual ele conheceria vários advogados e um deles seria o competente. Ele também poderia se referir a uma entidade, pois, neste caso, não conheceria nenhum advogado e estaria na busca por um que fosse competente, portanto, ele existiria em algum lugar, mas não seria conhecido.  $A = \{i+c\}$ : Total de advogados é a soma dos competentes mais os incompetentes.

Figura 12 – Conjunto representando  $A=\{i+c\}$



Fonte: Liberato, 1997

Em outra análise teremos o advogado competente como um papel desempenhado por um entre todos os advogados. Então, podemos dizer que, no primeiro exemplo, o adjetivo tem uma função restritiva e, no segundo exemplo, é uma característica.

Vale destacar neste momento que todas as análises que demandam um conhecimento prévio do ouvinte não foram consideradas, pois o objetivo deste trabalho é uma análise computacional, portanto, impossibilitada quando há a necessidade de interpretação extralinguística.

### 3.3 A identificação do classificador

Como vimos nas sentenças, os falantes podem se referir a uma entidade através de um nome ou descrevendo-a da maneira que lhe convém. Atribuir uma característica a um referente é, de certa forma, destacá-lo dentro de uma classe, como na sentença 18:

*18. Um exercício aeróbico pode lhe fazer bem.*

Enquanto “*exercício*” é claramente o referente, “*aeróbico*” está classificando o referente. Ou seja, não é qualquer exercício, tem que ser aeróbico. O nível de detalhamento, sua forma e as características escolhidas pelo falante poderão variar muito e ainda depender de cada situação ou caso em que é utilizado. Os recursos linguísticos do falante também podem influenciar nesse classificador, como também o ouvinte que ele tem em mente.

*19. O engenheiro*

*20. O engenheiro da Vale do Rio Doce*

A sentença 19 poderia ser utilizada quando o falante estivesse dentro da empresa Vale do Rio Doce. Já a sentença 20 poderia ser utilizada para o mesmo engenheiro, mas em uma situação em que o falante estivesse fora da empresa Vale do Rio Doce. Pode-se perceber um processo *top-down* de detalhamento do referente como forma de um detalhamento nas sentenças abaixo:

Engenheiro	TOP
Engenheiro de minas	
Engenheiro de minas recém-formado	
Engenheiro de minas recém-formado na UFOP	DOWN

Toda descrição mais ou menos detalhada começa com uma classe mais geral e vai sendo restringida, à medida que for necessária para a compreensão pelo ouvinte. Portanto, um classificador delimita a classe mais ampla em que o referente está enquadrado, em uma determinada descrição (LIBERATO, 2003).

### 3.4 Subclassificador e qualificador

Itens de sentido adjetivo parecem não funcionar como classificadores, mas apenas como subclassificadores. Os adjetivos podem ser de dois tipos: restritivo e explicativo. Os dois tipos podem ser vistos em 21 e 22:

21. *Os alunos que estudam se preparam melhor para o vestibular.*

22. *Flamengo, que é um time de futebol, está sediado no Rio de Janeiro.*

Em 21, “*que estudam*” pretende restringir os alunos, separando-os em uma subclasse de alunos (aqueles que estudam). Já em 22, “*que é um time de futebol*” poderia ser retirado da sentença sem maiores perdas de compreensão, pois apenas explica o substantivo Flamengo. Segundo Bechara (1996 apud PERINI, 2003, p. 75): “Chamam-se restritivas as que servem para delimitar ou definir melhor o seu antecedente, o qual, sem o concurso da oração adjetiva, pode ou não fazer sentido ou dizer coisa diferente do que se tem em mente”.

Bechara destaca também um detalhe que pode ser importante: aparentemente, o papel restritivo particulariza o substantivo, enquanto o explicativo o apresenta em um sentido mais universal. No exemplo 22, “*que é um time de futebol*” colocaria o substantivo Flamengo dentro de uma classe maior. Já “*que estudam*”, na sentença 21, estaria destacando alguns dentro de um universo maior, particularizando, portanto, apenas alguns alunos do universo.

Vejamos a explicação acima através de um diagrama:

$A = B + C$  (*Todos os presentes recebidos*)

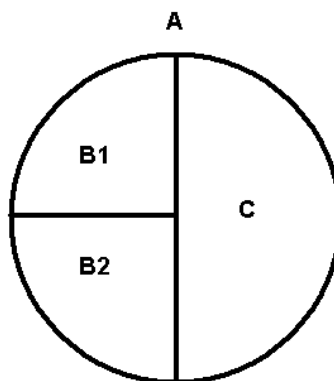
$B = B1+B2$  (*Todos os presentes que Raquel trouxe*)

$X = A - B$  (*Todos os presentes menos os que Raquel trouxe*)

$B1 =$  Os presentes de Raquel que estão fechados

$B2 =$  Os presentes de Raquel que estão abertos

Figura 13 – Conjunto representando  $A=\{B1+B2\}+C$



Fonte: Liberato, 1997

Nos exemplos acima é possível perceber claramente que as orações restritivas delimitam uma classe dentro de outra mais ampla, enquanto as explicativas não têm essa função de serem diferenciadoras.

Entretanto, os adjetivos também podem, dependendo da oração, serem colocados como explicativos:

23. *A lógica perversa do terrorismo*

24. *Terá espaço em sua agenda lotada.*

Percebe-se que “*perversa*” não é um tipo (classe) de lógica, mas sim a única lógica para o terrorismo, portanto explicativa (e não restritiva). O mesmo pode-se dizer de “*agenda lotada*”. Não existem duas agendas, uma lotada e outra vazia, é a mesma agenda, mas com a característica de ser lotada.

Portanto, Perini (2000 pg. 85) define: **Subclassificador** – delimita uma classe mais ampla em que o referente é enquadrado (SUB); **Qualificador** – fornece características do referente, mas não faz uma delimitação de classe ou subclasse a que pertence (QUA).

Segundo Perini (2000), na posição posposta ao classificador, podem ocorrer tanto subclassificadores quanto qualificadores. Na posição anteposta ao classificador ocorrem apenas qualificadores.

Retirado do contexto, um classificador na posição posposta pode ser considerado um classificador, entretanto, ao se considerar o contexto como um todo, ele poderá ser também um qualificador.

25. *A agenda lotada*

Fora do contexto, em 25, “a agenda” poderia ser um subclassificador, ou seja, existem vários tipos de agenda, inclusive uma delas que é a lotada.

*26. A agenda lotada do governador*

Em 26 o governador não teria várias agendas e uma delas seria a lotada. Nesse contexto existe apenas uma agenda. Deve-se perceber que, para essa análise, será exigido um conhecimento prévio do ouvinte sobre a não possibilidade de o governador ter várias agendas no seu dia a dia. Portanto:

*27. A análise tradicional não aponta esse erro. (SUB)*

*28. A lógica perversa. (QUA)*

*29. A agenda lotada (QUA)*

*30. A tradicional análise não apresenta erro. (SUB)*

*31. A perversa lógica (QUA)*

Percebe-se em 27 e em 30 que tanto o subclassificador quanto o classificador podem ocorrer depois do referente.

*32.. Esse resultado tem uma função prática sobre o destino.*

No caso de 32, “prática” está delimitando um tipo de função (poderia ser função lúdica, função criminal, etc.), portanto “prática” tem uma função restritiva.

### **3.5 O recortador e o quantificador**

Segundo Liberato (1997), há inúmeros estudos para determinar o papel de um artigo quando utilizado em um sintagma nominal. Russel (1971) traça um compromisso do artigo com o mundo real, o que pode ser percebido no exemplo abaixo:

*33. O atual presidente da França*

Em 33 pode-se perceber que o artigo “O” indica que apenas um presidente pode existir em cada momento (O atual). Já 34 confirma a existência de múltiplos habitantes.

*34. Um habitante de Londres*

Liberato pondera: devemos considerar apenas o mundo real? Chafe, Du Bois, Halliday e Hasan acreditam na noção de identificabilidade (LIBERATO, 1997). Nesse

caso, o ponto de vista deixa o autor e passa para o ouvinte e para sua capacidade de identificar referentes.

35. *Vou alugar o apartamento da Sara.*

36. *Vou alugar um apartamento da Sara.*

Em 35, o ouvinte pode claramente identificar o imóvel em questão a ser alugado. Nesse caso, Sara poderia ter vários apartamentos e apenas um deles estaria disponível para ser alugado, porém ela poderia também ter apenas um apartamento para ser alugado. Já em 36 está claro que Sara possui mais de um apartamento.

Pode-se dizer então que a identificabilidade parece suficiente para explicar a oposição entre os arquivos, ou seja, “o” é usado quando o referente é identificável pelo ouvinte, enquanto “um” é usado quando o referente não é identificável pelo ouvinte. Entretanto, neste caso a seguir, não existe uma entidade em particular para receber o documento:

37. *Por favor, entregue este documento à **secretária de um diretor**.*

Pode-se dizer que o “um” funciona como um subclassificador, tratando-se de uma delimitação quantitativa e não qualitativa. Dizemos, assim, que o chamado artigo indefinido exerce uma função quantificadora na delimitação do referente. Já o artigo definido não se confunde com o subclassificador, pois não restringe uma subclasse dentro de outra classe maior. Sua função é exatamente sinalizar que a quantidade total dos elementos da classe delimitada constitui o referente do SN (LIBERATO, 1997). Assim, concordamos com Liberato ao chamar os artigos, ou melhor, a função que eles exercem de *recortadores*.

Aparentemente o “um” funciona sempre como recortador parcial e “o” funciona como recortador universal. “Dois”, “três”, “quatro”, “vários”, “certos”, “alguns” também podem ter a função de recortador parcial dentro do SN.

38. *As poucas pesquisas indicam um campeão no primeiro turno.*

Nesse caso, “as poucas” já é a totalidade e não exerce a ideia de recortador como em 39, que determina a existência de muitas pessoas e, dentre elas, mesmo com poucas, já podemos mudar o mundo.

39. *Mesmo com poucas pessoas podemos mudar o mundo.*

Apesar de apresentado para a língua inglesa, Barkema (1994) coloca três critérios que, nesta tese, foram utilizados para determinar o nível de extratibilidade



da informação semântica do sintagma nominal: 1) Composicionalidade: indica quanta informação semântica pode ser extraída de um sintagma, principalmente quando este é composto por vários níveis; 2) Colocabilidade: indica o quanto um item pode ser substituído por outro que seja sinônimo ou antônimo; 3) Flexibilidade: indica o quanto o sintagma aceita variações morfológicas em seus termos. Complementando essa lista com as ideias de Nunberg, Sag e Wasow (OLIVEIRA, 1996): 4) Figuração: o grau de influência de metáforas e hipérboles que podem ser utilizadas; 5) Proverbialidade, Informatividade, Afetação: a influência de expressões idiomáticas sobre o sintagma.

Oliveira (1996) destaca que o critério de flexibilidade é mais diretamente relacionado à morfologia, apesar de poder haver uma relação entre propriedades semânticas e versatilidade sintática, e entre o critério de composicionalidade e flexibilidade. Barkema propõe um estudo estatístico baseado em corpus, para estabelecer a frequência de uma construção e de suas variantes, gerando o perfil de flexibilidade de uma expressão (OLIVEIRA, 1996).

### **3.6 Influências na produção do texto escrito**

Os processos de comunicação costumam ser tratados muitas vezes de maneira linear: emissor – mensagem – meio – receptor. Há uma concentração na troca de mensagens, mas não há um olhar para os emaranhados de relações que existem em todo o complexo processo comunicacional, especialmente midiático. Hall diz que é “possível (e útil) pensar esse processo em termos de uma estrutura produzida e sustentada através da articulação de momentos distintos, mas interligados – produção, circulação, distribuição/consumo, reprodução” (HALL, 2003, p. 387).

Cada pessoa tem sua maneira própria de falar, cada região possui marcas linguísticas ou maneiras diferentes de dizer a mesma coisa, fato que as diferencia. Cada grupo adquire vocabulários que o caracterizam, principalmente em um país marcado pela diversidade. Tudo isso influencia no momento da escrita, porque, querendo ou não, deixamos nossas marcas – é inevitável querermos “esconder” a forma como cada um escreve (LÚZIO, 2011).

Nesta seção procuramos levantar algumas questões sobre a formação estrutural de um sintagma, ou seja, o que pode influenciar o autor na escolha de

determinadas palavras em detrimento de outras. A ideia não era esgotar os incontáveis motivos que podem levar à escolha de determinadas estruturas em detrimento de outras, mas apenas mostrar os possíveis impactos na produção da escrita.

Parte das respostas que buscamos pode ser encontrada no estudo estilométrico da linguagem. Estudos desse tipo não são recentes: em 1851, ferramentas estatísticas foram utilizadas para testar questões de autoria. O matemático Augustus de Morgan propôs usar o comprimento médio da palavra para caracterizar numericamente um estilo de autoria (PATERIYA, 2003).

Em seguida, o físico Thomas Mendenhall propôs que um autor tem uma "curva característica da composição" determinada pela forma como o autor usa palavras de diferentes comprimentos com frequência. Em 1888, o matemático William Benjamin Smith publicou dois artigos descrevendo uma "curva de estilo" para distinguir estilos autorais com base em sentença média (PATERIYA, 2003).

Devemos nos ater a alguns detalhes do ato da escrita humana que podem ser extremamente importantes ao utilizarmos a estrutura dos sintagmas na recuperação da informação. Por exemplo, ao observar a estilística na linguística pode-se concluir que: 1) dois escritores nunca vão escrever da mesma forma; 2) um mesmo autor nunca vai escrever da mesma forma o tempo todo. Os impactos, ao especular que essas duas características nunca irão mudar em um mesmo documento ou em documentos diferentes, podem ser desastrosos para a robustez dos sistemas de recuperação da informação.

Nas próximas seções, descreveremos os fatores que do nosso ponto de vista podem influenciar na automatização da RI, com uma visão obviamente reducionista sobre os aspectos que podem ser automatizados, mas sem perder de vista o assunto demasiadamente complexo e que toca não apenas em áreas como a linguística, mas também na cognição, na psicologia, na filosofia e em diversas outras.

### **3.6.1 A influência do estilo literário do autor**

Utilizando ferramentas quantitativas é possível identificar o quanto e com que frequência uma determinada forma sintática foi utilizada em um grupo de

documentos escritos por um mesmo autor ou por autores diferentes (JOHNSTONE, 2000). Em geral essa análise está baseada nos atributos estilométricos, tais como a taxa de aparecimento de palavras incomuns ao vocabulário médio de uma população, o tamanho médio de cada oração criada pelo autor, o quociente de palavras diferentes em relação ao total de palavras, entre outros. Esse conjunto de informações quantitativas poderá, por exemplo, identificar um determinado autor de documento, sem que o mesmo tenha sido citado diretamente no texto (PAVELEC, 2006).

Os métodos automáticos de verificação da autoria de textos baseiam-se usualmente em duas abordagens: global e pessoal. A abordagem pessoal utiliza um modelo por autor, enquanto a abordagem global faz uso de um modelo geral para todos os autores. O modelo pessoal usualmente exige um conjunto elevado de textos de um dado autor, para treinamento e geração de um modelo robusto, mas apresenta a vantagem de modelar adequadamente os atributos estilométricos do autor. O modelo global possui a desvantagem da generalização, mas possui a vantagem de necessitar de um número reduzido de textos para cada autor e de não necessitar de um novo treinamento do modelo diante da inclusão de novos autores (PAVELEC, 2006).

### **3.6.2 A influência do sexo do autor**

Outro fator que pode influenciar na estrutura do SN é o sexo do autor. Existe atualmente uma série de pesquisas que tratam do processo de identificação da semelhança de escrita entre pessoas do mesmo sexo. A identificação do sexo de um autor pode ser realizada através de técnicas que permitem uma identificação correta em até 80% dos casos (KOPPEL, 2014).

Segundo os resultados apresentados por Koppel, o emprego de aprendizagem de máquina por técnicas de inteligência artificial pode contribuir substancialmente na identificação do sexo do autor de determinado documento, a partir do treinamento desse sistema, utilizando um corpus para o aprendizado. De acordo com Koppel (2003), sua técnica pode ser utilizada também em outros tipos de categorização, tais como: distinguir textos de ficção e de não ficção, classificar o grupo demográfico do autor e classificar de forma cronológica.

Bortoni-Ricardo (2004) salienta essa relação do gênero. Ela diz que as mulheres usam mais diminutivos e marcadores conversacionais, já os homens usam uma linguagem menos formal, mais pejorativa, com gírias. A autora salienta ainda que essas diferenças entre os vocabulários feminino e masculino fazem referência aos papéis sociais, culturalmente condicionados.

### **3.6.3 A influência da oralidade na escrita**

A fala é diferente da escrita sob muitos aspectos. Cada uma dessas modalidades tem características próprias, mas uma influencia a outra, especialmente a fala na escrita. Segundo os gramáticos, a oralidade é mais fácil, mais usada em nosso cotidiano, nela são permitidos alguns “erros”; enquanto a escrita é mais complexa, rígida, rebuscada. A oralidade, talvez por ser mais usada, deixa, muitas vezes, suas marcas em textos escritos (LÚZIO, 2011).

Pode-se deduzir que a escrita tem uma estrutura canônica muito mais convencionada do que a fala, entretanto, o autor pode utilizar recursos da língua falada para, por exemplo, chamar a atenção do leitor. Quando alguém se pronuncia, propicia a presença real, mais íntima do interlocutor, além de ser mais fácil de convencê-lo de alguma coisa (dependendo da intenção do emissor), cobrando-se de maneira rápida a receptividade. Na modalidade escrita isso não acontece tão facilmente e, quando acontece, dependerá também da elaboração e da construção de cada sintagma no texto.

### **3.6.4 A influência do suporte digital**

É inegável e claramente perceptível a influência do suporte digital no momento da produção da escrita, a ponto de Prensky (2001) acreditar que é possível identificar esses nativos digitais, inclusive pela sua linguagem.

Os nativos digitais estão acostumados a receber informações muito rapidamente. Eles gostam do processamento paralelo e da multitarefa. Eles preferem gráficos antes do texto e trabalham melhor quando conectados em rede. Eles progridem com gratificações instantâneas e recompensas frequentes. Eles preferem jogos ao invés de trabalho sério (PRENSKY, 2001, p. 77).

Há nessa linguagem uma tendência em tomar o ato de fala com base na produção escrita, pois os usuários se adaptam às operações instantâneas e automáticas de conversação. Ou seja, eles se apropriam de uma linguagem oral, que de modo geral tem muita expressividade, e transferem-na para a escrita. Desse modo, o texto produzido por eles se aproxima bastante dessa representação oral. Observa-se certo grau de informalidade na comunicação escrita típica do meio digital, influenciando diretamente o tamanho das orações e as palavras utilizadas que, em geral, são resumidas e tendem a conter uma carga oral mais acentuada.

O uso de meios de texto limitados pode fazer com que, aos poucos, o próprio cérebro perca gradativamente sua capacidade de raciocínio, já que o discurso utilizado nas salas de bate-papo e em outros locais, onde se exige orações curtas, pode interferir na produção textual realizada.

Segundo Amaral, “a linguagem adotada no mundo digital requer habilidades de escrita rápida para esta geração net, o que cria uma solução intermediária de comunicação” (AMARAL, 2003).

### **3.6.5 A influência do uso de certas palavras no decorrer do tempo**

Uma verdade indiscutível na ciência linguística é a de que todas as línguas mudam com o passar do tempo: elas vão evoluindo, adaptando-se aos usos inovadores da comunidade falante. Apesar do que supõem a tradição de ensino gramatical e a escola, a língua não pode ser entendida como uma entidade imutável e estanque. Ela é, ao contrário, dinâmica e passível de mudanças (OTHERO, 2004), constatação também feita por Possenti (apud OTHERO, 2004, p. 98): “não há língua que permaneça uniforme”, todas as línguas mudam.

Com o passar do tempo, novas formas sintáticas serão criadas, ou antigas serão modificadas; haverá criações de neologismos e formação de palavras inéditas, empréstimos linguísticos etc. Dessa forma, o idioma estará se adaptando o melhor possível à sua comunidade linguística e, principalmente, estará atendendo às exigências de seus falantes (OTHERO, 2004a). Estamos presenciando mudanças linguísticas com a língua portuguesa em dois diferentes meios linguísticos, como em Portugal e no Brasil. Em cada lugar, estão acontecendo evoluções linguísticas diferentes, que estão fazendo com que a língua portuguesa do Brasil e a língua portuguesa de Portugal se diferenciem.

Alguns teóricos se apoiam na *teoria do uso e desuso*, em que apenas aquilo que é efetivamente usado e útil, que realmente sirva para o falante, é aproveitado. O que não é usado e não é exercitado (como a palavra vós) dificilmente passará às gerações. Dessa forma, certas palavras e formas da língua podem entrar em desuso, tornando-se arcaicas e obsoletas, e não chegar ao conhecimento de futuros falantes.

Pela mesma regra, o contrário também ocorre com frequência: se bastante utilizadas e “reforçadas” através do uso, a tendência é que novas formas se incorporem à língua, deixando-a mais rica e mais adaptada para melhor atender a seus usuários.

### 3.6.6 A influência do gênero textual

A produção e o uso de certas palavras podem estar diretamente ligados ao gênero no qual o texto produzido irá se inserir. Biber (1996) já preconizava ser possível uma análise quantitativa de um texto com o objetivo de classificá-lo como informativo, narrativo etc.

O estilo é frequentemente diferenciado por uma questão de estética textual, que está intimamente ligada às convenções de cada gênero (BIBER, 1986). Exemplos que comprovam isso podem ser encontrados na classificação de gênero proposta por Kessler et al. (1997), na identificação do perfil de autor proposta por Garena (2011), na análise de sentimentos de Wilson (2005) e na classificação de legibilidade proposta por Collins e Callan (2005).

Exemplos da automação desse processo podem ser vistos no trabalho de Brett, Kessler, Geoffrey, Nunberg, Hinrich e Schfitze, que propuseram um algoritmo automático para determinar o gênero de um determinado texto. Esse recurso seria importante no momento de identificar se determinado texto é informativo, poético, jornalístico, literário ou se pertence a outro gênero qualquer. Os autores apresentam três sugestões genéricas para identificar e classificar um texto segundo o gênero: **Análise estrutural** – observando a forma como o texto se apresenta, os tipos de sentenças, a média do tamanho das sentenças, entre outros aspectos; **Análise lexical** – o uso de determinadas estruturas como “Sr.”, “Sra.”, pode sugerir textos jornalísticos, assim como textos com datas são mais utilizados em documentos históricos; **Análise de caracteres** – o uso excessivo de exclamações ou de

interrogações pode sugerir determinados tipos de texto, assim como uso de siglas, e outros recursos da língua.

Scott A. Crossley e Max Louwerse apresentaram um trabalho no qual bigramas podem ser utilizados para classificar documentos segundo quatro linhas de estilo: 1) roteirizado ou não roteirizado; 2) planejado ou não planejado; 3) localizado ou não localizado; 4) direcionado ou não direcionado (CROSSLEY; LOUWERSE, 2014). Esses e outros aspectos poderiam influenciar sobremaneira o documento a ser recuperado para um usuário, ao utilizar um sistema de recuperação da informação.

### **3.6.7 A influência de fatores sociais, emocionais e de formação do autor**

No que concerne ao status socioeconômico, de acordo com Bortoni-Ricardo (2004) tais diferenças representam desigualdades na distribuição de bens materiais e de bens culturais, o que se reflete em diferenças sociolinguísticas. Entre os bens culturais, ressalta-se a inclusão digital, o acesso ao computador e à internet, claramente associados ao status socioeconômico.

Com referência ao grau de escolarização, está evidente que há diferenças. Uma pessoa que teve a oportunidade de estudar terá uma forma de escrever bem diferente de uma pessoa que não teve acesso a uma universidade, por exemplo.

Com relação à influência do meio sobre o “jeito” de escrever de cada ser humano, Calvet (2002) menciona que as expressões distinguem-se geográfica, social e historicamente. A fala espontânea varia da mesma forma: não há as mesmas atitudes linguísticas na burguesia e na classe operária, na conversação de adultos e na de adolescentes, em um grupo de escolarizados e em um grupo de pessoas com pouca instrução (LÚZIO, 2014). Esses e outros fatores podem influenciar significativamente na forma como o texto é produzido, impactando diretamente a estrutura do sintagma e a qualidade final dos sistemas que utilizam essa estrutura.

## 4 EXPERIMENTO DE PROSPECÇÃO

Nas próximas seções serão apresentados os *corpora* utilizados durante a pesquisa, para garantir algumas das ponderações detalhadas nas conclusões. Também serão apresentadas as ferramentas tecnológicas utilizadas e, quando necessário, será aberta uma discussão comparando o processo manual ao objetivo de validar o processo automático. Entretanto, a validação da extração automática dos sintagmas utilizando processos manuais não faz parte do escopo deste trabalho e apenas será descrito e comparado de forma superficial.

### 4.1 Primeira etapa: Comparação direta entre os documentos

Foi perceptível, durante o início dos trabalhos, a necessidade de dividir o experimento em dois momentos. No primeiro momento foram feitas tentativas de tornar a estrutura de documentos sobre um mesmo assunto o mais semelhante possível. No segundo momento, agora com a hipótese já comprovada, a metodologia criada seria experimentada no treinamento de uma máquina de vetor de suporte, com a intenção de classificar documentos em três áreas do conhecimento.

Nessa primeira etapa de prospecção, pretende-se demonstrar a extração dos sintagmas de forma automática de um grupo de documentos, que tratam do tema “Inteligência Artificial”, e realizar uma primeira comparação entre esses documentos.

O tema Inteligência Artificial (IA) foi escolhido por ser um tema recorrente em vários grupos de pesquisa e também porque esse assunto tem uma relação muito próxima com a questão da classificação da informação na área de Ciência da Informação, servindo de base para construção de softwares ditos inteligentes no trato com a informação.

Nesta etapa serão extraídos os sintagmas nominais dos *corpora* escolhidos, seus elementos serão planilhados e alguns aspectos serão identificados, tais como: os quantificadores, o referente e os classificadores do referente.

Ainda nesta etapa de prospecção será demonstrado o grau de semelhança entre os sintagmas do corpus com o tema escolhido, tendo em vista que qualquer algoritmo de classificação baseia-se primordialmente em encontrar semelhanças entre as estruturas (características) que serão apresentadas ao sistema.



O objetivo desta etapa de prospecção é encontrar um mecanismo que possa aumentar a semelhança estrutural entre os sintagmas identificados em documentos de um mesmo tema, mas que, por outro lado, não aumente a semelhança estrutural de documentos de temática diferente. Esse ponto é crucial para o andamento da pesquisa.

O enfoque proposto terá como objetivo validar três etapas:

1. **Quantificação:** eliminar os quantificadores dos sintagmas (ex.: muitos, vários, um, uma etc.);
2. **Sinonimização:** encontrar sinônimos para os referentes e qualificadores do sintagma;
3. **Stemming:** transpor o sintagma para sua forma mais primitiva através de *stemming* das palavras que compõem os sintagmas escolhidos.

É importante ressaltar que a cada etapa da metodologia prospectiva será realizada uma nova comparação entre os sintagmas dos documentos do corpus. Através das planilhas apresentadas em cada etapa, poderemos concluir sobre a melhora ou não na quantidade de sintagmas semelhantes encontrados, o que deverá comprovar ou não a melhora no grau de semelhança entre os documentos.

Ao final dessa primeira etapa já poderemos, mesmo que de forma preliminar, quantificar a capacidade do sistema de melhorar sua performance final na realização de comparação entre dois ou mais documentos. Passaremos então para a etapa apresentada no próximo capítulo, que terá como objetivo comprovar que o uso do mecanismo proposto pode melhorar o grau de acertos na classificação de documentos, utilizando, para este teste, um algoritmo para IA.

Vale ressaltar que, apesar dos poucos documentos tratados nesta etapa preliminar, eles devem ser suficientes para demonstrar a melhora que pode ser obtida com os primeiros aspectos propostos. Entretanto, o processo será consolidado apenas com um grupo maior de documentos, tarefa que será vista na classificação com três temáticas, no próximo capítulo.

É preciso mencionar que várias ferramentas tecnológicas serão utilizadas e serão preferencialmente usadas quando suficientes para facilitar ou mesmo eliminar os processos manuais. Algumas etapas, no entanto, ainda carecem de processamento manual, como, por exemplo, limpar documentos PDF após convertidos em texto, etapa importante para eliminar grande parte do “lixo textual” que poderia influenciar nos algoritmos que serão utilizados. Como são utilizados

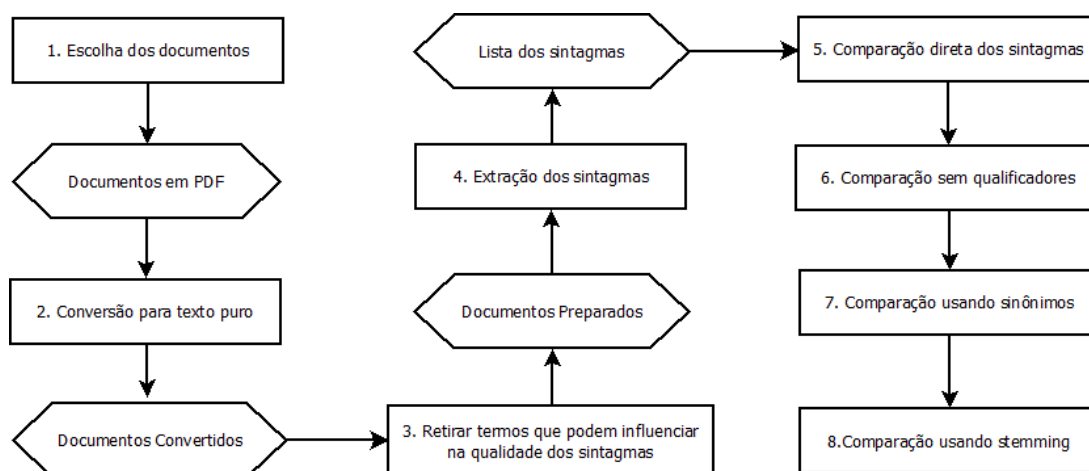
artigos científicos no corpus, palavras como “resumo”, “abstract”, “conclusões”, “referências” e “página” etc. formam o grupo de *stop-words* que serão removidas.

É importante lembrar que, para comprovar o funcionamento da etapa prospectiva, apenas o processamento manual do corpus poderia ser utilizado, uma vez que o principal objetivo dessa etapa é realizar a comparação entre os documentos antes e depois da metodologia proposta, a fim de demonstrar sua eficácia.

Entretanto, uma classificação eficaz de dezenas de milhares de documentos espalhados pelo meio digital só é possível com processos automáticos.

A Figura 14 apresenta o diagrama com o fluxo principal desta primeira etapa:

Figura 14 – Diagrama com etapas do tratamento de prospecção



Fonte: O próprio autor

#### 4.1.1 Escolha dos documentos

A escolha do primeiro grupo de documentos foi realizada seguindo os critérios especificados a seguir. Eles podem ser encontrados e acessados em revistas científicas da área das ciências exatas a partir do portal Scielo. Como dito anteriormente, foram escolhidos três artigos relacionados ao tema de IA. O único critério a respeito dos documentos escolhidos é que eles deveriam possuir mais de cinco páginas de texto em formato PDF, gerando assim uma quantidade razoável de sintagmas a serem extraídos. Textos de outras áreas e com menor tamanho foram utilizados durante a segunda etapa de nossa metodologia, como também para realização da contraprova do cálculo de similaridades.

Para exemplificar, seguem abaixo o título e o resumo de cada documento utilizado no primeiro corpus:

Documento 1 (referenciado a partir de agora como Doc 1): ***Inteligência Artificial Aplicada a Ambientes de Engenharia de Software: Uma Visão Geral***

*Resumo: A Inteligência Artificial sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana. Softwares são produtos intangíveis e utilizam no seu processo de construção recursos intelectuais humanos, que vão desde sua especificação até sua distribuição e pleno funcionamento. Como meio de auxiliar o processo de Engenharia de Software, foram criados os ambientes de Engenharia de Software centrados no processo, que possuem um conjunto de ferramentas integradas. Baseado neste contexto, este artigo vem mostrar alguns ambientes existentes que utilizam técnicas de Inteligência Artificial e propor o uso de outras técnicas para melhorar os Ambientes de Engenharia de Software, trazendo uma maior facilidade de construção de softwares e uma maior qualidade para os mesmos.*

Documento 2 (Referenciado a partir de agora como Doc 2): ***Inteligência Artificial e aprendizado***

*Resumo: Este tutorial tem por objetivo apresentar uma introdução ao aprendizado artificial e automatizado (machine learning), focalizando-se sobre os aspectos referentes a uma técnica em particular, as redes neurais artificiais – R.N.A. Na primeira seção vamos discutir sobre a Inteligência Artificial, sobre a aquisição de conhecimentos e sobre a importância do aprendizado na construção de sistemas inteligentes. Na segunda seção iremos abordar as redes neurais artificiais (modelos conexionistas), onde vamos destacar: os diferentes tipos de redes e de algoritmos de aprendizado existentes; a representação do conhecimento neural; as características e limitações de uso deste tipo de técnicas, bem como mostraremos alguns exemplos de aplicações das RNAs. Para concluir, iremos discutir sobre os caminhos da pesquisa atual nesta área e tendências futuras no que diz respeito ao desenvolvimento dos sistemas inteligentes.*

Documento 3 (Referenciado a partir de agora como Doc3): ***Aplicação da Inteligência Artificial em Sistemas de Gerenciamento de Conteúdo***

*Resumo: Existem vários arcabouços para apoiar o gerenciamento do conhecimento. Alguns apoiam o ciclo de vida do conhecimento, outros a sua produção. Atualmente a perspectiva está sobre o gerenciamento do conhecimento. Técnicas de inteligência artificial podem auxiliar nesse gerenciamento. Este trabalho mostra como estas técnicas podem ser aplicadas nos sistemas de gerenciamento de conteúdo para melhorar o gerenciamento do conhecimento destes sistemas.*

#### **4.1.2 Preparação do Corpus**

A utilização de documentos eletrônicos em pesquisas necessita de tratamento prévio para sua efetiva utilidade. Torna-se fundamental uma preparação utilizando ferramentas tecnológicas ou processos manuais para que eles fiquem em um

formato razoável para tratamento. Problemas com formatos (doc, xls, pdf), uso de imagens e de tabelas ou uso de versões diferentes de software no momento da criação podem limitar o uso desses documentos ou introduzir elementos indesejados à realização do processamento.

Uma solução que já vem sendo utilizada pela comunidade científica é a conversão desses formatos em um padrão mais simples, conhecido como texto puro (txt). Apesar da aparente perda de informação nesta passagem, trata-se de um formato que é utilizado como fonte de dados para a maioria dos sistemas de PLN. A seguir apresentaremos todo o processo.

#### 4.1.3 Converter os documentos para texto puro

Este processo frequentemente pode ser realizado através do uso de algum software ou biblioteca de software específica. Trata-se de uma etapa importante, pois o documento de origem pode estar codificado em diversos formatos binários, o que inclui diversos idiomas e padrões de metadados, além de conter imagens e tabelas, dificultando a conversão.

Abaixo temos uma pequena amostra do documento PDF e seu trecho em TXT:

Figura 15 – Trecho em PDF de um dos documentos utilizado

### **Inteligência Artificial Aplicada a Ambientes de Engenharia de Software: Uma Visão Geral**

RENATO AFONSO COTA SILVA<sup>1</sup>

Departamento de Informática – Universidade Federal de Viçosa  
CEP 36570-000 Viçosa, MG

<sup>1</sup>[renatoacs@dpi.ufv.br](mailto:renatoacs@dpi.ufv.br)

**Resumo.** A Inteligência Artificial sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana. Softwares são produtos intangíveis e utilizam no seu processo de construção recursos intelectuais humanos, que vão desde sua especificação até sua distribuição e pleno funcionamento. Como meio de auxiliar o processo de Engenharia de Software, foram criados os ambientes de Engenharia de Software centrados no processo, que possuem um conjunto de ferramentas integradas. Baseado neste contexto, este artigo vem mostrar alguns ambientes existentes que utilizam técnicas de Inteligência Artificial e propor o uso de outras técnicas para melhorar os Ambientes de Engenharia de Software, trazendo uma maior facilidade de construção de softwares e uma maior qualidade para os mesmos.

**Palavras-Chave:** Inteligência Artificial, Ambientes de Engenharia de Software, Processo de desenvolvimento de Software

Fonte: O próprio autor.

A saída deste trecho em texto puro:

*Resumo. A Inteligência Artificial sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana. Softwares são produtos intangíveis e utilizam no seu processo de construção recursos intelectuais humanos, que vão desde sua especificação até sua distribuição e pleno funcionamento. Como meio de auxiliar o processo de Engenharia de Software, foram criados os ambientes de Engenharia de Software centrados no processo, que possuem um conjunto de ferramentas integradas. Baseado neste contexto, este artigo vem mostrar alguns ambientes existentes que utilizam técnicas de Inteligência Artificial e propor o uso de outras técnicas para melhorar os Ambientes de Engenharia de Software, trazendo uma maior facilidade de construção de softwares e uma maior qualidade para os mesmos.*

Nessa etapa todos os documentos foram convertidos para formato de texto puro (txt), gerando a seguinte lista:

Tabela 1 – Lista dos documentos utilizados na primeira etapa

ID	Documento original	Documento texto
01	IA Aplicada a Ambientes de Engenharia de software.pdf	Doc1.txt
02	Inteligência artificial e aprendizagem.pdf	Doc2.txt
03	Aplicação da IA em sistemas de gestão de conteúdos.pdf	Doc3.txt

Fonte: O próprio autor

#### 4.1.4 Filtragem preliminar do conteúdo

Foi observado certo cuidado com palavras que poderiam influenciar na quantidade e na qualidade dos sintagmas encontrados nos textos. Para evitar esse problema, algumas estruturas recorrentes em artigos científicos foram removidas antes da extração dos sintagmas. São elas:

1. As palavras: resumo, abstract, introdução, referências (Removidas de forma automática);
2. Todas as referências citadas no final dos documentos (Removidas manualmente);
3. Títulos e subtítulos de capítulos ou seções dentro do documento (Removidas manualmente);

4. Aspectos recorrentes na escrita técnica, tais como RNAs, PDFs, nos quais o “s” colocado no final da sigla também foi removido (De forma automática).

As etapas 1 e 4 foram realizadas de forma automática por algoritmo construído para esta finalidade os quais estão presentes nos anexos. As etapas 2 e 3 foram realizadas manualmente.

#### 4.1.5 Extração dos sintagmas nominais

Chegamos, então, a uma etapa importante dos trabalhos: a extração dos sintagmas nominais utilizando PLN. O software escolhido é denominado *Palavras* e foi desenvolvido pela *Southern University of Denmark*, com licença de uso adquirida recentemente pelo Departamento de Letras da UFMG e sendo utilizado no Laboratório de Estudos Empíricos da Linguagem LEEL/FALE/UFMG. Esse software faz parte da ferramenta chamada VISL (Virtual Interactive Syntax Learning) e pode ser acessado e utilizado livremente, apesar de limitações no seu uso via Internet.

O VISL opera documentos em diversas línguas trabalhando de forma automática ou semiautomática, recebendo documentos em texto e retornando o conteúdo marcado em nível morfológico, sintático e semântico. Cada oração é marcada com todas as suas possibilidades sintáticas e semânticas, após diversas interações para resolução de ambiguidades, o que o torna um software extremamente robusto.

O exemplo a seguir apresenta um trecho do arquivo Doc1 após a marcação pelo software *Palavras*:

<i>potencialmente</i>	[potencialmente]	ADV @<ADVL #13->12
<i>relevante</i>	[relevante]	<jh> ADJ F S @<SC #14->12
<i>para</i>	[para]	PRP @<ADVL #15->12
<i>qualquer</i>	[qualquer]	<quant> DET F S @>N #16->17
<i>esfera</i>	[esfera]	<cc> <geom> <Labs> N F S @P< #17->15
<i>de</i>	[de]	<sam-> <np-close> PRP @N< #18->17
<i>a</i>	[o]	<-sam> <artd> DET F S @>N #19->20
<i>atividade</i>	[atividade]	<act-d> <activity> N F S @P< #20->18
<i>intelectual</i>	[intelectual]	<jh> <np-close> ADJ F S @N< #21->20
<i>humana</i>	[humano]	<jh> <np-long> ADJ F S @N< #22->20

Na primeira coluna temos o texto enviado, na segunda coluna a forma canônica e na terceira coluna a totalidade das marcações morfossintáticas. No trecho acima temos como exemplo o lexema “humana”, sua forma canônica “humano” e sua classificação como ADJ F S (adjetivo, feminino, singular). Nos anexos temos uma lista dos símbolos possíveis para marcação do *Palavras*.

Uma grande inovação do *Palavras* é a sua capacidade de extrair os sintagmas presentes nos documentos. Esse processo não precisa, necessariamente, ser feito no mesmo software que realiza a marcação morfossintática, mas mostra-se mais simples se utilizado no mesmo ambiente. O seguinte trecho mostra a extração do sintagma:

#### A ATIVIDADE INTELECTUAL

<i>a</i>	[o]	<-sam> <artd> DET F S @> <u>N</u> #19->20
<i>atividade</i>	[atividade]	<act-d> <activity> N F S @P< #20->18
<i>intelectual</i>	[intelectual]	<jh> <np-close> ADJ F S @N< #21->20

A seguinte marcação <np-close> identifica o término do sintagma. O sintagma também pode ser identificado pelas marcações @>N (início do sintagma) e @N< (final do sintagma). A partir desse documento é possível remover outras instâncias morfossintáticas, tais como verbos, pronomes e adjetivos, entre outros.

Com relação ao total de sintagmas extraídos, o *Palavras* realiza essa busca de forma exaustiva, ou seja, ele cobre a totalidade do texto gerando sintagmas que podem ou não ser relevantes para determinada finalidade. A título de exemplo, para o Doc1 foi gerada uma lista de 1903 sintagmas, incluindo nessa lista os repetidos.

Para extração dos sintagmas a partir do documento marcado, foi utilizado um *script* fornecido pelo próprio laboratório VLSI, escrito na linguagem PERL, que gera uma saída semelhante à mostrada a seguir:

```
A *Inteligência Artificial
*tarefas
qualquer *esfera de__a atividade intelectual humana .
_a *atividade intelectual humana .
*Softwares
*produtos intangíveis
_o seu *processo de construção
*construção
*recursos intelectuais humanos
```

A saída pelo *script* lista todos os sintagmas, um em cada linha, além de identificar o referente do sintagma utilizando um sinal de \* (asterisco).

#### 4.1.6 Normalização da lista gerada

A próxima etapa tratou de limpar a lista de sintagmas gerados pelo *Palavras*, para evitar que elementos como aspas, pontuação, asteriscos, sinais matemáticos, excesso de espaços entre palavras e outras estruturas pudessem interferir na qualidade do sintagma, quando utilizado na classificação dos documentos. Para esse processo foi utilizado o *Microsoft Excel*.

A tabela 2 apresenta o total de sintagmas extraídos e utilizados de cada documento.

Tabela 2 – Total de sintagmas extraídos dos documentos

ID		Total de sintagmas	Sintagmas únicos	Percentual de únicos
Doc1		1903	1324	69%
Doc2		3542	2525	71%
Doc3		1635	1169	71%

Fonte: O próprio autor

Os seguintes caracteres foram tratados no Excel:

- . Remoção do \* que estava sendo utilizado para destacar o referente;
- . Remoção do \_ que estava sendo utilizado para destacar o quantificador;
- . Remoção de aspas;
- . Remoção de parênteses e chaves;
- . Remoção de pontuação (? ! , . : ; );
- . Remoção dos textos: A seção 1, A seção 2, A seção n, Figura 1, Figura 2, Figura n;
- . Os 's (apostrofo s) para denotar plural em siglas, tais como: RNAs, RBCs;
- . Remoção de números dentro de sintagmas

Um trecho da lista final pode ser vista abaixo:

a área  
a área  
a área de conhecimento especializado  
a área jurídica que recupera casos para auxiliar a denúncia de homicídios  
a arquitetura  
A arquitetura de o ambiente  
A arquitetura de um PSEE



*a atividade*  
*a atividade intelectual humana*

...

Os scripts utilizados para essa tarefa encontram-se nos anexos.

#### 4.1.7 Sintagmas únicos e ordenados

Após a remoção dos caracteres indesejados, a etapa 4 foi a execução de um processo automático para identificação dos sintagmas únicos (resultado de uma comparação letra por letra de cada SN) já que vários deles se repetiam ao longo dos documentos, conforme mostrado na listagem anterior.

Nessa etapa também foram utilizados *scripts* do Excel e, ao final, temos como exemplo de saída a seguinte listagem:

*a área*  
*a área de conhecimento especializado*  
*a área jurídica que recupera casos para auxiliar a denúncia de homicídios*  
*a arquitetura*  
*A arquitetura de o ambiente*  
*A arquitetura de um PSEE*  
*a atividade*  
*a atividade intelectual humana*

## 4.2 Aplicação do método preliminar

Para realização da metodologia proposta, foi necessário o uso de algumas ferramentas tecnológicas que serão apresentadas durante o desenvolvimento da pesquisa, sendo elas: *Oracle Virtual Box*<sup>5</sup>, *Excel 2010*<sup>6</sup>, *Palavras*<sup>7</sup>, *SVMLight*<sup>8</sup>, a *linguagem PHP*<sup>9</sup> e, por último, o *MS Visual Studio*<sup>10</sup>. Abaixo, temos uma amostra dos sintagmas extraídos do *Palavras* e já exportados para o *Excel*.

<sup>5</sup> Mais informações podem ser encontradas no site: <<https://www.virtualbox.org/>>.

<sup>6</sup> Mais informações podem ser encontradas no site: <<http://www.microsoft.com.br/office>>.

<sup>7</sup> Mais informações podem ser encontradas no site: <[http://beta.visl.sdu.dk/constraint\\_grammar.html](http://beta.visl.sdu.dk/constraint_grammar.html)>.

<sup>8</sup> Mais informações podem ser encontradas no site: <<http://svmlight.joachims.org/>>.

<sup>9</sup> Mais informações podem ser encontradas no site: <<http://www.php.net/>>.

<sup>10</sup> Mais informações podem ser encontradas no site: <<http://www.microsoft.com/visualstudio>>.

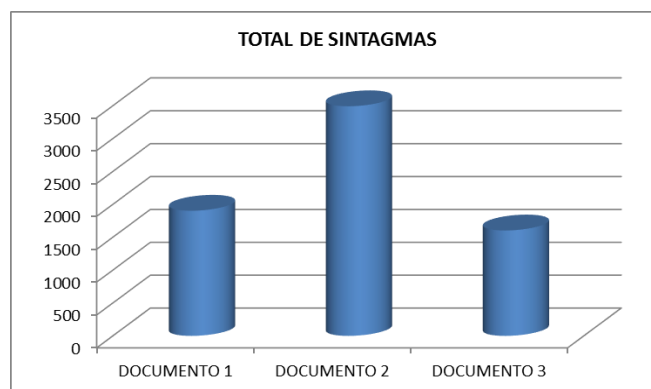
Figura 16 – Lista de sintagmas importados para o Excel

	A	B	C	D	E	F	G	H	I	J	K	L
1		A			*Inteligência Artificial							
2					*tarefas							
3					qualquer *esfera de _a atividade intelectual humana .							
4					_a *atividade intelectual humana .							
5					*Softwares							
6					*produtos intangíveis							
7					_o seu *processo de construção							
8					*construção							
9					*recursos intelectuais humanos							
10					sua *especificação							
11					sua *distribuição							
12					pleno *funcionamento .							
13					*meio de auxiliar							
14					*auxiliar							
15					o *processo de Engenharia de Software							
16					*Engenharia de Software							

Fonte: O próprio autor

O total de sintagmas detectados pelo *Palavras* em cada um dos documentos pode ser visto no gráfico 1:

Gráfico 1 – Total de sintagmas nos três primeiros documentos

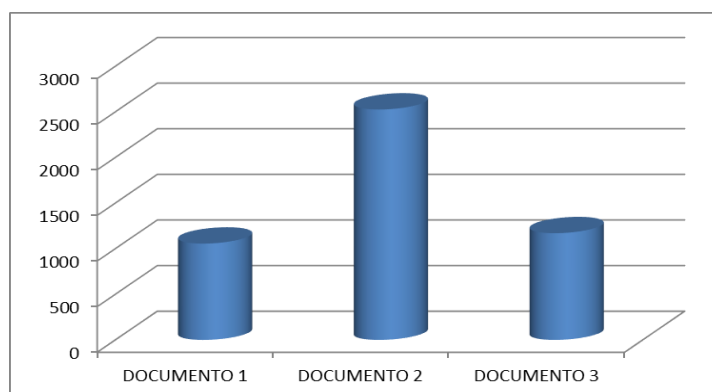


Fonte: O próprio autor

Após a normalização dos sintagmas, com a respectiva remoção de símbolos e de outros elementos que pudessem intervir na qualidade dos sintagmas, além da

contabilização dos sintagmas únicos e da eliminação do excesso de espaços no início, no meio ou no final do sintagma, temos então o seguinte resultado:

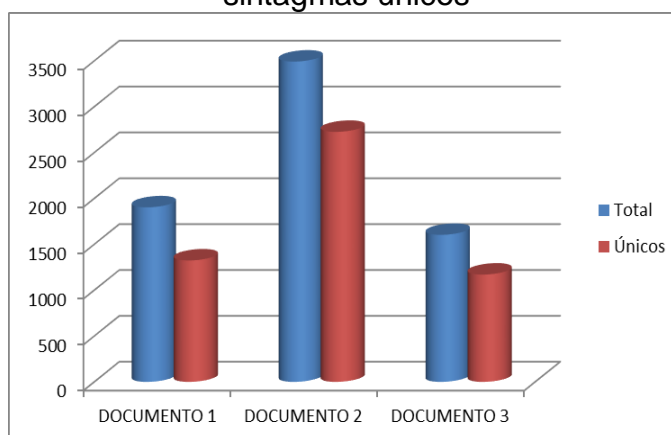
Gráfico 2 – Total de sintagmas após a normalização nos três primeiros documentos



Fonte: O próprio autor

Comparando a quantidade total de sintagmas com os sintagmas únicos encontrados em cada um dos textos, temos:

Gráfico 3 – Total de sintagmas nos três primeiros documentos comparando com os sintagmas únicos



Fonte: O próprio autor

Uma métrica importante é a quantidade de vezes em que determinados sintagmas ocorrem no corpo de um documento. Essa informação pode ser relevante para sistemas de classificação, já que sintagmas que se repetem ao longo de um ou mais documentos de um corpus podem frequentemente se tornar um descritor

importante nesse conjunto. Apesar de apresentada abaixo, essa abordagem não será considerada nesta tese.

O Doc1 ficou com a seguinte distribuição:

Tabela 3 – Quantidade de repetições de determinados sintagmas

<b>A abordagem de agentes</b>	<b>33</b>
<b>A abordagem reativa</b>	25
<b>a ação</b>	20
<b>a aplicação</b>	18
<b>a área</b>	13
<b>conhecimento especializado</b>	12
<b>a arquitetura</b>	11
<b>A arquitetura de o ambiente</b>	11
<b>a atividade</b>	10
<b>A arquitetura de um PSEE</b>	10
<b>a atividade intelectual humana</b>	8
<b>a Base de Conhecimento</b>	8
<b>BC</b>	
<b>a BC</b>	8
<b>a biblioteca</b>	7
<b>a capacidade</b>	7
<b>a capacidade de aprendizado de os agentes</b>	7

Fonte: O próprio autor

Já para o Doc2 temos:

Tabela 4 – Lista dos sintagmas que mais repetem

<b>Aprendizado</b>	<b>74</b>
<b>a rede</b>	58
<b>as redes neurais</b>	29
<b>conhecimentos</b>	27
<b>IA</b>	26
<b>Redes Neurais</b>	25
<b>o aprendizado</b>	20
<b>redes</b>	19
<b>exemplos</b>	17
<b>uma rede</b>	15
<b>as redes</b>	12
<b>conexionistas</b>	12
<b>inteligência</b>	11

<b>BackPropagation</b>	10
<b>a base de aprendizado</b>	9
<b>As redes conexionistas</b>	9
<b>As unidades</b>	9
<b>dados</b>	9
<b>ativação</b>	8
<b>este tipo de redes</b>	8
<b>o problema</b>	8
<b>rede</b>	8
<b>a atualidade</b>	7
<b>a rede neural</b>	7
<b>Classificação</b>	7
<b>Rumelhart</b>	7

Fonte: O próprio autor

A Tabela 5 apresenta os resultados para o Doc3:

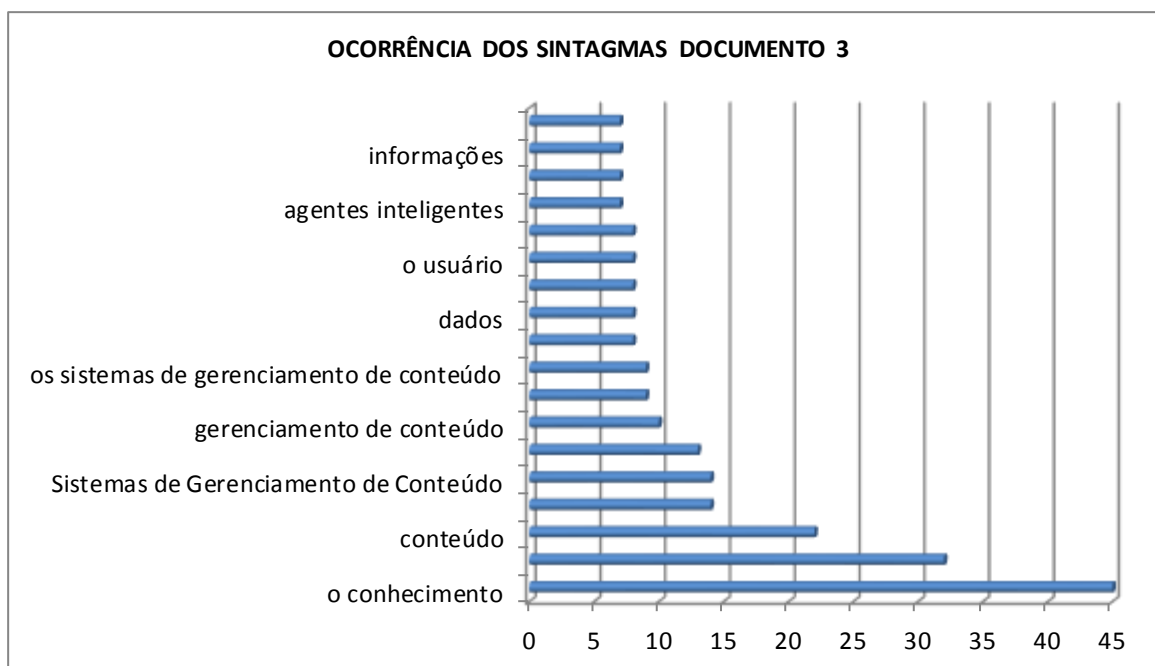
Tabela 5 – Lista dos sintagmas que mais repetem em doc3

<b>o conhecimento</b>	<b>45</b>
<b>conhecimento</b>	32
<b>conteúdo</b>	22
<b>as organizações</b>	14
<b>Sistemas de Gerenciamento de</b>	14
<b>Conteúdo</b>	
<b>a organização</b>	13
<b>gerenciamento de conteúdo</b>	10
<b>conhecimentos</b>	9
<b>os sistemas de gerenciamento de</b>	9
<b>conteúdo</b>	
<b>a rede</b>	8
<b>dados</b>	8
<b>GC</b>	8
<b>o usuário</b>	8
<b>problemas</b>	8
<b>agentes inteligentes</b>	7
<b>Aplicação da Inteligência Artificial</b>	7
<b>informações</b>	7
<b>Os SGC</b>	7

Fonte: O próprio autor

A título de curiosidade, a distribuição do Doc3 gerou o seguinte gráfico:

Gráfico 4 – Distribuição gráfica dos sintagmas para o Doc3



Fonte: O próprio autor

Os sintagmas que entraram nessa lista são os que apareceram, pelo menos, sete vezes no documento em questão. Esse limiar foi escolhido após a observação de que o número de repetições para o sintagma que mais aparece no documento, em relação ao segundo, tem uma queda abrupta. O mesmo acontece até o sétimo sintagma que mais aparece. A partir desse ponto, existe uma tendência em variar apenas minimamente o aparecimento do próximo sintagma, em relação aos anteriores (que apareceram mais vezes). Isso pode ser constatado no Gráfico 4. A partir do sintagma “gerenciamento de conteúdo”, ocorre uma estabilização no número de vezes que o sintagma se repete no texto. Após a extração dos sintagmas dos documentos apresentados, foi possível constatar que cerca de 10% dos sintagmas identificados se repetem pelo menos uma vez, número que pode ser influenciado por diversos fatores, dentre eles a forma como o autor escreve, o tipo de texto produzido e a mídia em que é publicado.

Como um dos objetivos desta tese foi demonstrar que sintagmas escritos de forma diferente podem conter a mesma carga informacional, ter tratado 100% dos sintagmas identificados de forma automática foi razoável, uma vez que mesmo sintagmas que aparecem apenas uma única vez podem ser relevantes para a classificação.

Como foi comentado acima, a avaliação da metodologia será realizada em dois momentos distintos: no primeiro momento será avaliado o quanto dois documentos podem se tornar estruturalmente semelhantes, aumentando a possibilidade de serem classificados em uma mesma classe, considerando-se para essa comparação a totalidade dos sintagmas; e, no segundo momento, serão utilizados contraexemplos para avaliar o quanto o mesmo processo poderia deixar documentos de temas diferentes mais próximos entre eles, o que deveria ser evitado ao se utilizar o método proposto. A seguinte formula será utilizada para realizar essa comparação:

$$TxSemelhanca = (\text{Sintagma } D_n \cap \text{Sintagma } D_x) / \text{Sintagma } D_n$$

Quanto maior o número de sintagmas semelhantes, maior será a pontuação (txSemelhança) atribuída durante a comparação dos documentos envolvidos. A fórmula baseia-se em se encontrar o número de sintagmas semelhantes entre dois documentos comparados para em seguida dividir este valor pelo total de sintagmas no primeiro documento sendo comparado  $D_n$ .

#### 4.2.1 Comparação direta entre sintagmas dos documentos

A primeira comparação entre os documentos foi realizada de forma direta, entre os sintagmas identificados automaticamente nos três documentos tratados. Foram considerados todos os sintagmas, independentemente do número de vezes em que eles apareceram no documento. Como o objetivo deste trabalho é melhorar a qualidade informacional do sintagma, não foi utilizado nenhum tipo de comparação com métodos que utilizem sintagmas ponderados (ROCHA, 2004) ou termos (MAIA, 2010).

A Tabela 7 mostra as comparações realizadas entre o par de documento Doc1 e Doc2:

Tabela 7 – Comparando Doc1 e Doc2

Total de sintagmas		Sintagmas únicos		Doc1/Doc2	Doc2/Doc1	Sintagmas idênticos
Doc1	Doc2	Doc 1	Doc2			
1903	3492	1324	2524	6,64%	3,48%	88

Fonte: O próprio autor

**Total de sintagmas** apresenta a quantidade total dos sintagmas identificados nos dois documentos, sendo comparados. **Sintagmas únicos**: representa o total dos sintagmas após o processo de remoção dos sintagmas duplicados, que representaram, na média, 10% do total dos sintagmas encontrados nos textos. **Doc1/Doc2**: percentual da quantidade de sintagmas presente no documento 1 e que também apareceram no documento 2. No caso acima, 6,6% do total dos sintagmas únicos do documento 1 também apareciam na lista de sintagmas únicos do documento 2. **Doc2/Doc1**: cálculo do percentual da quantidade de sintagmas presente no documento 2 e que também aparecem no documento 1. Trata-se de um valor menor do que 6,6% pela quantidade maior de sintagmas presentes no documento 2. **Total de sintagmas semelhantes**: valor total de sintagmas semelhantes entre os documentos após a comparação direta entre eles, sem levar em consideração nenhum tipo de tratamento a priori.

O mesmo processo acima foi realizado mais três vezes com os seguintes resultados apresentados nas tabelas 8 e 9:

Tabela 8 – Comparando Doc1 e Doc3

Total de sintagmas		Sintagmas únicos		Doc1/Doc3	Doc3/Doc1	Sintagmas idênticos
Doc1	Doc3	Doc 1	Doc3			
1903	1603	1324	1067	7,7%	9,6%	103

Fonte: O próprio autor

Tabela 9 – Comparando Doc2 e Doc3.

Total de sintagmas		Sintagmas únicos		Doc2/Doc3	Doc3/Doc2	Sintagmas idênticos
Doc2	Doc3	Doc2	Doc3			
3492	1603	2524	1067	4,4%	10,4%	112

Fonte: O próprio autor

Após as comparações entre os documentos, encontramos o resultado seguinte:



Tabela 10 – Comparação entre todos os documentos

<b>Documento</b>	<b>Doc1</b>	<b>Doc2</b>	<b>Doc3</b>
<b>Doc1</b>	100%	6,6%	7,7%
<b>Doc2</b>	3,4%	100%	4,4%
<b>Doc3</b>	9,6%	10,4%	100%

Fonte: O próprio autor

Salvo alguns problemas com a própria identificação indevida da estrutura como sendo um sintagma, quando não deveria ser classificado como tal, ou ainda problemas de não se identificar um sintagma quando o mesmo seria um sintagma, no geral, a identificação foi satisfatória, uma vez que se trata de um processo complexo e que precisa levar em consideração uma base muito dinâmica, que é a própria língua. Esses detalhes serão explicados no final deste capítulo.

#### 4.2.2 Comparação após remoção dos quantificadores

Identificaremos, nas próximas seções, formas de tornar semelhantes os sintagmas que estão estruturalmente diferentes, mas que podem contar com uma carga informacional semelhante. Obviamente que tanto pode haver perda como ganho, no que diz respeito à possibilidade informacional do sintagma após sua transformação. Essa análise será feita de forma manual e pontual em cada sintagma modificado, tendo como objetivo avaliar o quanto a mudança na estrutura do sintagma pode impactar na comparação.

A primeira proposta para deixar os documentos mais semelhantes foi extrair os quantificadores encontrados nos sintagmas detectados. Após a remoção dos quantificadores presentes no início de cada sintagma, os documentos Doc1 e Doc2 passaram de 88 para 89 sintagmas semelhantes, o que não representou uma grande melhora na semelhança entre os documentos. Os seguintes quantificadores foram removidos de forma automática: A, O, a, o, As, Os, as, os, Um, um, uma, Uma, Essa, Esse, Este, Esta, Essas, Esses, Estas, Estes.

O código em *Excel* para realização dessa tarefa está apresentado no Anexo C. A tabela abaixo apresenta os resultados até a execução desta etapa (remoção dos quantificadores):

Tabela 11 – Comparação após remoção dos quantificadores

Total de sintagmas		Sintagmas únicos		Doc1/Doc2	Doc2/Doc1	Sintagmas idênticos
Doc1	Doc2	Doc 1	Doc2			
1903	3492	1324	2524	6,72%	3,4%	89

Fonte: O próprio autor

Para fins de análise, a lista gerada pelo algoritmo em *Excel* foi verificada manualmente, após a remoção dos quantificadores, com o objetivo de encontrar estruturas que pudessem ter perdido sua capacidade informativa devido à transformação automática realizada.

Salvo alguns problemas com a própria identificação indevida da estrutura como sendo um sintagma, a lista se manteve coerente com o assunto tratado no corpo do documento, já que a maior parte dela é composta por sintagmas aplicáveis a qualquer documento que trate do assunto de Inteligência Artificial.

A hipótese de melhora nos sintagmas semelhantes não foi observada em nenhuma das comparações realizadas. Portanto, a remoção ou não dos quantificadores não implica em mudanças nos percentuais. Essa remoção dos quantificadores aparentemente não gera melhora na semelhança dos sintagmas e, assim, não precisará ser necessariamente utilizada na etapa consolidada. A seguir, apresentamos os resultados dos cálculos após a aplicação do processo de *stemming*.

#### 4.2.3 Comparação após o processo de *stemming*

Partimos para uma nova etapa de comparação após a realização do processo de *stemming*. Repetiremos cada um dos testes feitos anteriormente entre os três documentos, mas agora após a realização desse processo, como pode ser visto abaixo, no trecho da lista de sintagmas retirados do Doc1.

##### **STEMMING**

*adaptaçã*  
*adaptaçã red*  
*adaptaçã pes*  
*adaptaçã pes red*  
*adaptaçã pes red neur*

##### **SINTAGMA**

*adaptação*  
*adaptação de a rede*  
*adaptação de os pesos*  
*adaptação de os pesos de a rede*  
*adaptação de os pesos de uma rede neural*

<i>adaptaçã seu pes sináp</i>	<i>adaptação de os seus pesos sinápticos</i>
<i>adaptiv line element</i>	<i>Adaptive Linear Element</i>
<i>adaptiv ressonanc theroy</i>	<i>Adaptive Ressonance Theroy criado</i>
<i>cri</i>	
<i>adiçã nov exempl</i>	<i>adição de novos exemplos</i>
<i>adiçã nov neurôni</i>	<i>adição de novos neurônios</i>
<i>adiçã nov protótíp</i>	<i>adição de novos protótipos</i>

Algumas observações pontuais nessa lista são interessantes: 1) O algoritmo trata português e inglês, portanto *Adaptive* foi convertido para *Adaptiv*; 2) A escrita no plural foi eliminada, tal como aconteceu em “*adaptação de os pesos*”, que passou a ser “*Adapataçã pes*”; 3) Acentuações foram mantidas: “*sinápticos*” foi convertido para “*sináp*”, portanto, caso o autor escreva alguma palavra sem acentuação, ela será considerada no momento da comparação, mesmo após o *stemming*.

Ao utilizar o sintagma após o processo de *stemming*, o grande ganho é em relação ao uso de palavras no plural e no singular e, em alguns casos, no tempo verbal. Mas outros ganhos, como uma redução da ordem de 30% do tamanho dos documentos, também são importantes, ao considerar questões ligadas ao custo computacional de processamento.

Portanto, vamos agora a uma comparação, após o uso do sintagma nominal com seu *stemming*.

Tabela 12 – Percentual de melhora com stemming

<b>Doc1/Doc2</b>	<b>SN semelhantes</b>	<b>SN com <i>stemming</i></b>	<b>% de melhora</b>
6,6%	89	101	13,48%

Fonte: O próprio autor

Nessa primeira comparação foi observada uma melhora na ordem de 15% na semelhança dos sintagmas. Uma observação manual mostrou que esta melhora aconteceu principalmente devido ao fato de que, agora, sintagmas no singular, presentes no Doc1, foram comparados a sintagmas no plural, presentes em Doc2, e vice versa. Abaixo, temos um resumo final da melhora que foi percebida.

Tabela 13 – Percentual de melhora com stemming

Primeiro documento	Segundo documento	SN sem tratamento	SN com <i>stemming</i>	% de melhora na semelhança
Doc1	Doc2	88	121	37,27%
Doc2	Doc3	112	157	40,18%
Doc1	Doc3	103	146	41,75%

Fonte: O próprio autor

Ficou evidente que o uso médio do *stemming* no lugar do sintagma sem tratamento aumentou em cerca de 39% o número de sintagmas coincidentes entre os documentos, o que confirma nossa hipótese de melhora da semelhança, por não mais considerar termos no plural ou no singular.

O uso desse processo mostra-se promissor em vários tipos de sistemas para recuperação da informação, nos quais há algum tipo de coincidência entre termos. No próximo tópico trataremos da comparação após a convergência de sinônimos encontrados nos sintagmas.

#### 4.2.4 Comparação após a convergência de sinônimos

Na seção que tratou dos fatores característicos da língua escrita, falamos que um dos aspectos que influenciam na comparação de documentos é o de que autores diferentes, ou até mesmo o mesmo autor em documentos diferentes, podem utilizar estruturas diferentes de acordo com o contexto, mas com a finalidade de dizer a mesma coisa.

Isso acontece rotineiramente com determinados adjetivos. Determinados tipos de texto exigem do autor um conhecimento maior da língua, ao passo que textos com a mesma carga informacional, mas escritos por outros autores com um vocabulário menos vasto, podem exigir apenas conhecimentos mais rústicos em relação à escrita.

A próxima hipótese baseia-se na ideia de que pode haver uma troca entre palavras que sejam sinônimas antes de se fazer a comparação entre os sintagmas. É provável que a melhora da semelhança não seja tão marcante quanto foi a do *stemming*, mas uma melhora razoável deverá ser observada.

Em tempo, uma questão deve ser levantada: por se tratar de um processo automático de troca de sinônimos, uma observação minuciosa foi realizada a fim de verificar uma possível perda informacional quando há substituição de uma palavra por outra.

Para a convergência de sinônimos foi utilizado o seguinte algoritmo:

1. Identifica o primeiro sintagma da lista no primeiro documento;
2. Procura por este sintagma em todos os sintagmas do documento comparado;
3. Caso o sintagma esteja na lista, então não ajusta o sinônimo;
4. Caso o sintagma não seja encontrado:
  - a. Identifica o referente no primeiro sintagma;
  - b. Identifica o pós-qualificador;
  - c. Localiza o próximo sinônimo para o qualificador;
  - d. Monta um novo sintagma ajustado;
  - e. Procura novamente na lista de sintagmas do doc2;
  - f. Caso encontre, então o ajuste do sinônimo funcionou;
  - g. Busca o próximo sintagma em Doc1.

O código desse algoritmo escrito em VBA para *Excel* pode ser visto na lista dos anexos. Foi realizado acesso automático à lista de sinônimos mantida pelo site <sinônimos.com.br><sup>11</sup> Abaixo, os resultados após a execução do algoritmo para convergência de sinônimos:

Tabela 14 – Percentual de melhora com convergência de sinônimos

Avaliação	SN idênticos	Após convergência de sinônimos	% de melhora em relação ao total de SN
Doc1/Doc2	88	+2	<1%

Avaliação	SN idênticos	Após convergência de sinônimos	% de melhora em relação ao total de SN
Doc2/Doc3	112	+1	<1%

Avaliação	SN idênticos	Após convergência de sinônimos	% de melhora em relação ao total de SN
Doc1/Doc3	113	+1	<1%

Fonte: O próprio autor

<sup>11</sup> Disponível em: <<http://www.dicio.com.br>>. Acesso em: 03 jan. 2014.

Após a convergência de sinônimos realizada, percebeu-se pouca melhora na semelhança entre os sintagmas dos documentos comparados. A partir de uma análise manual, foi comprovado que diversos fatores interferem nesse processamento. O primeiro fator observado é o não tratamento, pelo software de dicionário utilizado, de palavras no plural ou no singular e também a não identificação de gêneros diferentes. Por exemplo, a palavra “grandes” não foi identificada no dicionário, mas a palavra “grande” foi.

Para qualificadores formados por mais de um termo, é impossível o processamento pelo dicionário. É o caso do qualificador “cognitiva humana”, encontrado no sintagma “\*aprendizagem cognitiva humana”. Neste caso, o algoritmo em VBA consegue identificar o “cognitiva humana” como sendo um qualificador para “aprendizagem”, mas a identificação de sinônimos para “cognitiva humana” é uma tarefa que os dicionários atualmente não realizam.

Outros fatores que dependem do processamento sintático também foram observados, como no seguinte caso: “\*protótipo do ambiente” foi traduzido para “\*protótipo de o ambiente” pelo software *Palavras*. Dessa forma, o qualificador “do ambiente” passou a ser “de o ambiente”, o que também dificultou a identificação do qualificador “ambiente” pelos dicionários de sinônimos.

Portanto, optou-se por seguir adiante com os testes, mas sem o tratamento dos sinônimos pelos motivos apresentados.

#### **4.2.5 Contraprova**

Para validação dos resultados encontrados, foi necessário realizar uma contraprova denominada qualitativa, por ter como principal objetivo uma comparação criteriosa entre os três primeiros corpora da área de inteligência artificial com outros três documentos de outras áreas. O principal objetivo desta etapa foi verificar se a aplicação das etapas propostas em documentos de um mesmo assunto os aproxima em termos estruturais, o mesmo não acontecendo em documentos de temas diferentes.

Essa contraprova se mostra extremamente importante, pois o treinamento de sistemas automáticos classificadores funciona de maneira tão mais eficiente quanto

maior for a distância entre os dados utilizados para seu treinamento de classes diferentes e menor a distância entre os dados de corpus do mesmo tema.

Essa será a última análise qualitativa a ser realizada, na próxima etapa partiremos para um processo quantitativo de treinamento, utilizando máquinas de vetor de suporte, o que irá envolver o processamento de uma quantidade maior de documentos em domínios a serem definidos.

Para esta contraprova foram utilizados três documentos de áreas diferentes do conhecimento. Cada um deles será comparado com os outros três que foram utilizados na primeira etapa do processo.

Os seguintes documentos foram utilizados:

Tabela 15 – Origem dos documentos utilizados

Área	Título	Autor	Origem
Educação	Desafios da construção de um saber informacional na educação à distância	Magdalena José Avena e Ivete Pieruccini	Scielo
História	O sacerdócio goiano. Celibato e historiografia	Maria da Conceição Silva; Wellington Moreira Coelho	Scielo
Robótica	Aprendizado em informática de forma lúdica	Renato Ferreira Soares, Marcos Augusto Francisco Borges	Scielo

Fonte: O próprio autor

Os resumos podem ser vistos a seguir:

**DocP1 - O sacerdócio goiano. Celibato e historiografia**

A historiografia, de um modo geral, tipificou por desvios morais as conjugalidades constituídas por clérigos durante o século XIX. Ao fazer isto, ela absorveu em seu discurso historiográfico a visão da Igreja Católica, que ponderou o sacerdócio pelo normativo do celibato. A esse respeito, pode-se notar que a historiografia goiana não ficou alhures a esta influência.

Palavras-chave: sacerdócio, Igreja Católica, família

**DocP2 - Desafios da construção de um saber informacional na educação a distância**

O artigo discute o papel do ambiente de aprendizagem virtual, constituído de informações e espaço de trocas de experiências, destinado à apropriação de processos e práticas de pesquisa, por educadores que atuam em contextos de educação não formal. Analisa novas abordagens acerca da prática da pesquisa na construção de conhecimento e diferentes dispositivos não presenciais

voltados ao Aprender a Pesquisar, categoria que integra questões tratadas pela Infoeducação. O objeto empírico da investigação foi desenvolvido no âmbito das ações da Estação do Conhecimento Einstein, dispositivo cultural, implantado sob a orientação do Colaboratório de Infoeducação – Colabori -, da Escola de Comunicações e Artes, da Universidade de São Paulo. Conclui que a concomitância entre a realização de oficinas presenciais de aprendizagem da pesquisa e os dispositivos virtuais de aprender a pesquisar é um caminho a ser conquistado para a criação de condições favoráveis à autonomia e participação dos educadores nesse campo.

### **DocP3 - Aprendizado em informática de forma lúdica**

Este resumo de artigo apresenta a robótica como uma ferramenta para o auxílio no aprendizado em informática. O artigo apresenta experiências de uso da robótica e os resultados obtidos.

Apesar de pertencerem a áreas diferentes, para a utilização como contraprova, tomou-se o cuidado de selecionar documentos que pudessem conter certa semelhança estrutural com os três primeiros.

Não foram percebidos vários termos como “educação”, “treinamento”, “aprendizagem”, que coincidiam com termos na área de Inteligência Artificial. No caso de robótica, também foram encontrados termos como “agentes”, “algoritmos” etc., que foram semelhantes aos encontrados na área de IA. O artigo da área de História foi escolhido de forma aleatória, mas também com a intenção de perceber o quanto os documentos poderiam se tornar semelhantes.

## **4.2.6 Comparações da contraprova**

### **4.2.6.1 Comparação do Doc1 do treinamento e doc1 da contraprova**

A primeira comparação entre os documentos foi realizada de forma direta entre os sintagmas identificados automaticamente no primeiro documento do treinamento e no primeiro documento da contraprova. Também foram considerados todos os sintagmas, independentemente do número de vezes em que apareceram no documento.

A Tabela 16 mostra as comparações realizadas entre pares de documentos:



Tabela 16 – Comparação direta documento 1 do treino e documento1 da prova

Número total de sintagmas identificados		Sintagmas únicos após a remoção de duplicidades		Sintagmas presentes no primeiro Doc1 e também no segundo DocP1	Sintagmas presentes em Doc1 e também em DocP1	Total de sintagmas semelhantes
Doc1	DocP1	Doc 1	DocP1			
1903	1721	1324	1714	1,5%	1,2%	21

Fonte: O próprio autor

O DocP1 da contraprova trata de História Brasileira e os únicos sintagmas semelhantes versavam sobre sociedade, valores, comportamento e regras, termos estes que são semelhantes em diversas áreas do conhecimento.

A seguir, temos uma comparação após o processo de *stemming*. O comportamento esperado, o de uma melhora insignificante nos percentuais, foi comprovado.

#### 4.2.6.2 Comparação do Doc2 treinamento com DocP2 da contraprova

Já o segundo documento da contraprova trata do assunto “Robótica”, área esta que por vezes trata do tema Inteligência Artificial. Portanto, é de se esperar um maior número de sintagmas idênticos, ao se comparar o Doc2 com o DocP2 utilizado no treinamento. Os resultados foram:

Tabela 17 – Comparação direta entre teste e contraprova (Doc2 e DocP2)

Número total de sintagmas identificados		Sintagmas únicos após a remoção de duplicidades		Sintagmas presentes no primeiro Doc2 e também no segundo DocP2	Sintagmas presentes em Doc2 e também em DocP2	Total de sintagmas semelhantes
Doc2	DocP2	Doc 2	DocP2			
3492	439	2524	367	1,8%	6,5%	24

Fonte: O próprio autor

O documento 2 utilizado na contraprova possuía apenas 367 sintagmas únicos, portanto o percentual de 6,5% de semelhança torna-se razoável. Porém, apenas 24 sintagmas foram identificados como semelhantes entre os documentos e versavam sobre: aprendizado, aprendizagem, algoritmo e programação, dentre

outros. Esses termos são relevantes para as áreas comparadas nos dois documentos, que são: IA, nos documentos de treinamento, e Robótica, no segundo do grupo de contraprova.

#### 4.2.6.3 Comparação direta Doc3 treinamento com DocP3 da contraprova

A terceira comparação direta contou com um documento da área de processos de aprendizagem humana. Portanto, era esperada a existência de termos coincidentes no momento da contraprova, o que ficou demonstrado pelos resultados da comparação direta:

Tabela 18 – Comparação direta entre teste e contraprova (Doc3 e DocP3)

Número total de sintagmas identificados		Sintagmas únicos após a remoção de duplicidades		Sintagmas presentes no primeiro Doc3 e também no segundo DocP3	Sintagmas presentes em Doc3 e também em DocP3	Total de sintagmas semelhantes
Doc3	DocP3	Doc 3	DocP3			
1603	1804	1067	1426	3,7%	2,8%	40

Fonte: O próprio autor

Dentre os termos coincidentes temos: trabalho, processo, controle, informação, conhecimento, dados e habilidades.

Até este ponto podemos perceber que os documentos da contraprova são estruturalmente menos semelhantes do que os documentos que foram utilizados na primeira etapa, em que todos versavam sobre inteligência artificial.

Na continuação da contraprova, faremos os mesmos testes anteriores, porém agora com a stemmização dos sintagmas e a remoção dos quantificadores. Trata-se de um teste crucial, uma vez que este processo não poderá tornar os documentos muito mais semelhantes do que já estão.

#### 4.2.6.4 Contraprova após *stemming* do Doc1 de treino com DocP1 da contraprova

Nesta etapa foi realizada a comparação dos documentos, após o processo de *stemming* dos documentos da contraprova, uma vez finalizado o processo de *stemming* dos documentos iniciais.

As tabelas 19,20 e 21 apresentam os resultados obtidos:

Tabela 19 – Comparação direta entre teste e contraprova (Doc1 e DocP1)

Doc1/DocP1	Sintagmas semelhantes	Semelhantes após <i>stemming</i>	% de melhora
1,5%	21	22	0,1%

Fonte: O próprio autor

#### 4.2.6.5 Contraprova após *stemming* do Doc2 treino com o DocP2 da contraprova

Tabela 20 – Comparação direta entre Doc2 e DocP2

Doc2/DocP2	Sintagmas semelhantes	Sintagmas com <i>stemming</i>	% de melhora
1,8%	24	24	0,0%

Fonte: O próprio autor

#### 4.2.6.6 Contraprova após *stemming* do Doc3 treino com o DocP3 da contraprova

Tabela 21 – Comparação direta entre Doc3 e Docp3

Doc3/DocP3	Sintagmas semelhantes	Sintagmas com <i>stemming</i>	% de melhora
3,7%	40	43	0,1%

Fonte: O próprio autor

Observou-se que o processo de *stemming* dos documentos com temas diferentes não trouxe significativas quantidades de sintagmas idênticos após o processo. Ficou comprovada a hipótese de que o processo de *stemming* não interfere no aumento da semelhança entre documentos de temas diferentes, mas é

sensível à melhora após o processo ser realizado em documentos que tratam do mesmo tema.

Uma análise mais cuidadosa dos resultados obtidos até este momento será apresentada na próxima seção. A seguir, será apresentado o processo de classificação dos documentos com a metodologia consolidada.

#### **4.2.7 Avaliação dos resultados da prospecção**

Nesta seção detalhamos os resultados obtidos até este momento da pesquisa. O objetivo será evidenciar os resultados obtidos e a metodologia final a ser utilizada na classificação de documentos, utilizando sintagmas após o processo de extração dos quantificadores e do processo de *stemming*.

Nesta primeira parte da pesquisa, foram realizados testes qualitativos no tratamento e no uso dos sintagmas em seis documentos diferentes. O primeiro grupo de documentos foi composto por três documentos que tratavam do tema “inteligência artificial”. O segundo grupo de documentos, utilizados para contraprova, foi composto por três documentos de áreas distintas, sendo elas Educação, História e Robótica.

Não foi objetivo, nesta etapa, validar a qualidade dos sintagmas extraídos pelo processador *Palavras*. Entretanto, em uma análise superficial, foram detectados vários problemas na identificação correta dos sintagmas. Isso decorre de vários fatores, até mesmo do fato de que, em alguns momentos, a identificação de sintagmas depende da compreensão do texto e não apenas da informação sobre a sintaxe extraída. Para uma maior eficiência do processo de classificação de documentos, faz-se necessária a identificação totalmente automática dos sintagmas, dado o volume de documentos a ser utilizado.

Uma análise de todos os sintagmas extraídos para treinamento de um sistema automático demandaria muito tempo, além de ser um trabalho interdisciplinar, pois dependeria de um especialista em linguística para classificar o grau de qualidade dos sintagmas extraídos pelo PLN. Esse não foi o objetivo desta pesquisa, mas poderia ser alvo de uma sequência de estudos a partir da metodologia proposta nesta tese.

Com relação ao tempo necessário para a extração dos sintagmas, observou-se que o tempo médio para extração de 1000 (mil) sintagmas foi de

aproximadamente 2 minutos, utilizando uma máquina Intel Core i3 com 4GB de memória RAM. Este tempo pode ser reduzido drasticamente se máquinas com múltiplos processadores, equipamento comum em centros de processamento especializados em PLN, forem utilizadas. Os tempos de preparação dos documentos, com conversão de PDF em texto, e das comparações, realizadas no software *Excel*, foram mínimos, não interferindo no tempo final do processo.

Verificou-se que o *Palavras* não identificou corretamente siglas utilizadas em alguns textos, por exemplo, nos textos sobre Inteligência Artificial era comum encontrar siglas como IA, Sistemas Baseados em Casos (SBC) e Sistemas Baseados em Regras (SBR), dentre outros. Como, em algumas áreas de temática mais técnica, o uso de siglas se tornou comum, o tratamento de siglas talvez apresente uma melhora no sistema de classificação com a conversão de todas as siglas em texto plano, o que demandaria o uso de um dicionário de siglas, estando fora do escopo desta tese.

Mesmo levando-se em consideração os problemas identificados o processo de extração automática se mostra adequado, uma vez que se observou que a extração equivocada de um sintagma em determinado texto ocorreu nos demais textos também, ao se tratar do mesmo sintagma. Obviamente que o tratamento automático do sintagma leva muito menos tempo do que uma extração manual, o que viabiliza a classificação automática dos documentos. Uma extração manual, ou até mesmo uma verificação manual posterior, inviabilizaria todo o processo.

O processo de substituição de sinônimos não surtiu o efeito esperado. Isso se deve principalmente à nossa limitação tecnológica para encontrar e substituir o termo de forma satisfatória.

## 5 METODOLOGIA CONSOLIDADA

Ficou evidente na etapa de prospecção que o processo de *stemming* pode aumentar a semelhança entre documentos que tratam do mesmo assunto, sem aumentar de forma significativa a semelhança entre documentos de temáticas diferentes.

Entretanto, apenas uma comparação qualitativa pode não ser suficiente para demonstrar a eficácia do processo. Desta forma tornou-se necessário o uso de uma ferramenta automática para cálculo das semelhanças entre os documentos.

Para essa demonstração, o número de temas (classes) e a quantidade de documentos aumentaram substancialmente, tendo em vista que sistemas classificadores baseados em treinamento tornam-se mais eficientes na medida em que são submetidos a uma quantidade maior de documentos durante a etapa de treinamento.

Neste capítulo serão discutidos os corpora utilizados e, em seguida, será apresentada a metodologia para extração e preparação dos sintagmas, tendo como base os resultados obtidos no capítulo anterior. Em seguida, será apresentada a ferramenta *SVMLight*, utilizada para o treinamento da máquina de classificação automática e, na sequência, uma discussão dos resultados encontrados.

Em resumo, a metodologia consolidada irá compreender:

- 1) Preparação dos documentos:
  - a. Conversão de PDF para texto;
  - b. PLN e extração dos SN;
  - c. Tratamento da lista de SN;
- 2) Preparação dos sintagmas:
  - a. Extração dos quantificadores;
  - b. Stemmização;
  - c. Separação dos sintagmas únicos após *stemming*;
- 3) Treinamento da SVM:
  - a. Quantificação dos SN;
  - b. Submissão ao treinamento;
  - c. Avaliação do aprendizado.

## 5.1 Preparação dos corpora

Os documentos incluídos nos corpora foram retirados de revistas publicadas no site Scielo. A escolha por este site foi feita por se tratar de um importante portal de divulgação científica disponível no Brasil, também recebendo artigos científicos de revistas sulamericanas e da região do Caribe. Os documentos foram separados em dois grupos:

- a) Para a etapa de treinamento, os documentos que formaram o primeiro corpus foram compostos por: 19 documentos da área de Engenharia, 13 documentos relativos à área de História e 14 documentos da área de Letras;
- b) O segundo corpus, escolhido para validar o treinamento da máquina classificadora nesta pesquisa de classificação automática, é composto por: 4 documentos da área de Engenharia, 5 documentos da área de História e 5 documentos da área de Letras.

Para escolha dos documentos foram tomados os devidos cuidados para que eles contivessem uma quantidade razoável de sintagmas e para que versassem sobre assuntos diferentes dentro da área, além de serem documentos coletados em diversas épocas dos últimos dez anos, o que foi facilitado pela ferramenta de busca do Scielo, que já separa os periódicos em ordem cronológica. Também foi observado o fato de se tratarem de artigos escritos por autores diferentes, evitando assim vícios de linguagem e possíveis repetições de temas.

Para a área de História as publicações escolhidas foram retiradas da Revista *Varia História*, publicada pelo Departamento de História da UFMG. A escolha por esta revista foi feita por se tratar de uma renomada publicação da área de Historiografia brasileira. Para a Engenharia foi escolhida a revista publicada pela Associação Brasileira de Soldagem. A escolha por essa revista foi feita por se tratar de publicação bem qualificada pelo programa Qualis da CAPES. Por último, foram escolhidos os documentos para compor o tema de Linguística.

Como é possível perceber, os documentos e as áreas escolhidas são relativamente distantes em suas temáticas, apesar de o campo da História apresentar, por vezes, textos de conhecimento que perpassam outras áreas. Observamos que os textos utilizados na área de História versavam sobre História da

Economia e História da Educação, e traziam uma carga de sintagmas que também ocorrem com frequência em outras áreas.

A escolha de três contextos específicos justifica-se pelo próprio fato de ser a máquina classificadora uma divisora de classes que pode trabalhar com uma ou N classes, bastam pequenos ajustes durante o treinamento. Além desses, outros critérios foram:

- Quantidade compatível com a capacidade de processamento disponível;
- Quantidade razoável para uma verificação superficial e manual da qualidade dos sintagmas extraídos;
- Atualidade dos textos, mas sem serem influenciados por termos que pudessem ter caído em desuso ao longo dos anos;
- Afinidade com as temáticas tratadas.

A quantidade também foi estabelecida após a realização dos primeiros testes de convergência no aprendizado do sistema.

A análise final dos resultados na quantidade de documentos utilizados também foi condizente com os resultados esperados, o que será relatado no capítulo de conclusões. A tabela 22 apresenta a listagem total dos documentos utilizados para treinamento e avaliação da metodologia proposta.

Tabela 22 – Total de documentos utilizados para treinamento

<b>TEMÁTICA</b>	<b>DOCS UTILIZADOS</b>
Engenharia	19
História	13
Letras	14
<b>TOTAL</b>	<b>46</b>

Fonte: O próprio autor

Após o processo de treinamento com os documentos acima definidos, foi dado início ao procedimento de testes com novos documentos sobre as mesmas temáticas anteriormente citadas. Os documentos são:



Tabela 23 – Total de documentos utilizados para teste

TEMÁTICA	DOCS UTILIZADOS
Engenharia	4
História	5
Letras	5
<b>TOTAL</b>	<b>14</b>

Fonte: O próprio autor

Como dito anteriormente, as principais etapas do processo foram:

### 5.1.1 Etapa 1a: Conversão de PDF para texto puro

Esta é a primeira etapa do trabalho e consiste em converter os documentos que compõem o *corpus* para um formato mais apropriado ao processamento pelo software de treinamento, que utiliza textos no padrão ASCII armazenados no formato UNICODE. A aparente trivialidade dessa tarefa pode esconder desafios, como no caso de imagens e textos em formato gráfico, que podem confundir o software responsável pelo processo de conversão. Isso inclui caracteres em outras línguas, metadados e acentuação (muitos dos softwares de conversão não atendem plenamente a língua portuguesa), além de sinais matemáticos que não são convertidos, como observado nos artigos utilizados que tratavam da área de Engenharia.

Percebeu-se que o próprio software *Adobe Reader*, na sua versão 11.0 lançada no ano de 2012, gerava uma boa saída em TXT, não sendo necessário nenhum outro software proprietário para realização desse processo.

Como a quantidade de documentos aumentou substancialmente nesta etapa, procedimentos que haviam sido feitos anteriormente não foram realizados neste momento, como a remoção de determinadas palavras: abstract, resumo, figura X e conclusões. Na etapa anterior isso foi realizado manualmente e seria inviável neste momento, já que essa remoção depende da posição do termo no texto. A palavra “Resumo”, por exemplo, quando aparece logo abaixo do título ou logo nas primeiras seções do texto, pode representar apenas uma padronização na forma de produzir o artigo. Mas a palavra “Resumo” também pode aparecer em outras posições do texto do artigo e, nesse caso, ela não deveria ser eliminada. Portanto, optou-se por manter essas palavras e realizar o treinamento com todas as palavras do texto.

Outros procedimentos, como separação de sentenças, identificação de siglas e identificação de orações, ficaram a cargo do software *Palavras* quando este foi utilizado para separar os sintagmas nominais.

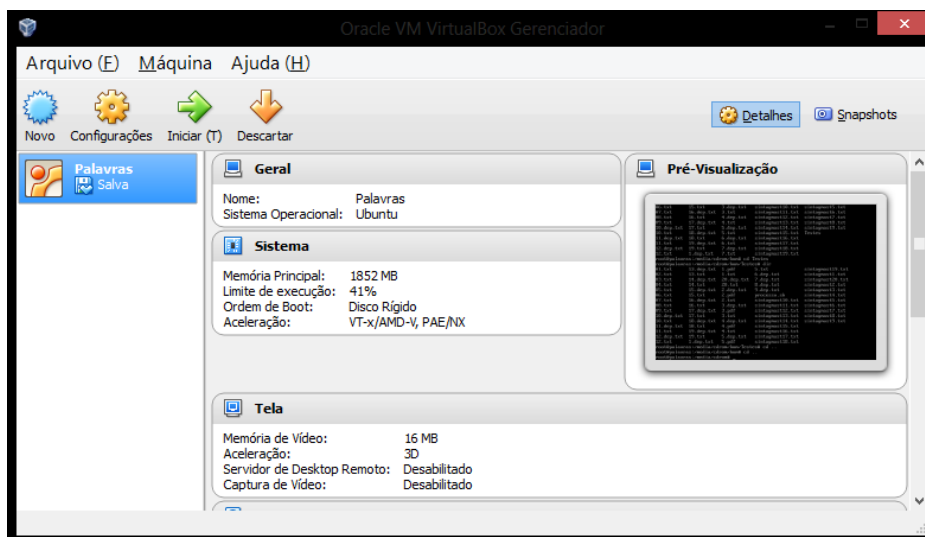
A remoção de *stop words*, algo comum em vários trabalhos nesta área de pesquisa, também não foi considerado, pois impactava a qualidade final dos sintagmas, podendo mudar o seu sentido final.

### 5.1.2 Etapa 1b: PLN e extração dos sintagmas

Ao contrário da etapa anterior, em que apenas seis documentos foram processados, agora temos um conjunto muito maior de documentos e algumas etapas foram automatizadas, a fim de viabilizar o processo de extração dos sintagmas.

A Figura 17 mostra o servidor virtual *Linux Ubuntu* montado utilizando o software *Oracle Virtual Box*, no qual toda a parte de PLN foi realizada:

Figura 17 – Tela do software Oracle Virtual Box



Fonte: O próprio autor

O uso deste servidor virtual facilitou sobremaneira o nosso trabalho, pois ele pode rodar sob uma plataforma *Windows*, onde o restante do processo está sendo realizado em paralelo. No caso, a etapa de treinamento que será vista nas próximas seções foram realizadas em *Java/Windows*.

Abaixo, temos o comando utilizado para automação do processamento pelo *Palavras*.

Figura 18 – Arquivos processados pelo software *Palavras*

```

Máquina  Visualizar  Dispositivos  Ajuda
06.txt    15.txt      3.dep.txt    sintagmast10.txt  sintagmast5.txt
07.txt    16.dep.txt  3.txt        sintagmast11.txt  sintagmast6.txt
08.txt    16.txt      4.dep.txt    sintagmast12.txt  sintagmast7.txt
09.txt    17.dep.txt  4.txt        sintagmast13.txt  sintagmast8.txt
10.dep.txt 17.txt      5.dep.txt    sintagmast14.txt  sintagmast9.txt
10.txt    18.dep.txt  5.txt        sintagmast15.txt  Testes
11.dep.txt 18.txt      6.dep.txt    sintagmast16.txt
11.txt    19.dep.txt  6.txt        sintagmast17.txt
12.dep.txt 19.txt      7.dep.txt    sintagmast18.txt
12.txt    1.dep.txt   7.txt        sintagmast19.txt

root@palavras:/media/cdrom/hum# cd Testes
root@palavras:/media/cdrom/hum/Testes# dir
01.txt    13.dep.txt  1.pdf        5.txt          sintagmast19.txt
02.txt    13.txt      1.txt        6.dep.txt      sintagmast1.txt
03.txt    14.dep.txt  20.dep.txt   7.dep.txt      sintagmast20.txt
04.txt    14.txt      20.txt       8.dep.txt      sintagmast2.txt
05.txt    15.dep.txt  2.dep.txt    9.dep.txt      sintagmast3.txt
06.txt    15.txt      2.pdf        processa.sh     sintagmast4.txt
07.txt    16.dep.txt  2.txt        sintagmast10.txt  sintagmast5.txt
08.txt    16.txt      3.dep.txt    sintagmast11.txt  sintagmast6.txt
09.txt    17.dep.txt  3.pdf        sintagmast12.txt  sintagmast7.txt
10.dep.txt 17.txt      3.txt        sintagmast13.txt  sintagmast8.txt
10.txt    18.dep.txt  4.dep.txt    sintagmast14.txt  sintagmast9.txt
11.dep.txt 18.txt      4.pdf        sintagmast15.txt
11.txt    19.dep.txt  4.txt        sintagmast16.txt
12.dep.txt 19.txt      5.dep.txt    sintagmast17.txt
12.txt    1.dep.txt   5.pdf        sintagmast18.txt

root@palavras:/media/cdrom/hum/Testes# cd ..
root@palavras:/media/cdrom/hum# cd ..
root@palavras:/media/cdrom/hum# ./processa.sh

```

Fonte: O próprio autor

O *script* de comandos *processa.sh* possuiu duas linhas de comando semelhantes aos exibidos abaixo, porém replicados pelos N documentos contidos em cada pasta com documentos dos corpus:

```
cat 1.txt | /opt/palavras/por.pl --dep> 1.dep.txt
```

```
cat 1.dep.txt | /media/cdrom/extract_np.pl > sintagmast1.txt > 01.txt
```

Após a realização de todo o processamento pelo *Palavras*, temos como saída o arquivo 01.txt, com todos os sintagmas identificados e separados linha a linha, como mostrado no trecho abaixo:

01.txt:

\_a \*soldagem robotizada

\*métodos

\*engenharia de usabilidade .

\*usabilidade .

\*laboratório

Percebemos a existência de determinados caracteres como \_ (sublinha) e \* (asterisco), o que identifica os quantificadores e os referentes em cada sintagma. Essa lista precisa ser regularizada para uso no software de treinamento. Alguns problemas, como o surgimento de caracteres indevidos e a inserção de espaços dentro dos sintagmas, atrapalharam sobremaneira o trabalho de classificação. Em seguida, foi realizado um tratamento e uma preparação dessa lista.

### 5.1.3 Etapa 1c: Tratamento da lista final

O uso do software *Palavras* e de um script criado na linguagem Perl pelo próprio criador do software facilitou muito todo o trabalho. Entretanto, como foi perceptível na lista final gerada, diversos erros de identificação dos sintagmas foram identificados. Mas, de uma maneira geral, e por ser um recurso totalmente automático e indispensável para este tipo de proposta, o seu uso foi mantido, uma vez que poucos são os softwares que realizam esse tipo de processamento.

Entretanto, como se trata de uma etapa hermética que envolve receber o texto, realizar o processamento e, em seguida, devolver uma lista de sintagmas, ela poderá ser realizada em outros softwares ou utilizando-se de recursos que envolvam também processos manuais, de maneira a aumentar a qualidade dos sintagmas.

Para efeitos da nossa proposta nesta tese e pelos bons resultados obtidos na primeira etapa de testes com poucos documentos, resolveu-se manter apenas o software e o processamento automáticos, sem maiores intervenções manuais. Mesmo assim, foi criada uma rotina em PHP, que pode ser acessada no site <clasdoc.com><sup>12</sup> e que contém diversas microrrotinas necessárias ao ajuste do arquivo final.

---

<sup>12</sup> O site clasdoc.com foi registrado e criado pelo próprio autor desta tese para manter toda a programação necessária à realização do processamento necessário nesta tese e está disponível para uso público.

Figura 19 – Tela do software clasdoc criado pelo autor



Fonte: O próprio autor

#### 5.1.4 Etapa 2a: Extração dos quantificadores e qualificadores

Como foi percebido durante a realização das comparações na primeira etapa de testes com poucos documentos, os quantificadores e qualificadores dos sintagmas podem impactar demais na comparação SN a SN, com o objetivo de detectar semelhanças entre documentos. A lista abaixo apresenta exemplos desta comparação removidos dos documentos 01.txt e 05.txt, do grupo de documentos que versavam sobre soldagem:

##### **01.txt**

...

*O processo de soldagem*

*Aquele mecanismo utilizado*

*A máquina de treinamento*

...

##### **05.txt**

...

*Processo de soldagem*

*Este mecanismo utilizado*

*Máquina de treinamento*

...

Após a remoção dos quantificadores e qualificadores, a seguinte lista foi gerada para os SN acima:

<b>01.txt</b>	<b>05.txt</b>
...	...
<i>Processo de soldagem</i>	<i>Processo de soldagem</i>
<i>Mecanismo utilizado</i>	<i>mecanismo utilizado</i>
<i>Máquina de treinamento</i>	<i>Máquina de treinamento</i>
...	...

Pode-se perceber uma melhora significativa no grau de semelhança estrutural da lista de sintagmas. O seguinte script foi utilizado com esta finalidade:

```
function retiraQuantificador($frase) {
    $palavra = $frase;
    $posicaoEspaco = strpos($palavra, " ");
    $quant = substr($palavra, 0, $posicaoEspaco);
    $quantificadores = array("a", "o", "as", "os", "um", "uma", "uns", "umas", "aquele",
    "aquela", "aqueles", "aquelas", "este", "esta", "essa", "esse", "essas", "esses");
    if (in_array(strtolower($quant), $quantificadores))
        $palavra = substr($palavra, $posicaoEspaco + 1, strlen($palavra) -
        $posicaoEspaco);
    if ($palavra != "")
        return $palavra;
    return $frase;
    break; }
```

### 5.1.5 Etapa 2b: Conversão do sintagma em seu *Stemming*

Este talvez seja o ponto central de mudança no uso do sintagma para realização da classificação dos documentos. Percebeu-se na etapa anterior, com poucos documentos, que o uso do *Stemming* do sintagma, e não do sintagma original, poderia aumentar substancialmente a semelhança entre os documentos.

Para realização do *stemming* foi utilizada a ferramenta *PTStemming*<sup>13</sup>, escrita na linguagem C# e com licença de uso baseada em GNU. Foram feitos o download e a instalação dessa ferramenta no mesmo servidor classdoc.com, porém agora no

<sup>13</sup> A ferramenta pode ser acessada no endereço: <<https://code.google.com/p/ptstemmer/>>.

ambiente *Microsoft Framework*, o que foi necessário para processamento em .NET, uma vez que a versão utilizada estava em C#.

Para esse processamento foi necessário apenas o envio dos arquivos para o servidor clasdoc.com, por meio de um upload simples da lista de sintagmas e, ao pressionar o botão “Limpa” (disponível no site), toda a limpeza e separação dos *stemmings* foi realizada após alguns instantes de processamento. A lista gerada de *stemming* foi semelhante à apresentada a seguir:

#### ***Sintagma original***

1. *\*abordagens diferentes*
2. *\*Abstratos*
3. *\*acesso*
4. *\*acesso a os documentos*
5. *\*acesso a um processo de esta biblioteca*
6. *\*ações*
7. *\*ações elementares*
8. *\*ações passadas*

#### ***Sintagma com stemming***

*abord difer*  
*abstrat*  
*acess*  
*acess document*  
*acess process est bibliotec*  
*açõ*  
*açõ element*  
*açõ pass*

Aqui é possível perceber a grande diferença ao utilizar o *stemming*. Vejamos o sintagma “*abordagens diferentes*”, em que após a conversão temos “*abord difer*”: ao se utilizar o *stemming* do sintagma, estamos automaticamente eliminando a questão da escrita do sintagma ou de seu plural. Se estivéssemos utilizando sintagmas verbais, até o tempo verbal poderia ser eliminado com esse processo.

#### **5.1.6 Etapa 2c: Separação dos *stemmings* únicos**

Após a realização do *stemming* e a consequente geração da lista de sintagmas para uso no treinamento, foi necessária uma nova remoção das duplicatas, já que sintagmas não semelhantes em um mesmo documento ficaram semelhantes após o processo de *stemming*.

Esse procedimento também foi realizado pelo servidor clasdoc.com através do mesmo botão “Limpar”, que gera a saída dos *stemming*.

## 5.2 Treinamento do sistema de classificação

Chegamos então à etapa que pretendeu classificar documentos de forma automática, utilizando o algoritmo SVM para detecção de padrões. O principal objetivo passou a ser o de comprovar que a remoção dos quantificadores e qualificadores do sintagma, em conjunto com o processo de *stemming*, pode permitir a identificação da semelhança temática entre documentos após uma etapa de treinamento da máquina.

Nesta etapa foi utilizado o software *SVMLight*<sup>14</sup> versão 6.02, desenvolvido pelo Departamento de Ciência da Computação da Universidade de Dortmund. O módulo de treinamento utilizado é denominado *svm\_multiclass\_classify.exe*, pois permite a identificação de mais de duas classes de treinamento.

Com a primeira parte do trabalho já realizada, neste momento temos vários arquivos, cada um equivalente a um documento do corpus e, em cada um dos arquivos, temos uma lista de todos os sintagmas nominais já normalizados e convertidos para seu *stemming*. Portanto, nossa massa de teste já está preparada.

### 5.2.1 Etapa 3a: Criação do arquivo de treinamento

O cálculo e a montagem desta tabela tornam-se necessários, uma vez que o software utilizado para treinamento, utilizando SVM, não trabalha diretamente com dados da forma textual. Ou seja, é necessária uma conversão de dado textual para dado numérico.

Para criação deste arquivo todos os documentos que compõem o corpus foram submetidos ao servidor classdoc.com, através de um upload. Após serem recebidos foi acionada a opção “*Gerar dados de treinamento*”, que gera uma sequência de dados para cada um dos arquivos como mostrado abaixo:

**01.txt:** 1:0 2:1 3:0 4:0 5:1 6:0 7:0 8:0 9:1 10:1 11:0 12:0 13:1 14:1 15:0 16:1 17:0 18:0 19:0  
20:0 21:1 22:0 23:1 24:0 25:1 26:1 27:1 28:0 29:0 30:0 31:0 32:1 33:0 34:1 35:0 36:0 37:1  
38:1 39:1 40:0 41:1 42:0 43:1 44:1 45:1 46:0 47:0 48:1 49:1 50:0 51:0 52:1 53:1 54:0 55:1  
56:0 57:0 58:0 59:0 60:0 61:0 62:1 63:1 64:0 65:1 66:0 67:0 68:1 69:0 70:0 71:0 72:0 73:1  
74:1 75:1 76:1 77:1 78:0 79:1 80:0 81:0 82:1 83:1 84:1 85:1 86:1 87:1 88:0 89:0 90:1 91:1

---

<sup>14</sup> O software pode ser acessado e baixado pelo link <<http://svmlight.joachims.org/>>.



92:0 93:1 94:1 95:0 96:0 97:1 98:0 99:0 100:1 101:1 102:0 103:0 104:1 105:1 106:0 107:0  
 108:0 109:1 110:0 111:0 112:1 113:1 114:1 115:1 116:0 117:1 118:1 119:0 120:0 121:0  
 122:0 123:0 124:0 125:0 126:1 127:1 128:1 129:1 130:1 131:0 132:1 133:0 134:0 135:0  
 136:1 137:1 138:0 139:1 140:1 141:1 142:1 143:1 144:0 145:0 146:1 147:1 148:1 149:1  
 150:0 151:1 152:1 153:0 154:1 155:0 156:1 157:1 158:0 159:0 160:1 161:0 162:0 163:0  
 164:1 165:1 166:0 167:0 168:1 169:0 170:1 171:0 172:1 173:0 174:1 175:0 176:0 177:0  
 178:0 179:0 180:0 181:1 182:0 183:0 184:0 185:0 186:0 187:0 188:0 189:0 190:0 191:0  
 192:0 193:0 194:1 195:0 196:1 197:0 198:0 199:0 200:0 201:0 202:0 203:0 204:0 205:0  
 206:0 207:0 208:1 209:0 210:0 211:0 212:0 213:1 214:1 215:0 216:0 217:1 218:1 219:1  
 220:1 221:0 222:0 223:1 224:0 225:1 226:0 227:0 228:0 229:0 230:0 231:0 232:1 233:1  
 234:0 235:0 236:0 237:0 238:1 239:1 240:1 241:1 242:1 243:1 244:1 245:0 246:1 247:0  
 248:1 249:0 250:0 251:0 252:0 253:1 254:1 255:0 256:0 257:0 258:1 259:0 260:0 261:1  
 262:1 263:0 264:0 265:0 266:0 267:0 268:0 269:0 270:1 271:0 272:0 273:0 274:1 275:0  
 276:1 277:1 278:1 279:1 280:0 281:1 282:1 283:0 284:1

**02.txt:** 1:1 2:1 3:0 4:1 5:1 6:0 7:1 8:1 9:1 10:0 11:1 12:1 13:0 14:0 15:0 16:0 17:0 18:1 19:1  
 20:0 21:0 22:0 23:1 24:0 25:1 26:1 27:1 28:0 29:0 30:0 31:1 32:1 33:0 34:0 35:0 36:0 37:0  
 38:0 39:1 40:0 41:1 42:1 43:0 44:0 45:1 46:1 47:0 48:0 49:1 50:1 51:1 52:0 53:0 54:0 55:1  
 56:0 57:1 58:1 59:1 60:0 61:0 62:1 63:1 64:0 65:0 66:0 67:0 68:1 69:1 70:0 71:0 72:0 73:0  
 74:1 75:0 76:1 77:0 78:0 79:0 80:0 81:0 82:1 83:0 84:0 85:1 86:0 87:0 88:1 89:1 90:0 91:0  
 92:0 93:0 94:0 95:0 96:0 97:0 98:0 99:0 100:0 101:0 102:0 103:1 104:1 105:0 106:0 107:1  
 108:0 109:1 110:0 111:0 112:0 113:0 114:0 115:1 116:1 117:0 118:0 119:0 120:0 121:1  
 122:0 123:0 124:1 125:1 126:1 127:0 128:0 129:0 130:0 131:1 132:0 133:0 134:1 135:1  
 136:1 137:0 138:0 139:1 140:0 141:0 142:0 143:0 144:1 145:1 146:0 147:0 148:1 149:1  
 150:1 151:0 152:0 153:0 154:0 155:1 156:1 157:1 158:1 159:1 160:1 161:1 162:0 163:1  
 164:1 165:0 166:1 167:0 168:0 169:0 170:0 171:0 172:0 173:1 174:0 175:1 176:0 177:1  
 178:1 179:1 180:0 181:0 182:0 183:1 184:0 185:1 186:0 187:0 188:0 189:0 190:0 191:0  
 192:1 193:0 194:0 195:1 196:1 197:0 198:1 199:0 200:0 201:0 202:0 203:1 204:0 205:0  
 206:1 207:0 208:1 209:1 210:1 211:0 212:1 213:0 214:0 215:0 216:0 217:1 218:1 219:0  
 220:0 221:0 222:0 223:1 224:1 225:0 226:0 227:0 228:1 229:0 230:0 231:1 232:1 233:1  
 234:0 235:0 236:1 237:0 238:1 239:0 240:1 241:1 242:0 243:0 244:0 245:0 246:0 247:1  
 248:1 249:1 250:1 251:1 252:0 253:0 254:0 255:1 256:0 257:0 258:0 259:0 260:0 261:1  
 262:1 263:0 264:0 265:1 266:1 267:0 268:1 269:1 270:0 271:0 272:0 273:0 274:1 275:1  
 276:1 277:1 278:0 279:1 280:0 281:0 282:1 283:0 284:0

O significado dessas listas está de acordo com o software de aprendizado – para cada arquivo de treinamento temos uma sequência como: **01.txt: 1:0 2:1 3:0.**

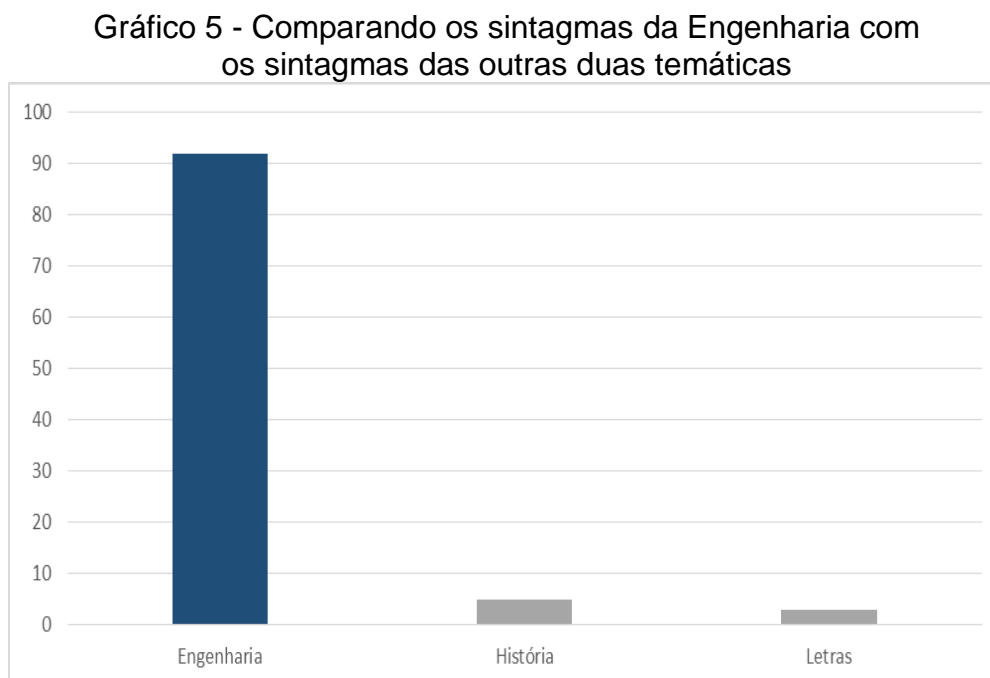
Neste caso, o documento é o **01.txt**. O sintagma número 1 não existe no documento (**1:0**), o segundo sintagma existe para esse documento (**2:1**) e o terceiro sintagma não existe (**3:1**).

Já para o documento **02.txt**, temos **1:1 2:1 3:0**, que significa: o SN 1 existe (**1:1**); o SN dois também existe nesse documento (**2:1**); e o SN 3 não existe (**3:0**). A lógica utilizada é a de encontrar, para cada sintagma de um documento, um sintagma igual em qualquer outro documento do conjunto de treinamento. Caso exista, ele irá então compor a lista de termos que aparecem duas ou mais vezes.

Na segunda etapa, para cada sintagma dessa lista, é verificado em qual documento ele existe (ou não), gerando a tabela como a apresentada acima.

O que obtemos no final é um arquivo com todos os documentos que serão utilizados no treinamento, já no formato adequado ao software *SVMLight*. Neste ponto os dados resultantes do processamento foram enviados para um gráfico de dispersão, gerando o seguinte resultado:

- a) Para os sintagmas dos textos sobre Engenharia tivemos o seguinte gráfico de dispersão:



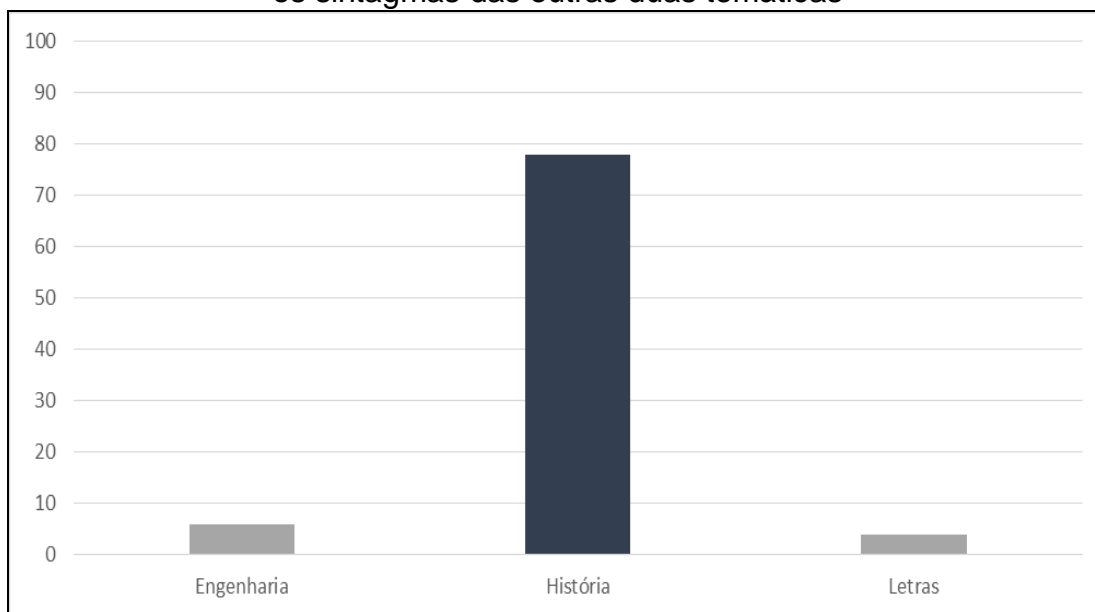
Fonte: O próprio autor

No caso da Engenharia ficou nítida a concentração dos termos dos 19 textos utilizados em uma área do gráfico onde tipicamente concentram-se os sintagmas

relativos à área de Engenharia. Apenas cerca de 8% dos termos foram identificados como sendo também de outras áreas.

- b) Para os sintagmas relativos aos documentos de História, tivemos a seguinte distribuição:

Gráfico 6 - Comparando os sintagmas da História com os sintagmas das outras duas temáticas

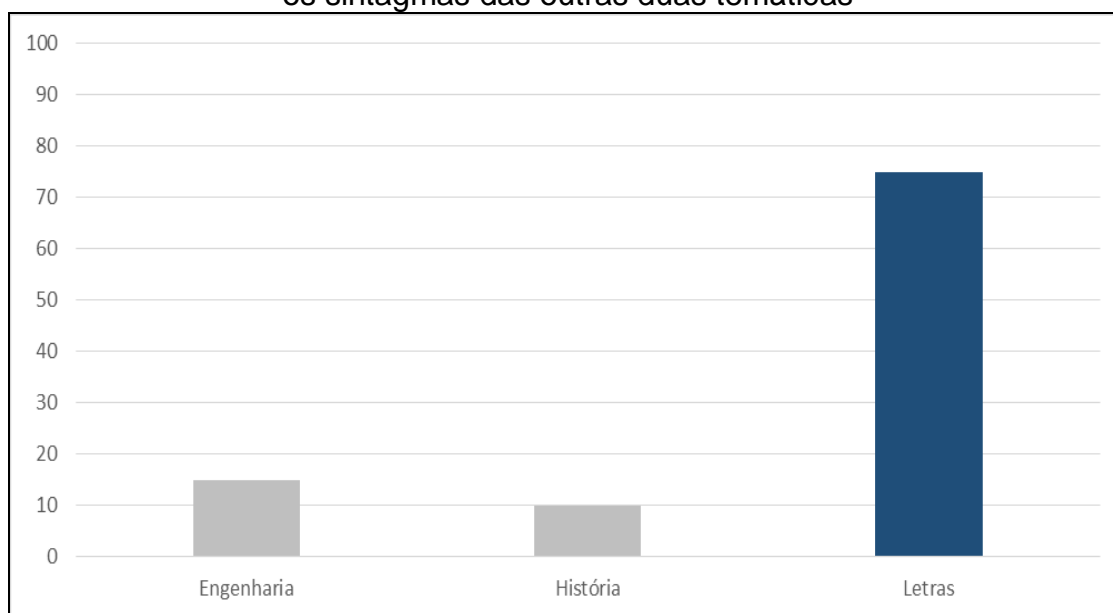


Fonte: O próprio autor

Também podemos perceber que a distribuição ficou maior na parte do gráfico destinada aos sintagmas próprios de documentos de História. Cerca de 10% dos sintagmas se distribuíram pelas outras duas áreas.

- c) Por último, tivemos a seguinte distribuição para os documentos da área de Letras:

Gráfico 7 - Comparando os sintagmas da Letras com os sintagmas das outras duas temáticas



Fonte: O próprio autor

Foram utilizados apenas 14 documentos para treinamento da área de Letras. Cerca de 25% dos documentos tiveram seus sintagmas dispersos pelas áreas de Engenharia e de História. Em três situações foram encontrados artigos de Letras que tratavam sobre História da Educação, o que pode ter confundido o sistema no momento do espalhamento do sintagma. Entretanto, podemos mesmo assim observar que a maior parte dos sintagmas (cerca de 75%) ficou concentrada na área do gráfico destinado ao tema de Letras.

Apesar de os gráficos apresentados evidenciarem o potencial agrupador de documentos quando se utilizam os sintagmas stemmizados, uma etapa de treinamento de máquina de classificação automática foi necessária para comprovar o funcionamento da metodologia. Na próxima seção será apresentada a forma utilizada para treinamento do sistema.

### 5.2.2 Etapa 3b: Treinamento e testes de classificação automática

Após as tabelas prontas, teve início o processo de treinamento do sistema. Ao todo, neste primeiro teste de classificação automática, foram utilizados para validação do treinamento 44 documentos para treinamento do sistema, divididos

entre as três temáticas escolhidas e já apresentadas, e 14 documentos, também das três temáticas.

Esta etapa foi toda realizada localmente sem o uso do servidor classdoc e os seguintes comandos foram utilizados:

<i>svm_multiclass_classify.exe</i>	<i>Software de avaliação do aprendizado</i>
<i>svm_multiclass_learn.exe</i>	<i>Software de treinamento</i>
<i>model</i>	<i>Modelo após o aprendizado</i>
<i>prediction</i>	<i>Modelo de predição</i>
<i>dadosteste.dat</i>	<i>Dados de teste do sistema</i>
<i>dadostreinamento.dat</i>	<i>Dados de treinamento</i>
<i>classificar.bat</i>	<i>Grupo de comandos para classificar</i>
<i>treinar.bat</i>	<i>Grupo de comandos para treinar</i>

Esta seção apresenta os resultados da submissão dos primeiros corpora de treinamento, composto por 44 artigos retirados do site da Scielo, como já apresentado anteriormente. Todos foram convertidos em texto, tratados de forma automática e, ao final, foram gerados os dados para treinamento.

O número de sintagmas em cada um dos documentos pode ser visto abaixo:

Tabela 24 – Número de sintagmas por documento de treinamento

<b>Tema</b>	<b>Documento</b>	<b>Sintagmas únicos</b>	<b>Total após <i>stemming</i></b>
Engenharia	1	977	901
	2	1340	1257
	3	1131	976
	4	947	871
	5	1047	873
	6	857	774
	7	1368	1199
	8	1065	940
	9	1158	1044
	10	1026	934
	11	2714	2355
	12	991	885
	13	800	694
	14	878	823
	15	1206	1079
	16	1096	958
	17	1291	1100
	18	1952	1397

Tabela 25 - Número de sintagmas por documento de treinamento

<b>Tema</b>	<b>Documento</b>	<b>Sintagmas únicos</b>	<b>Total após <i>stemming</i></b>
História	1	1838	1707
	2	1953	1713
	3	1756	1599
	4	2166	1569
	5	1763	1244
	6	2085	1542
	7	1407	1220
	8	2432	2184
	9	2295	2063
	10	2282	2050
	11	2368	2137
	12	2019	1851
	13	2302	2080

Fonte: O próprio autor

Tabela 26 - Número de sintagmas por documento de treinamento

<b>Tema</b>	<b>Documento</b>	<b>Sintagmas únicos</b>	<b>Total após <i>stemming</i></b>
Letras	1	1775	1580
	2	1282	1145
	3	1643	1388
	4	2107	1874
	5	1977	1754
	6	1779	1479
	7	2269	1974
	8	1571	1423
	9	1653	1425
	10	1245	1123
	11	1610	1413
	12	1400	1272

Fonte: O próprio autor

A tabela 27 traz o resultado da saída:

Tabela 27 - Número de sintagmas por documento de teste

<b>Tema</b>	<b>Documento</b>	<b>Sintagmas únicos</b>	<b>Total após <i>stemming</i></b>
Engenharia	1	1034	939
	2	1106	958

	3	1114	996
	4	1619	1414
	História		
		2321	2154
		1779	1649
		2377	2113
		1724	1572
		1673	1543
	Letras		
	1	1718	1537
	2	1616	1483
	3	1269	1140
	4	1265	1135
	5	1533	1387

Fonte: O próprio autor

Para realização do treinamento foi utilizado o seguinte comando:

```
svm_multiclass_classify dadosteste.dat model predictions type predictions
```

Figura 20 – Saída do processamento pelo software SVMLight

```

D:\TreinamentoSUM>treinar

D:\TreinamentoSUM>svm_multiclass_learn -c 1.0 dadostreinamento.dat model
Reading training examples... <42 examples> done
Training set properties: 3494 features, 3 classes
Iter 1: .....*(NumConst=1, SU=1, CEps=100.0000, QPEps=0.0000)
Iter 2: *(NumConst=2, SU=1, CEps=23.0771, QPEps=0.0000)
Iter 3: .....*(NumConst=3, SU=2, CEps=44.9847, QPEps=0.0000)
Iter 4: .....*(NumConst=4, SU=3, CEps=1.8226, QPEps=0.0000)
Iter 5: *(NumConst=5, SU=4, CEps=0.8688, QPEps=0.0000)
Iter 6: *(NumConst=6, SU=5, CEps=0.6171, QPEps=0.0000)
Iter 7: *(NumConst=7, SU=6, CEps=0.3447, QPEps=0.0000)
Iter 8: *(NumConst=8, SU=7, CEps=0.1831, QPEps=0.0000)
Iter 9: .....*(NumConst=9, SU=8, CEps=0.1011, QPEps=0.0000)
Iter 10: .....*(NumConst=9, SU=8, CEps=0.0733, QPEps=0.0000)
Final epsilon on KKT-Conditions: 0.07327
Upper bound on duality gap: 0.07327
Dual objective value: dval=83.52580
Primal objective value: pval=83.59908
Total number of constraints in final working set: 9 (of 9)
Number of iterations: 10
Number of calls to 'find_most_violated_constraint': 210
Number of SU: 8
Norm of weight vector: ||w||=5.74007
Value of slack variable (on working set): xi=67.05162
Value of slack variable (global): xi=67.12488
Norm of longest difference vector: ||Psi(x,y)-Psi(x,ybar)||=7.67162
Runtime in cpu-seconds: 0.06
Final number of constraints in cache: 126
Compacting linear model...done
Writing learned model...done

D:\TreinamentoSUM>_

```

Fonte: O próprio autor

Para avaliação dos resultados foi utilizado o comando:

```
svm_multiclass_classify dadosteste.dat model predictions type predictions
```

Isso gerou uma saída semelhante à apresentada na figura 21:

Figura 21 – Resultado final do processamento pelo software SVMLight

```

Prompt de Comando
D:\TreinamentoSUM>svm_multiclass_classify dadosteste.dat model predictions
Reading model...done.
Reading test examples... <13 examples> done.
Classifying test examples...done
Runtime <without IO> in cpu-seconds: -0.00
Average loss on test set: 0.0000
Zero/one-error on test set: 0.00% <13 correct, 0 incorrect, 13 total>

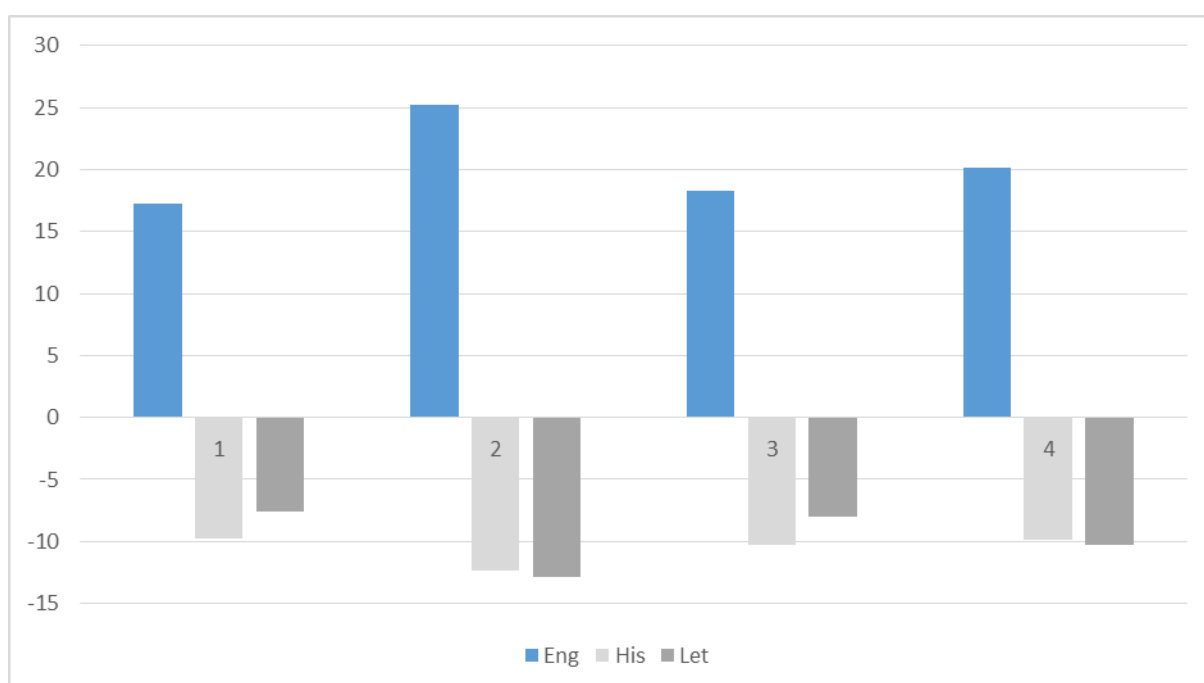
D:\TreinamentoSUM>type predictions
1 17.303192 -9.722189 -7.581003
1 25.230540 -12.325626 -12.904914
1 18.267675 -10.292333 -7.975343
1 20.153014 -9.840210 -10.312804
2 -9.464241 13.500689 -4.036448
2 -4.195493 8.021804 -3.826311
2 -6.430938 15.162568 -8.731630
2 -1.457812 4.071637 -2.613824
2 -5.387325 6.847135 -1.459810
3 -6.319567 -4.002161 10.321728
3 -4.153507 -1.805184 5.958691
3 -6.218073 -1.186119 7.404191
3 -6.005319 -3.287418 9.292737

```

Fonte: O próprio autor

Gráficos podem ilustrar melhor os resultados encontrados. No Gráfico 8 são apresentados, para cada um dos três temas, os documentos que foram submetidos ao teste de classificação automática e os comentários individuais em cada temática, iniciando pela Engenharia no Gráfico 8:

Gráfico 8 - Avaliação da classificação na área de Engenharia

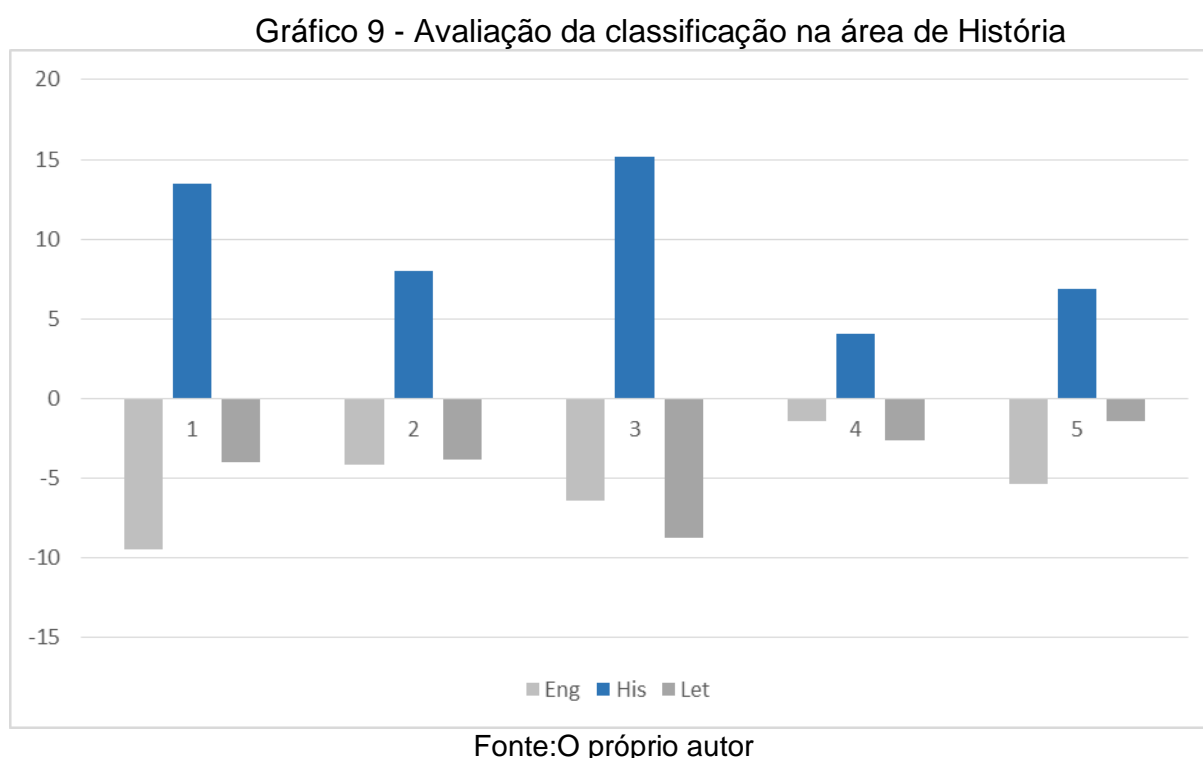




No total, foram quatro documentos para avaliação da temática Engenharia. Nenhum documento participou da etapa de treinamento, ou seja, são documentos também retirados do site da Scielo sobre a temática Engenharia, mas sem vínculo durante o treinamento.

Em azul escuro, temos a classificação correta para o tema de Engenharia. O documento melhor classificado foi o segundo documento, atingindo 25,23054 pontos, positivamente semelhante aos documentos de Engenharia utilizados durante o treinamento e -9,840210 e -10,312804, no que diz respeito à sua semelhança, com os documentos de treinamento ligados a História e a Letras.

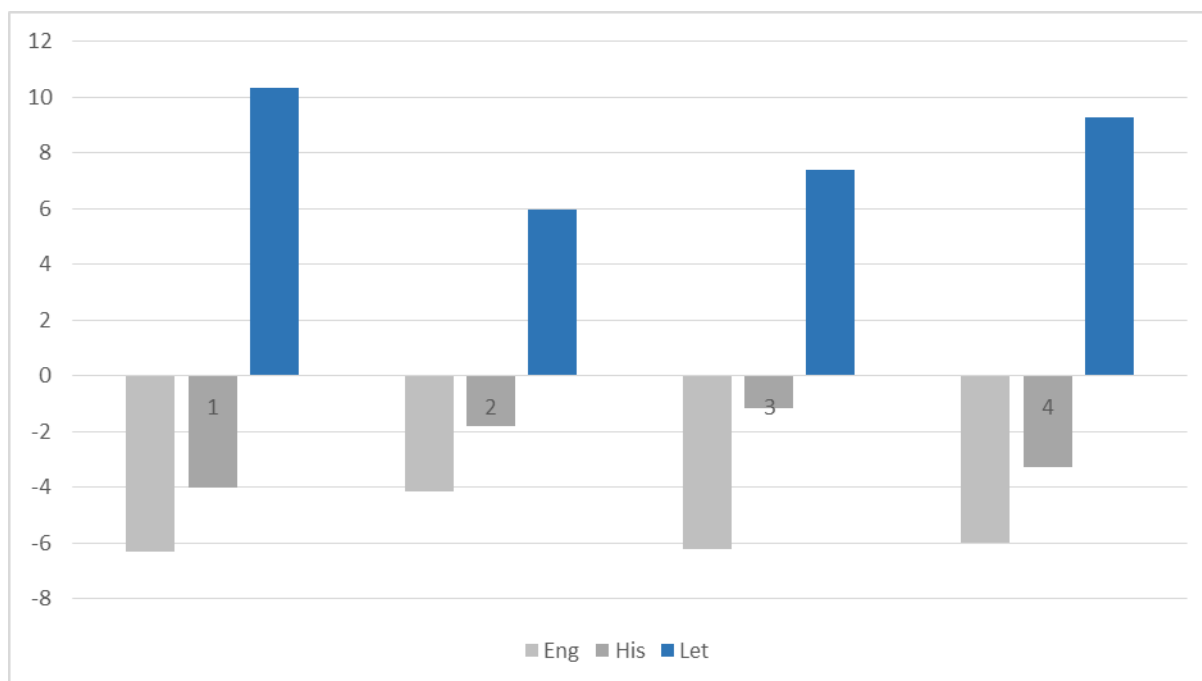
Para a temática História tivemos os seguintes resultados:



Nesse caso podemos perceber que o primeiro, o segundo e o terceiro documentos de testes foram muito bem classificados como sendo de História, os outros dois também foram classificados corretamente. No entanto, apesar de classificado corretamente, boa parte dos sintagmas do quarto documento também estavam presentes nas temáticas de Engenharia e de Letras.

E para o terceiro grupo de documentos tivemos os seguintes resultados:

Gráfico 10 - Avaliação da classificação na área de Letras



Fonte: O próprio autor

Para a área de Letras, também todos os quatro documentos de teste foram classificados corretamente, com um razoável grau de precisão, tendo apenas um dos documentos (número 2) ficado com um percentual de semelhança abaixo de 10%.

A seguir, temos as considerações acerca dos resultados encontrados na classificação automática, após o treinamento e o teste do sistema.

## 6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Qualquer sintagma sempre servirá apenas como uma pista para o significado propriamente dito, não garantindo a compreensão, também nunca nos dando a certeza de determinado tema em detrimento de outro. Isso pode ficar mais claro em trabalhos com temáticas interdisciplinares, como “História da Engenharia” e “Educação Organizacional”, entre outros temas que são recorrentes em vários artigos.

Como o o sentido e o significado de uma comunicação dependerá plenamente do ouvinte, nem toda comunicação é eficaz, estando sempre sujeita a interferências e falhas, como em qualquer outra atividade humana. O que está em jogo é o nosso conhecimento sobre o mundo. Portanto, qualquer tentativa de automatização da classificação de documentos será sempre limitada e fadada a equívocos naturais do ato comunicacional.

Dito isso, entramos então em nossa última etapa desta jornada, que é a de expor nossas considerações a respeito do que foi observado durante nossos trabalhos, ao longo destes quatro anos e, em seguida, apresentar algumas perspectivas de trabalhos futuros.

O principal objetivo desta tese era validar os sintagmas nominais como fontes de dados de treinamento para um sistema de classificação automática de documentos. Esses sintagmas foram escolhidos através de extração automática, utilizando ferramenta informatizada específica para este trabalho. Essa extração automática criou uma lista de SN que se tornariam a base para o treinamento da máquina de vetor de suporte utilizada para validar nossa hipótese.

Para validar nosso trabalho, foram utilizadas duas etapas. Na primeira tivemos uma visão qualitativa do processo, na qual ficou evidenciado que, com a observação dos dados mais de perto ao utilizar o *stemming* de um sintagma, podemos tornar mais semelhantes documentos com a mesma temática, estruturalmente falando. E, quando o mesmo processo é aplicado a documentos de temáticas diferentes, essa semelhança praticamente não é alterada. Em um segundo momento, agora fazendo uma avaliação quantitativa, foram utilizados 44 documentos de três temáticas diferentes para treinar a máquina, formando grupos que seriam úteis para a descoberta automática do tema de documentos que não participaram da etapa de treinamento.

Os pressupostos da pesquisa foram confirmados de forma extremamente satisfatória, mesmo ressaltando-se que nem todas as possibilidades de tratamento do sintagma foram usadas, como proposto durante a tese. Os resultados mostraram que o uso do *stemming* do sintagma, no lugar do próprio sintagma, proporciona vantagens em relação ao uso somente do sintagma no momento de treinar a máquina. O uso de sintagmas puros foi objetivo de outros trabalhos que apresentaram, em média, 80% de classificação satisfatória, enquanto no uso do *stemming* do sintagma esse percentual foi de 100% de classificações corretas. Entretanto, testes exaustivos ou com outros conjuntos de dados são deixados como sugestão para trabalhos futuros.

Na verdade, a inexistência, até a década passada, de ferramentas que permitissem a extração automática de SN foi um fator preponderante para o sucesso desta tese, observando-se, claro, que esse processo de extração ainda não atinge níveis de alta qualidade na lista final dos sintagmas, fato que decorre de dois fatores principais: 1) a própria dinâmica da língua e o não consenso sobre o sintagma em si; 2) o limite dos algoritmos em extrair uma informação, por vezes subjetiva, sobre o que é ou não um sintagma.

Podemos esperar que os *Parsers* sejam continuamente melhorados e que novas pesquisas surjam para atender à demanda de extração do SN para o português, softwares esses que raramente estão disponíveis para uso.

Há que se destacar a introdução do SN na Ciência da Informação pelas mãos de Kuramoto (1999, 2003), pois seu sistema de RI baseado em sintagmas já apontava para a direção das pesquisas que vieram posteriormente e que utilizavam o SN como corpus para sistemas automáticos de tratamento da informação. A pesquisa realizada por Kuramoto apresentou um modelo de RI baseado no sintagma, contando com a participação do usuário na escolha de documentos a partir de uma lista de sintagmas disponíveis.

Embora essa pesquisa tenha sido realizada a partir de vários outros trabalhos que envolvem a classificação de documentos, consideramos nosso trabalho de extrema relevância, pois o tratamento do sintagma antes de ser utilizado na implementação do sistema ainda não havia sido levado em consideração. Observamos que o *stemming* do sintagma deu resultados positivos durante o treinamento, conforme pode ser comprovado na metodologia proposta.

Vale ressaltar todo o trabalho já realizado anteriormente em PLN, recurso este amplamente utilizado nesta tese, amplamente pesquisado na Linguística Computacional, na Ciência da Computação e em estudos interdisciplinares que envolvem outras áreas, como a Neurociência.

Em relação ao processo de extração do sintagma realizado em um computador, baseado no processador Pentium i3 rodando o software *Palavras*, há que se dizer que os tempos não são atualmente razoáveis para uma computação nesse tipo de equipamento. Foram necessárias aproximadamente 3 horas de trabalho para o processamento de 44 textos do conjunto de treinamento e mais 1 hora, aproximadamente, para os demais 14 textos do corpus de teste.

Como conclusão, no que diz respeito ao processamento, é necessária uma computação de alto desempenho, baseada em sistemas em *clusters* ou em sistemas distribuídos, para que esse tempo possa cair para níveis razoáveis de aceitação a um trabalho de classificação automática. Quanto ao treinamento da máquina de classificação automática, o tempo gasto no processamento dos 44 textos não foi significativo, ficando abaixo de 1s, velocidade também observada quando o sistema SVM foi requisitado para classificar os 14 documentos de teste.

Apesar de ter sido observado que 100% dos documentos submetidos ao software SVM foram classificados corretamente, não devemos perder de vista determinadas áreas, dentre elas a Ciência da Informação, nas quais textos com viés interdisciplinar podem confundir a SVM, levando-a a classificar de forma incorreta as temáticas apresentadas. Portanto, consideramos que trabalhos futuros deveriam agregar novas etapas ao processamento dos sintagmas, além da sua conversão para *stemming*.

Podemos enumerar melhorias metodológicas que poderiam ser aplicadas no futuro:

1. Tratamento de adjetivos – Identificação de adjetivos correlatos em documentos diferentes, com a seguinte troca e padronização, utilizando recursos, tais como thesaurus;
2. Frequência no uso de determinadas palavras - Poderiam ser utilizados mecanismos, como o *Google trends*, que verifica a frequência diária do uso de determinadas palavras em detrimento de outras, o que poderia ser utilizado para troca e comparação entre palavras de mesmo sentido, mas que caíram em desuso ao longo do tempo;

3. Uso de thesaurus e ontologias de domínio – A consulta a thesaurus e a ontologias poderia resolver problemas de significado, tornando padronizadas determinadas sentenças estruturalmente diferentes, mas de mesmo potencial de significação;
4. Resolução anafórica – Este é um grande problema para a PLN: o uso de estruturas anafóricas pode resultar em sintagmas identificados de forma incorreta como também, em alguns casos, inviabilizar a identificação automática dos sintagmas;
5. Semelhança fonética – Foi observado que, em alguns documentos, o autor escreve determinadas palavras de forma errada, o que pode inviabilizar uma comparação direta. Apesar de o *stemming* resolver vários destes casos, muitos outros ainda causaram problemas de treinamento. O uso da fonética da palavra no lugar na palavra em si pode resultar em melhorias no sistema de classificação;
6. Hiperônimos – Em vários documentos foi percebido que a resolução dos hiperônimos poderia ajudar a tornar o sintagma mais semelhante, tendo em vista que estruturas como “equipamento de solda” e “máquina de solda” podem ter o mesmo sentido, apesar de suas estruturas serem completamente diferentes;
7. Tratamento de siglas – Percebeu-se também que em determinados documentos o autor utiliza muitas siglas, o que foi comum, por exemplo, nos documentos de Engenharia. Sintagmas como “SBC” e “Sistemas Baseados em Casos” podem não ser estruturalmente semelhantes, mas carregam potencialmente a mesma informação. Nos casos acima, a SVM os trataria como totalmente diferentes;
8. Outros elementos linguísticos – Diversos outros elementos linguísticos poderiam ser tratados antes de utilizar o sintagma com o objetivo de melhorar o desempenho final do sistema, tais como a meronímia, a holonímia, os hipônimos e a toponímia, como forma de identificar sintagmas como “Milho verde” como sendo uma cidade e não apenas um vegetal leguminoso.

O fator primordial na escolha de outras metodologias de tratamento do sintagma pode ser determinado pelo uso que o sistema final irá fazer do sintagma sendo processado. Alguns destes usos são:

1. A partir do treinamento com sintagma a SVM poderia ser utilizada para detecção do idioma do texto;
2. Detecção do estilo literário do autor;
3. Classificação automática de SPAN;
4. Detecção de documentos fraudulentos;
5. Mineração de textos e identificação de padrões em textos;
6. Análise de qualidade literária;
7. Construção automática de ontologias.

Ao nos aprofundarmos no estudo do sintagma, talvez estejamos trilhando um caminho ainda pouco explorado nas pesquisas, mas que aparentemente pode tornar o texto semanticamente mais rico. Talvez devêssemos deslocar nosso olhar com um pouco mais de atenção para o corpus, deixando os algoritmos de treinamento, como os SVM, para os especialistas da área de Inteligência Artificial.

Existe um longo caminho a ser percorrido e a demanda por sistemas cada vez mais eficientes e eficazes no tratamento e recuperação da informação é assunto imediato, mesmo que a mente, o sentido e o significado ainda não estejam ao nosso alcance.

## REFERÊNCIAS

- ALVARENGA, Lídia. **A Teoria do Conceito Revisitada em Conexão com Ontologias e Metadados no Contexto das Bibliotecas Tradicionais e Digitais**. Datagrama Zero, dez. 2001.
- AMARAL, S. F. do. Internet: novos valores e novos comportamentos. In: SILVA, E. T. (Coord.). **A leitura nos oceanos da internet**. São Paulo: Cortez, 2003.
- ARAÚJO, V. M. R. H. de. **Sistemas de recuperação da informação**: nova abordagem teórico-conceitual. 240 f. Tese (Programa de Pós-graduação em Educação) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 1994.
- ARAÚJO JÚNIOR, R. H. de. **Precisão no processo de busca e recuperação da informação**. Tese (Programa de Pós-graduação em Ciência da Informação) – Universidade de Brasília, Brasília, DF, 2005.
- ARCOVERDE, J. M. A. **Indução de filtros linguisticamente motivados na recuperação de informação**. 107 f. Dissertação (Programa de Pós-graduação em Ciências de Computação e Matemática Computacional) – Universidade de São Paulo, São Carlos, SP, 2007.
- ARISTÓTELES. **Organon**: V Tópicos. Lisboa: Guimarães Editores, 1987.
- AUSTIN, J. L. **How to do things with words**. Oxford: Clarendon Press, 1962.
- BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern information retrieval**. New York: ACM Press, 1999.
- BATES, M. **Models of natural language understanding human-machine communication by voice**. Irvine: National Academy of Sciences, 1993.
- BIBER, D. Spoken and written textual dimensions in English: Resolving the contradictory findings. **Language**, v. 62, n. 2, p. 384-413. 1986.
- BICK, E. Automatic Parsing of Portuguese. In: II Encontro para o Processamento Computacional de Português Escrito e Falado, CEFET-PR, Curitiba, p. 91-100, 1996. **Anais...** Curitiba: CEFET-PR, 1996.
- BORKO, H. Information Science: What is it? **American Documentation**, v. 19, n. 1, p. 3-5, Jan. 1968.
- BORTONI-RICARDO, S. M. A comunidade de fala brasileira. In: \_\_\_\_\_. **Educação em língua materna**: a sociolinguística na sala de aula. São Paulo: Parábola Editorial, 2004.
- BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDERMIR, T. B. **Redes neurais artificiais**: teoria e aplicações. 2. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2007. 260 p.



BROOKE, J.; HIRST, G. A Multi-Dimensional Bayesian Approach to Lexical Style. In: **Proceedings of NAACL-HLT**. 2013. p. 673-679.

CALADO, P. et al. Link-based similarity measures for the classification of Web documents. **Journal of the American Society for Information Science and Technology**, v. 57, n. 2, p. 208-221. 2006.

CARVALHO, Bernardo Penna Resende de (2005), **Novas Estratégias para Detecção Automática de Vetores de Suporte em Least Square Support Vector Machines**. Trabalho de pós-graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais. 2005.

CHAFE, W. Givenness, Constastiveness, Definiteness, Subjects, Topics and Point of View. IN: LI, C. N. (Ed.). **Subject and Topic**. New York: Academic Press, 1976.

COSTA, F. N. Q. M. C. **Deep Linguistic Processing of Portuguese Noun Phrases**. 214 f. Dissertação (Programa de Pós-graduação em Informática) – Faculdade de Ciências da Universidade de Lisboa, Lisboa, 2007.

COUTINHO, Gabriel. **Reconhecimento facial baseado em Support Vector Machines**. Dissertação. UFLA, 2007.

COVINGTON, M. A.; NUTE, D.; VELLINO, A. **Prolog Programming in Depth**. New York: Prentice-Hall, 1997.

CROSSLEY, S. A.; LOUWERSE, M. Multi-dimensional register classification using bigrams. **International Journal of Corpus Linguistics**, v. 12, n. 4, p. 453-478. 2007. Disponível em: <gsu.edu/~wwwesl/Files/ALSL/cross\_classification\_using\_bigrams.pdf>. Acesso em: 2014.

DAVID, K. A. **Sintaxe das expressões nominais no português do Brasil: uma abordagem computacional**. 118 f. Dissertação (Programa de Pós-graduação em Linguística) – Universidade Federal do Ceará, Fortaleza, CE, 2007.

DIAS, E. W. Análise de assunto: percepção do usuário quanto ao conteúdo de documentos. **Perspectivas em ciência da informação**, Belo Horizonte, v. 9, n. 2, p. 146-157, jul./dez. 2004.

GEY, F. Models in Information Retrieval. **Folders of Tutorial Presented at the 19th ACM Conference on Research and Development in Information Retrieval (SIGIR)**, 1992.

GLENDAY, C. **Chomsky e a linguística cartesiana**. Disponível em: <<http://www2.marilia.unesp.br/revistas/index.php/transformacao/article/viewFile/1026/925>>. Acesso em: mar. 2014.

GONZALEZ, M.; LIMA, V. L. S. Sintagma Nominal em Estrutura Hierárquica Temática na Recuperação de Informação. **Relatório PPGCC–PUCRS**, Porto Alegre, 2006. Disponível em: <<http://www.inf.pucrs.br/~gonzalez/docs/sneht.pdf>>. Acesso em: fev. 2014

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.

GROSSI, M. G. R. **Estudo das características de software e implementação de um software livre para o sistema de gerenciamento de Bibliotecas Universitárias Federais Brasileiras**. Tese (Programa de Pós-Graduação em Ciência da Informação) – Universidade Federal de Minas Gerais, Belo Horizonte, MG, 2008.

HALL, S. **Da diáspora: Identidade e Mediações Culturais**. Belo Horizonte: Editora UFMG, 2003.

HAYKIN, Simon. **Neural Networks: A comprehensive Foundation**. Prentice Hall. 842p. 1999.

JACOB, E. K.; SHAW, D. Sociocognitive perspectives on representation. **Annual Review of Information Science and Technology**, v. 33, p. 131-185, 1998.

JOHNSTONE, B. **Qualitative Methods in Sociolinguistics**. New York: Oxford University Press, 2000.

JURAFSKY, D.; MARTIN, J. H. **An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 2000.

KAULA, P. N. **Repensando os conceitos no estudo da classificação**. 1986. Disponível em: <<http://www.conexaorio.com/bit/kaula/index.htm>>. Acesso em: 13 jan. 2011.

KESSLER, B. et. al. **Automatic Detection of Text Genre**. 1997. Disponível em: <<http://acl.ldc.upenn.edu/P/P97/P97-1005.pdf>>. Acesso em: 26 jun. 2014.

KOCH, I. V.; SILVA, M. C. P. S. **Linguística aplicada ao português: sintaxe**. São Paulo: Cortez, 1985.

KOPPEL, M. et. al. **Automatically Categorizing Written Texts by Author Gender**. Disponível em: <[clips.uantwerpen.be/~walter/educational/material/Koppel\\_LLC2003.pdf](http://clips.uantwerpen.be/~walter/educational/material/Koppel_LLC2003.pdf)>. Acesso em: 2014.

KUNH, T. **A estrutura das revoluções científicas**. São Paulo: Perspectiva, 1994.

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

LACERDA, W. S.; BRAGA, A. P. Experimento de um Classificador de Padrões Baseado na Regra Naive de Bayes. **INFOCOMP Journal of Computer Science**, Lavras, v. 3, n. 1, p. 30-35, 2004.

LANCASTER, F. W. **Indexação e resumos: teoria e prática**. 2. ed. Brasília: Briquet de Lemos, 2004.

LIBERATO, Y. G. **A estrutura do SN em português**. Tese (Programa de Pós-graduação em Letras) – Universidade Federal de Minas Gerais, Belo Horizonte, MG, 1997.

LOPES, L. **Extração automática de conceitos a partir de textos em língua portuguesa**. 156 f. Tese (Programa de Pós-Graduação em Ciência da Computação) – Pontifícia Universidade Católica Do Rio Grande Do Sul, Porto Alegre, RS, 2011.

LOPES, M. C. S. **Mineração de Dados Textuais Utilizando Técnicas de Clustering para o Idioma Português**. 180 f. Tese (Programa de Pós-Graduação em Engenharia Civil) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2004.

LORENA, A. C.; CARVALHO, A. C. P. L. F. de. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43-67, 2007.

LUZIO, E. R. C.; RODRIUGES, M. L. Marcas da Oralidade em textos escritos. **Web revista páginas de debates**, 2011.

MAIA, L. C. G.; SOUZA, R. R. Uso de sintagmas nominais na classificação automática de documentos eletrônicos. **Perspectivas em Ciência Informação**, Belo Horizonte, v. 15, p. 154-172, 2010.

MANNING, C. D. RAGHAVAN, P.; SCHÜTZE, H. **Introduction to Information Retrieval**. Cambridge: Cambridge University Press, 2008.

MARTINS, A. L. Potenciais aplicações da Inteligência Artificial na Ciência da Informação. **PBCIB**, v. 5, n. 2, 2012.

MIORELLI, S. T. Extração do sintagma nominal em sentenças em português. 98 f. Dissertação (Programa de Pós-Graduação em Ciência da Computação) – Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, RS, 2001.

NEHMY, R. M. Q. et al. A ciência da informação como disciplina científica. **Perspectivas em Ciência da Informação**, Belo Horizonte, v. 1, n. 1, p. 9-25, jan./jun. 1996.

OBERMEIER, K. K. Natural Language Processing. **Byte**, v. 12, n. 14, p. 225-232, 1987.

OLIVEIRA, C.; FREITAS, M. C. de. Um modelo de sintagma nominal lexical na recuperação de informações. In: XI Simpósio Nacional e I Simpósio Internacional de Letras e Linguística (XI SILEL), Uberlândia, p. 778-786, 2006. **Anais...** Uberlândia, 1996.

OTHERO, G. A. **Grammar Play**: um parser sintático em Prolog para a língua portuguesa. 265 f. Dissertação (Programa de Pós-Graduação em Letras) – Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, RS, 2004a.

\_\_\_\_\_. Sobre a evolução linguística. **Revista Eletrônica de Divulgação Científica em Língua Portuguesa, Linguística e Literatura**, n. 01, ano 01, 2004b.

\_\_\_\_\_. **A gramática da frase em português**: algumas reflexões para a formalização da estrutura frasal em português. Porto Alegre: EDIPUCRS, 2009.

PATERIYA, L. P. K. A Study on Author Identification through Stylometry. **International Journal of Computer Science & Communication Networks**, v. 2, n. 6, p. 653-657, 2003.

PAVELEC, D. F. et. al. Identificação da Autoria de Documentos Digitais com Base em Atributos Estilométricos da Língua Portuguesa. In: **Proceedings of the International Joint Conference IBERAMIA/SBIA/SBRN 2006 - 4th Workshop in Information and Human Language Technology (TIL'2006)**, Ribeirão Preto, Brazil, October 23–28, 2006.

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

\_\_\_\_\_. **Gramática Descritiva do Português**. 4. ed. São Paulo: Editora Ática, 2003.

PIMENTEL, E. P.; FRANÇA, V. F. de.; OMAR, N. A identificação de grupos de aprendizes no ensino presencial utilizando técnicas de clusterização. In: **Anais do Simpósio Brasileiro de Informática na Educação**, Rio de Janeiro, p. 495-504, 2003.

PINHEIRO, L.V.R. **A Ciência da informação entre sombra e luz: domínio epistemológico e campo interdisciplinar**. Tese (Programa de Pós-Graduação em Comunicação e Cultura) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 1997.

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 em: <<http://www.educ.fc.ul.pt/docentes/opombo/investigacao/opombo/classificacao.pdf>>. Acesso em: 13 jan. 2011.

PRENSKY, M. Digital natives, digital immigrants. **On the horizon**, v. 9, n. 5, 2001.

SALTON, G.; MCGILL, M. J. **Introduction to modern information retrieval**. New York: McGraw-Hill, 1983.

SANTOS, C. N. dos. **Aprendizado de máquina na identificação de sintagmas nominais**: o caso do português brasileiro. Tese (Programa de Pós-Graduação em

Sistemas e Comunicação) – Instituto Militar de Engenharia, Rio de Janeiro, RJ, 2005.

SANTOS, I. de G. A influência do suporte digital na produção escrita de aprendizes de língua inglesa: um estudo sobre Netspeak. **Domínios de Linguagem**, v. 6, n. 2, p. 191-206, 2012.

SARACEVIC, T. Information Science: origin, evolution and relations. In: VAKKARI, P.; CRONIN, B. (Ed). Conceptions of library and information science. **Proceeding of the international conference for the celebration of 20th anniversary of the Department of Information Studies**, University of Tampere, Finland, 26-28, 1991. London; Los Angeles: Taylor Graham, 1992.

\_\_\_\_\_. Ciência da informação: origem, evolução e relações. **Perspectivas Ci. Inf.**, Belo Horizonte, v. 1, n. 1, p. 41-62, jan./jun. 1996.

SOUZA, R. R. **Uma proposta de metodologia para a escolha automática de descritores utilizando sintagmas nominais**. 215f. Tese (Programa de Pós-Graduação em Ciência da Informação) – Universidade Federal de Minas Gerais, Belo Horizonte, MG, 2005.

\_\_\_\_\_. Sistemas de recuperação de informações e mecanismos de busca na web: panorama atual e tendências. **Perspectivas em Ciência Informação**, Belo Horizonte, v. 11, n. 2, aug. 2006.

SUNG, A. H.; MUKKAMALA, S. **Identifying important features for intrusion detection using support vector machines and neural networks**. In: Applications and the Internet, 2003. **Proceedings. 2003 Symposium on**, p. 209-216, jan. 2003.

TRISTÃO, A. M. D. et al. Sistema de classificação facetada e tesauros: instrumentos para organização do conhecimento. **Ci. Inf.**, Brasília, v. 33, n. 2, p. 161-171, maio/ago. 2004.

VAPNIK, V. N. **The nature of statistical learning theory**. New York: Springer-Verlag, 1995. ISBN 0387945598.

VIEIRA, R.; LIMA, V. L. S. Linguística computacional: princípios e aplicações. In: **Anais do XXI Congresso da SBC - I Jornada de Atualização em Inteligência Artificial**, p. 47-86, 2001.

VOUTILAINEN, A. NPtool: a detector of English noun phrases. **Proceedings of Workshop on Very Large Corpora** held on June 22, 1993 at Ohio State University.

ZINS, C. Redefining information science: from “information science” to “knowledge science”. **Journal of Documentation**, v. 62, n. 1, p. 447-461, 2006.

## ANEXOS

### ANEXO A – TEXTOS COMPLETOS

#### **Documento 1: Aplicação da Inteligência Artificial em Sistemas de Gerenciamento de Conteúdo**

F. C. Santos C. L. de Carvalho

The contents of this document are the sole responsibility of the authors. O conteúdo do presente documento é de única responsabilidade dos autores.

Instituto de Informática Universidade Federal de Goiás  
www.inf.ufg.br

Aplicação da Inteligência Artificial em Sistemas de Gerenciamento de Conteúdo  
Fernando Chagas Santos \*  
fernandosam@gmail.com

Cedric Luiz de Carvalho  
cedric@inf.ufg.br

**Abstract.** There are many frameworks to help in knowledge management. Some frameworks support the knowledge life cycle, others support knowledge production. Recently, the perspective has been carried over to knowledge management. Artificial intelligence techniques can be used to assist this management. This work shows how these techniques can be applied in content management systems to improve the management of knowledge in these systems. **Keywords:** Content Management Systems, Artificial Intelligence, Artificial Intelligence Techniques, Knowledge Management **Resumo.** Existem vários arcabouços para apoiar o gerenciamento do conhecimento. Alguns apoiam o ciclo de vida do conhecimento, outros a sua produção. Atualmente, a perspectiva está sobre o gerenciamento do conhecimento. Técnicas de inteligência artificial podem auxiliar nesse gerenciamento. Este trabalho mostra como estas técnicas podem ser aplicadas nos sistemas de gerenciamento de conteúdo para melhorar o gerenciamento do conhecimento destes sistemas. **Palavras-Chave:** Sistemas de Gerenciamento de Conteúdo, Inteligência Artificial, Técnicas de Inteligência Artificial, Gerenciamento do Conhecimento

#### 1 Introdução

A turbulência vivenciada pelas organizações contemporâneas pode ser identificada como a transição para aquilo que Drucker [6] identificou como “a sociedade global do conhecimento”. A globalização tem feito com que as organizações se coloquem em uma nova posição competitiva, onde o conhecimento e o comportamento dos seus colaboradores têm possibilitado vantagens competitivas. Dentro deste contexto, as organizações tentam melhorar suas posições no cenário competitivo através do uso do conhecimento, procurando meios de reter a experiência e os

recursos intelectuais que elas possuem, enquanto procuram aplicar os novos conhecimentos adquiridos [7]. Além do conhecimento, o sucesso de uma organização depende de muitos fatores [19]:

A habilidade dos empregados e departamentos executarem os serviços da organização no prazo definido; • O trabalho em equipe, através da coordenação, cooperação e colaboração; • O grau de inovação, como ele é capturado, comunicado e aplicado; • A efetivação de sistemas, procedimentos e políticas organizacionais. Por outro lado, considerando que as organizações estão se tornando complexas e algumas vezes, multinacionais, a tomada de decisão está se tornando muito complicada, difícil e arriscada [10]. De modo geral, existe uma aceitação de que a economia baseada em conhecimento tem crescido e que as organizações de sucesso, na sociedade global do conhecimento, são aquelas que conseguirão identificar, valorar, criar e envolver os seus conhecimentos como ativos. A base da economia do conhecimento é formada pela tecnologia da informação e comunicação (TIC). É ela que permite armazenar, processar, e fazer circular, rapidamente e a baixo custo, um número de dados cada vez maior, sendo uma fonte cada vez mais importante de ganhos de produtividade. A complexidade das organizações levanta a questão de como adquirir, armazenar, acessar e reutilizar o conhecimento. Diante dessa necessidade, o Gerenciamento do Conhecimento (GC) surgiu como uma disciplina de gerenciamento na segunda metade da década de 90. Antes de conceitualizar o que é GC, os conceitos de dado, informação, conhecimento e inteligência precisam ser discernidos. Os dados são considerados como fatos brutos, enquanto a informação se refere a como é organizado um conjunto de dados [3]. A Figura 1 mostra os estágios da evolução da aprendizagem.

**Estágios de Evolução da Aprendizagem [10]** O conhecimento pode ser interpretado como a informação baseada em experiências, habilidades e competências de cada pessoa, já a inteligência é adquirida através das transformações das experiências e da aquisição de novos conhecimentos. Para a psicologia, a inteligência é a capacidade mental para calcular, raciocinar, perceber relacionamento e analogias, aprender, armazenar e recuperar informações, usando linguagem fluente, classificando, generalizando e ajustando às novas situações. De acordo com Alfred Binet [9], a inteligência é a totalidade do processo mental envolvido na adaptação com o ambiente. O GC é um arcabouço conceitual que circunda todas as atividades das organizações e que é requerido para que as atitudes inteligentes dessas organizações estejam sobre uma base sustentável [19]. O GC também pode ser visto como uma disciplina que incorpora os processos organizacionais e busca uma combinação sinérgica de dados, capacidade de processamento de informação das tecnologias de informação, criatividade e habilidade de inovação das pessoas [1]. A área de GC foca na exploração e no desenvolvimento de conhecimentos (explícito, documentado, tático e subjetivo) para as organizações alcançarem os seus objetivos. Atualmente, três campos de pesquisa em GC podem ser identificados [17]:

- A teoria do conhecimento, o conhecimento da cultura organizacional, a mensuração do capital intelectual e a aprendizagem organizacional.

A retenção corporativa (também conhecida como retenção organizacional ou retenção de sistemas de informação organizacionais) para melhorar a tomada de decisões. • Os agentes inteligentes, as ontologias e a colaboração mediada por computador. Vários autores analisam o papel da “Organização do Conhecimento” no processo de criação do conhecimento, enfatizando que as companhias de sucesso são aquelas que criam novos conhecimentos, dissemina-os através da organização e que rapidamente incorpora-os como novas tecnologias e produtos [7]. Este processo promove a inovação e o desenvolvimento de vantagens competitivas [12]. O GC se relaciona com diferentes áreas, tais com: recursos humanos, marketing e inteligência artificial. Além disso, o GC abrange técnicas e processos para a criação, coleta, indexação, organização, distribuição, acesso e avaliação do conhecimento organizacional [17]. De um modo geral, os principais assuntos do GC são a organização, a distribuição e o refinamento do conhecimento [10]. O conhecimento pode ser gerado por ferramentas de mineração de dados, pode ser adquirido de terceiros ou através de sistemas gerenciadores de conteúdo ou ser refinado e atualizado a partir de uma base de conhecimento. O conhecimento coletado pode ser organizado através de relacionamentos entre os elementos de conhecimento, integrado em uma base de conhecimento e distribuído para ser utilizado por aplicações de suporte à tomada de decisões. Como resultado, as aplicações de suporte à tomada de decisões são usadas para refinar o conhecimento existente e exibir o conhecimento solicitado. A Figura 2, mostra como o processamento do conhecimento ocorre:

Figura 2: Estágios do Processamento do Conhecimento [10] A apresentação do conhecimento, ou seja, como o conhecimento é mostrado aos membros da organização é um tópico importante. Em geral, uma organização possui diferentes procedimentos para formatar sua base de conhecimento. Devido aos diferentes estilos de apresentação, os membros da organização podem ter dificuldades para reconfigurar, recombina e integrar o conhecimento de fontes separadas ou distintas. Entretanto, para que os processos e as técnicas do GC tenham êxito, aspectos culturais e humanos devem ser considerados, bem como o desenvolvimento de sistemas inteligentes para melhorar a performance e a execução das tarefas que envolvam o conhecimento. É possível distinguir três fases no gerenciamento do conhecimento. Na primeira fase, as corporações possuem um repositório de informações centralizado. A segunda fase é marcada

Aplicação da Inteligência Artificial em Sistemas de Gerenciamento de Conteúdo por comunidades que compartilham conhecimento. A terceira fase é referente ao uso de wikis, blogs, websites e outros recursos para expandir o gerenciamento do conhecimento. Devido ao avanço das tecnologias baseadas na Web e ao desenvolvimento baseado em componentes, surgiram diversas abordagens para o desenvolvimento de técnicas de inteligência artificial para serem usadas no contexto do GC. Como exemplo [17]: os perfis de usuários, a personalização das interações homem-computador e o gerenciamento de conteúdo. Os sistemas gerenciadores de conteúdo, além de permitirem a aquisição de conhecimento também podem ser utilizados para a distribuição do conhecimento. A Seção 2, apresenta esses sistemas e a Seção 3 demonstra as técnicas de inteligência artificial e como elas podem ser utilizadas nos sistemas de gerenciamento de conteúdo.



Os primeiros websites foram desenvolvidos na primeira metade da década 90 por universitários que desejavam divulgar informações para seus colegas. Devido aos recursos limitados da linguagem HTML, utilizada para o desenvolvimento dos websites, apenas a informação essencial era apresentada [11]. Na segunda metade da década de 90, a proliferação da quantidade de websites, tornou ineficiente a gestão deles nas organizações. Neste período, surge o papel do webmaster, que tinha a responsabilidade de gerir o conteúdo dos websites das organizações. A ineficiência na gestão dos websites ocorreu devido à grande quantidade de conteúdo que o webmaster deveria incluir no portal da organização, às limitações técnicas das linguagens utilizadas, tal como a linguagem HTML, que não permitia separar a informação da sua formatação e a grande rotatividade tecnológica. As organizações adotaram medidas para tornar eficiente a gestão de seus websites. A mais importante dessas medidas foi a descentralização da gestão dos websites [11]. As funções, que eram de responsabilidade apenas do webmaster, foram divididas entre várias pessoas nas organizações. Os Sistemas de Gerenciamento de Conteúdo (SGC) surgem neste período. Os SGC são sistemas que têm sido desenvolvidos para integrar os sistemas de gerenciamento de documento com os sistemas de recuperação de informação. Desde então, tem havido uma convergência entre estas duas plataformas. O conhecimento do processo organizacional permite projetar um cenário onde os SGC possibilitam a criação, o armazenamento, a manipulação e a apresentação de informações sobre a organização em um ambiente virtual. A crescente complexidade de muitos portais organizacionais, desenvolvidos com o uso de SGC, bem como a complexidade do processo de publicação digital, tornaram o desenvolvimento dos SGC complexo. Sob o ponto de vista humano, os SGC devem ser compreendidos como um ambiente de trabalho colaborativo e distribuído, fornecendo suporte para a realização de tarefas que são desempenhadas pelas pessoas. A gestão do conteúdo deve ser definida sob o ponto de vista das atividades das pessoas e dos seus objetivos. Para isso, um conjunto de processos deve ser estruturado para a produção de publicações digitais. [2] Os requisitos mínimos para os SGC são: fornecer aplicações para a criação, edição e armazenamento de conteúdo, permitir controle de fluxo, possuir um repositório de informações, fornecer ferramentas para a integração de informações externas e fornecer modelos (templates). Uma arquitetura com estas características pode ser visualizada na Figura 3. Para auxiliar a coleta de informações, os SGC devem ser flexíveis e simples de utilizar. A manutenção e a atualização do conteúdo do website geralmente é responsabilidade de pessoas

Arquitetura de um Sistema Gerenciador de Conteúdo com competências técnicas [11]. Entretanto, o SGC deve permitir à qualquer membro de uma organização, incluir ou alterar informações sem dificuldades técnicas. Para isso, o SGC deve disponibilizar interfaces intuitivas, que devem ser acessadas a partir de um navegador Web. A exigência de programas específicos para a publicação de conteúdo impede a portabilidade do SGC [11]. Dessa forma, o SGC deve possibilitar ao produtor de conteúdo publicar suas informações em qualquer lugar e momento. A gestão de conteúdo correta possibilita que qualquer colaborador da organização, detentor de informação produza o seu conteúdo no website da organização. Além disso, reduz erros de publicação e facilita o processo de validação. Entretanto, é importante destacar que o sucesso ou o fracasso de um SGC em uma organização

não está relacionado à tecnologia e sim as pessoas e ao processo adotado. Para a definição de um SGC, o processo mais simples é utilizar um arcabouço integrado por um conjunto de módulos com objetivos específicos. Entretanto este processo se torna complexo quando a quantidade de informações são muitas e a natureza destas informações são diferentes. Os fornecedores dos SGC precisam adaptar as funcionalidades necessárias para cada organização. De um modo geral, não existe uma solução universal que atenda todos os requisitos das organizações, desta forma, é necessário escolher entre a aderência da funcionalidade a um módulo fornecido pelo SGC ou ao desenvolvimento da funcionalidade solicitada. Selecionar, implementar e implantar um SGC resulta em um estudo e análise detalhada da organização que utilizará esse sistema, dos objetivos da organização, dos processos de trabalho, dos recursos de informação utilizados e dos usuários que o sistema afetará. A complexidade do gerenciamento da informação levou ao desenvolvimento de duas áreas específicas: O Gerenciamento de Conteúdo Web (GCW) e o Gerenciamento de Conteúdo de Negócios (GCN). Os GCW utilizam o ambiente da Internet e seus objetivos e métodos estão focados na produção de documentos digitais e informação para a Internet. Os GCN, por outro lado, são baseados na idéia do gerenciamento total das informações dentro das organizações, através da integração de todas as informações necessárias para a organização alcançar os seus objetivos. As aplicações típicas de um SGC são os websites editoriais, as comunidades de prática e os portais corporativos [11]. Diversos SGC de código livre podem ser encontrados na Internet ([15] e [16]). Os websites editoriais permitem que um grupo de indivíduos publiquem informações sobre assuntos específicos. Os portais de informação verticais, tais como jornais, revistas e blogs, são os websites mais comuns desta natureza. As comunidades de prática são utilizadas por comunidades que compartilham interesses pessoais ou profissionais. Estes websites possibilitam às pessoas contribuírem com artigos e notícias. As listas de discussão, chats, wikis e fóruns, permitem aos membros da comunidade compartilhar conhecimentos. Os portais corporativos permitem centralizar o conhecimento das organizações em um único local [11]. As idéias, documentos e procedimentos administrativos devem ser coletados de maneira estruturada para garantir a segurança ao acesso às informações. Os problemas típicos encontrados nas organizações que os SGC auxiliam na solução são [11]:

- Gargalos que dificultam a produção de conteúdos para a Web.
- Falta de comprometimento dos usuários, devido à dificuldades técnicas de publicação e uso.
- Falta de organização do conteúdo.
- Riscos de erros e informações de baixa qualidade.
- Interfaces misturadas ao conteúdo de maneira rígida.

Além da solução para os problemas típicos das organizações, os websites editoriais, as comunidades de prática e os portais corporativos podem ser transformados com o uso de sistemas de gerenciamento de conhecimento, permitindo agregar valor às informações incluídas nessas aplicações. O armazenamento de conhecimento requer que o módulo de manipulação de conteúdo crie e altere conhecimento ao invés de dados. Outra abordagem é obter os dados e transformá-los em conhecimento a partir de relações com outros dados e elementos do sistema. Os principais SGC disponíveis não possibilitam a manipulação do conhecimento. Desta forma, a segunda abordagem, que busca obter os dados e transformá-los em conhecimento se torna mais viável. Para isto, a inteligência artificial fornece tecnologias que permitem esta transformação.

Para a aplicação da IA nos sistemas gerenciadores de conteúdo, existem duas questões que são comumente encontradas pelos pesquisadores da IA dentro da área de gerenciamento de conhecimento [17]. A primeira questão é “Depois de décadas de pesquisa na área de engenharia de conhecimento, o que é exatamente engenharia do conhecimento?” É possível compreender a engenharia do conhecimento e o gerenciamento do conhecimento como áreas de conhecimento distintas [17]. A engenharia de conhecimento tem, como consenso geral, um foco mais técnico sobre o conhecimento (por exemplo: representação, organização, raciocínio e procura). Já o gerenciamento do conhecimento, está mais alinhado aos objetivos de capturar, compartilhar e reutilizar o conhecimento em uma organização ou entre organizações. Os projetos de gerenciamento do conhecimento podem continuar sem qualquer esforço de engenharia de conhecimento aplicado em um sistema gerenciador de conteúdo. Entretanto, todo projeto de gerenciamento de conhecimento envolve alguma engenharia de conhecimento para o fornecimento de serviços com valor agregado [17].

A segunda questão sobre a aplicação da IA nos sistemas de gerenciamento de conteúdo é: ainda não existem sistemas de IA que podem conversar com um humano. Apesar de tudo, a IA deveria atender os problemas mais difíceis da gestão de conteúdo. As ferramentas de gerenciamento de conhecimento mais sofisticadas já aceitam alguma forma de tecnologia de IA, como por exemplo, raciocínio bayesiano, ontologias, mineração de dados e agentes inteligentes [17]. IA e Gestão de Conhecimento (GC) são práticas integrativas [1]. Estas soluções estão sendo utilizados por muitas companhias que começam a compreender como aplicar as práticas de GC para agregar valor aos seus serviços. As tecnologias usadas no suporte às iniciativas de GC estão sendo rapidamente incorporadas aos softwares de GC. Um interesse particular no uso de arcabouços, é o papel da IA nesses arcabouços. A IA tem recebido atenção durante as duas últimas décadas e tem sido amplamente aplicada em muitas áreas de negócio. As principais categorias analisadas são: Sistemas Especialistas (SE), Redes Neurais Artificiais (RNA) e Agentes Inteligentes (AI) [10].

### Sistemas Especialistas

Os sistemas especialistas surgiram como uma área da Inteligência Artificial (IA) durante a década de 70, a partir do esforço de pesquisadores para desenvolver programas computacionais que pudessem raciocinar como humanos. Um sistema especialista é um programa de computador que tem uma base de conhecimento sobre um domínio e utiliza o raciocínio para executar tarefas que especialistas humanos poderiam executar [18]. Em outras palavras, um sistema especialista é um sistema computacional que possui um corpo de conhecimento bem organizado que tem o objetivo de solucionar problemas do mundo real que envolvam habilidades de especialistas em um domínio específico. Este tipo de sistema é capaz de apresentar conclusões sobre determinado assunto neste domínio. Para que o sistema especialista atinja o seu objetivo, ele deve interagir com o usuário assim como um especialista humano faria, por exemplo, ouvindo o usuário, evitando perguntas cuja resposta pode ser deduzida, mudando a forma de apresentação de acordo com o usuário e tirando conclusões, mesmo que os dados fornecidos não sejam totalmente

completos. Entre as características para a interação do sistema especialista com o usuário destaca-se as seguintes [13]:

- Explicar seu raciocínio. Para convencer o usuário de que a solução apresentada é adequada ao problema, é necessário que o sistema descreva de forma clara e precisa o raciocínio utilizado que o levou àqueles resultados.
- Adquirir conhecimento novo e modificar o conhecimento antigo. Um especialista humano está sempre atento a novas informações que o levem a modificar seu conhecimento ou mesmo complementá-lo. Da mesma forma, um sistema especialista deve manter sempre atualizadas suas bases de conhecimento.
- Manter interações contínuas entre o especialista humano e o sistema especialista. Uma outra maneira é submeter os mesmos dados brutos utilizados pelo especialista humano e permitir que o sistema especialista aprenda com ele. A base de conhecimento é o coração de um sistema especialista e fornece o conhecimento necessário para solucionar problemas específicos. O conhecimento pode estar na forma de fatos, heurísticas (por exemplo: experiências, opiniões, julgamentos, previsões, algoritmos) e

Aplicação da Inteligência Artificial em Sistemas de Gerenciamento de Conteúdo é normalmente coletado de um especialista, através de métodos de aquisição de conhecimento (por exemplo: entrevistas, análise de protocolos, questionários) [10]. As técnicas de aquisição de conhecimento também podem ser aplicadas para capturar o conhecimento e desenvolver repositórios de conhecimento para os sistemas de gerenciamento de conteúdo. O método de representação do conhecimento é outro aspecto importante de um sistema especialista. A linguagem associada ao método escolhido deve ser suficientemente expressiva (por exemplo, lógica) para permitir a representação do conhecimento a respeito de um domínio de forma completa e eficiente. Regras de Produção é o método mais comum de representação do conhecimento. Sistemas de produção é um nome genérico para os sistemas baseados em regras de produção, ou seja, pares de expressões consistindo em uma condição e uma ação. As principais vantagens dos sistemas de produção como método de representação de conhecimento são: a modularidade, a uniformidade e a naturalidade. Como desvantagens considera-se: ineficiência em tempo de execução e complexidade do fluxo de controle para solucionar problemas. Outra característica comum nos sistemas especialistas é a existência de um mecanismo de raciocínio incerto que permita representar a incerteza a respeito do conhecimento do domínio. Devido à necessidade de expressar o conhecimento incerto, ocorreu o desenvolvimento de diversos métodos de representação do conhecimento:

- Lógica: base para a maioria dos formalismos de representação de conhecimento, seja de forma explícita, como nos sistemas especialistas baseados na linguagem Prolog, seja mascarada na forma de representações específicas que podem facilmente ser interpretadas como proposições ou predicados lógicos.
- Redes semânticas: consiste em um conjunto de nós conectados por um conjunto de arcos. Os nós, em geral, representam objetos e os arcos, relações binárias entre esses objetos. Mas, os nós podem também ser utilizados para representar predicados, classes, palavras de uma linguagem, entre outras possíveis interpretações, dependendo do sistema de redes semânticas adotado.
- Quadros ou frames: permitem a expressão das estruturas internas dos objetos, mantendo a possibilidade de representar herança de propriedades como nas redes semânticas.

Os Sistemas de Gerenciamento de Conteúdo podem obter vantagem dessas técnicas e aplicá-las no suporte à codificação do conhecimento. A Figura 4 apresenta uma arquitetura de um sistema especialista. Segundo a Figura 4, um

sistema especialista apresenta uma arquitetura com os subsistemas de aquisição e explanação do conhecimento e os módulos: base de conhecimentos, máquina de inferência, memória de trabalho e interface com o usuário. A base do conhecimento reúne o conhecimento do especialista modelado de acordo com o método de representação do conhecimento definido. A máquina de inferência examina o conteúdo da base de conhecimentos e define a ordem em que se fazem as inferências. Desta forma, de acordo com uma consulta do usuário, a máquina de inferência transfere fatos e regras para a memória de trabalho, que armazena os fatos e as regras mais recentes. O subsistema de aquisição de conhecimentos é responsável pela atualização da base de conhecimentos, através da interação com o especialista, o engenheiro de conhecimento e o módulo de explanação. O subsistema de explanação é responsável pela descrição do raciocínio do sistema para o usuário, ou seja, detalha o raciocínio utilizado pelo sistema para a obtenção do resultado (solução).

Figura 4: Arquitetura de um sistema especialista. Adaptado de [7] Entre os benefícios da utilização dos sistemas especialistas, pode-se destacar [8]: • Ajuda a reduzir falhas humanas e acelerar tarefas; • Aumenta o desempenho e a qualidade na resolução de problemas; • Apresenta estabilidade e flexibilidade; • Combina e preserva o conhecimento dos especialistas; • Contempla hipóteses múltiplas simultaneamente; • Integra várias ferramentas; • Apresenta maior eficiência e otimização de resultados; • Não é afetado por questões psicológicas, estresse e fatores externos; • Possui maior rapidez na resolução de problemas; • Pode solucionar problemas tão bem quanto um especialista humano.

### Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) foram desenvolvidas, originalmente, na década de 40, pelo neurofisiologista Warren McCulloch, do MIT, e pelo matemático Walter Pitts, da Universidade de Illinois. Eles foram os primeiros pesquisadores a tratar o cérebro como um “organismo computacional” [5]. As RNA consistem em um método para solucionar problemas de IA, a partir do desenvolvimento de sistemas que simulem o cérebro humano, inclusive seu comportamento, ou seja, aprendendo, errando e fazendo descobertas.

A abordagem da RNA é vista como completamente diferente dos sistemas especialistas (Seção 3.1), pois nessa abordagem não existe uma base de conhecimento explícita, e sim um conjunto de relações derivadas entre dados. Assim, não se pode afirmar que as redes neurais artificiais possuem conhecimento sobre um domínio específico [10]. As RNA são compostas por um grande número de neurônios artificiais que são interconectados em rede para solucionar problemas. A técnica para a solução de problemas é semelhante à utilizada pelos humanos. O neurônio artificial é uma estrutura lógico-matemática análoga a uma unidade de processamento que aceita e combina estímulos de vários outros neurônios artificiais e procura simular a forma, o comportamento e as funções do neurônio biológico. A Figura 5 mostra um modelo básico de um neurônio  $j$  com entrada  $x_k$ , pesos sinápticos  $w_j$ , nível de ativação  $J$  e saída  $f(J)$ .

Figura 5: Modelo básico de um neurônio Como exibido na Figura 5, cada neurônio  $j$  possui um vetor de dados de entrada  $x_k = [x_1, x_2, \dots, x_n]^T$ , uma ativação interna  $J$ ,

uma função de ativação  $f(J)$  e os pesos sinápticos  $w_j = [w_{j1}, w_{j2}, \dots, w_{jd}]^T$ , que conectam os elementos de  $x_k$  ao neurônio  $j$ . O aprendizado das redes neurais ocorre quando há modificações significantes nas sinapses entre neurônios. Uma sinapse é o nome dado à conexão existente entre neurônios. Nestas conexões são atribuídos valores, chamados de pesos sinápticos, que são usados para armazenar o conhecimento. Para determinar se uma modificação é significativa, verifica-se a ativação dos neurônios. Se determinadas conexões são mais usadas, então estas conexões são reforçadas enquanto que as demais são enfraquecidas. Há basicamente 4 tipos de aprendizado nas redes neurais artificiais: • Supervisionado: são sucessivamente apresentadas à rede, conjuntos de padrões de entrada e seus correspondentes padrões de saída. A rede ajusta os pesos das conexões entre os elementos de processamento ('neurônio'), até que o erro entre os padrões de saída gerados pela rede alcancem um valor mínimo definido previamente; • Reforço: ao invés de fornecer as saídas corretas para a rede relativas ao treinamento individual, a rede recebe um valor que diz se a saída está correta ou não; • Não-supervisionado: a rede analisa os conjuntos de dados de entrada, determina algumas propriedades do conjunto de dados e aprende a refletir estas propriedades na sua saída;

Híbrido: as camadas da rede neural podem utilizar o aprendizado supervisionado ou não-supervisionado. Quando uma rede neural artificial é implantada para uma determinada aplicação, é necessário um período para que esta rede seja treinada. Assim como nos sistemas biológicos, aprender envolve ajustes nas conexões que existem entre os neurônios. Em outras palavras, a informação "aprendida" é armazenada na forma de valores numéricos, chamados pesos, que são designados para conexões entre o processamento dos neurônios da rede [10]. Com uma rede neural estruturada, uma série de valores podem ser aplicados sobre um neurônio, que está conectado a outros neurônios pela rede. Estes valores são multiplicados no neurônio pelo valor do peso de sua sinapse. Então, esses valores são somados. Se esta soma ultrapassar um valor limite estabelecido, um sinal é propagado pela saída (axônio) deste neurônio. Em seguida, essa mesma etapa se realiza com os demais neurônios da rede. Ou seja, os neurônios vão enfrentar algum tipo de ativação, dependendo das entradas e dos pesos sinápticos. As RNA's podem ser categorizadas por sua topologia, isto é, pelo número de camadas, de elementos de processamento e de conexões; pelas características de seus elementos de processamento; e pelas leis de aprendizagem a que foram submetidas [5]. A Figura 6 apresenta uma arquitetura de uma rede neural com 3 camadas.

Figura 6: Rede Neural com 3 camadas A principal vantagem desta tecnologia é que ela pode operar com dados incompletos para gerar e demonstrar a intuição aparente [10]. Além disso, com as RNA's é possível trabalhar analogicamente com o cérebro, o que deve ser potencializado dentro de um arcabouço de gerenciamento de conhecimento. Também é possível criar perfis de usuários para permitir informações a serem encaminhadas para indivíduos específicos de acordo com suas preferências e interesses. Os sistemas de gerenciamento de conteúdo podem obter vantagens desta tecnologia na distribuição e no compartilhamento do conhecimento. Ao invés de simplesmente possuir um modo de distribuição passiva, onde o repositório de conhecimento da organização é fornecido para os indivíduos acessá-lo, um módulo específico pode ser alterado para análise e distribuição do conhecimento para os stakeholders [10]. Por outro lado, uma desvantagem da

tecnologia de RNA é o fato de que elas necessitam de entradas para serem apresentadas em diversas formas, assim elas podem estar sujeitas ao peso

Aplicação da Inteligência Artificial em Sistemas de Gerenciamento de Conteúdo dos algoritmos de aprendizagem [10]. Ao contrário dos sistemas especialistas, que também aceitam símbolos de entrada, isto não está de acordo com o domínio do gerenciamento do conhecimento, que assume um mundo, baseado em entidades e entendimento prático.

A Inteligência Artificial (IA) possui 3 abordagens distintas, a abordagem simbólica, a conexonista e a distribuída. A Inteligência Artificial Distribuída (IAD) origina-se na sociologia, utiliza modelos de inteligência baseados no comportamento social e busca solucionar problemas de maneira cooperativa em um certo ambiente através de agentes distribuídos. Para a compreensão de Agentes Inteligentes (AI), é necessário antes a compreensão do que é um agente. Um agente é qualquer coisa que possa perceber o ambiente e agir sobre ele [14]. Um agente inteligente é um sistema computacional situado em algum ambiente e que é capaz de executar ações autônomas neste ambiente para atingir os objetivos para os quais ele foi planejado [20]. O agente inteligente deve ser sensível ao ambiente, responder às mudanças ocorridas neste ambiente, possuir ações orientadas a metas, ter iniciativa e interagir com outros agentes para solucionar problemas ou auxiliar na solução de problemas de outros agentes. De modo geral, os agentes são dotados de uma grande quantidade de conhecimento, experiências profissionais e crenças que eles usam para realizar suas tarefas. O estudo de agentes inteligentes tem se tornado um dos mais importantes campos na inteligência artificial distribuída [10]. Os sistemas multi-agentes são sistemas que utilizam vários agentes para realizar suas tarefas. Estes sistemas podem ser reativos ou cognitivos. Os agentes reativos não armazenam as suas ações, não representam o ambiente, não representam o conhecimento explicitamente e agem de acordo com a situação instantânea. A Figura 7 mostra a arquitetura de um agente reativo simples.

**Figura 7: Agente Reativo Simples** Os agentes reativos simples possuem sensores e atuadores que observam o ambiente. Quando o agente observa algo relevante através dos sensores, ele verifica a aparência atual do mundo e executa ações neste ambiente através dos atuadores. As ações são orientadas de acordo com o Estímulo-Resposta (Ação-Reação).

Os agentes cognitivos possuem uma representação explícita de conhecimento sobre o ambiente e outros agentes que colaboram com ele e podem armazenar suas ações. Além dos agentes reativos simples e dos agentes cognitivos, os agentes podem ser baseados em modelos, baseados em objetivos, baseados na utilidade ou com aprendizagem. Maiores detalhes sobre esses agentes podem ser encontrados em [14]. Os agentes inteligentes diferem dos objetos (do paradigma da orientação a objetos) em diversos pontos. Agentes inteligentes manipulam objetos para executarem suas tarefas (um agente inteligente pode ser visto como um objeto com uma cabeça). O comportamento de um agente inteligente (as tarefas que eles executam e como as tarefas são executadas) pode ser modificado dinamicamente, devido ao aprendizado ou a influência de outros agentes. Agentes inteligentes podem ser autônomos, podem executar ações de forma independente e podem ser móveis. Eles podem efetuar buscas de maneira dinâmica para auxiliar na solução de

problemas [10]. A busca e recuperação de métodos de conhecimento nos sistemas gerenciadores de conteúdo podem ser auxiliados pelos agentes inteligentes. Além disso, eles podem ser usados para auxiliar na combinação de conhecimentos, para a criação de novos conhecimentos. A partir do relacionamento de conhecimentos, os agentes inteligentes podem criar múltiplas perspectivas da mesma situação. Essas perspectivas podem contribuir para aumentar a quantidade de possíveis soluções e melhorar a qualidade do processo de tomada de decisões [4]. Além disso, agentes inteligentes podem ser aplicados para analisar o conhecimento e disseminar determinadas partes da informação e do conhecimento (ex: sumários, recomendações) para aqueles que poderiam fazer o uso destas partes.

### Considerações Finais

As técnicas de inteligência artificial podem ser utilizadas nos sistemas de gerenciamento de conteúdo para melhorar o gerenciamento do conhecimento destes sistemas. Entretanto, não existe uma técnica ideal a ser adotada. O módulo de manipulação de conteúdo de um sistema de gerenciamento de conteúdo com características de comunidade de prática pode ser aprimorado com o uso de sistemas especialistas. O aprimoramento consiste em permitir a manipulação do conhecimento de especialistas devido à natureza das comunidades de prática. O módulo de template de um sistema de gerenciamento de conteúdo, pode ser adaptado para exibir interfaces de acordo com o perfil do usuários com o uso das redes neurais. Os agentes inteligentes possibilitam transformar os dados dos sistemas de gerenciamento de conteúdo em conhecimento. Além disso, os agentes inteligentes são bastante flexíveis e podem ser utilizados nos principais módulos dos sistemas de gerenciamento de conteúdo. Enfim, escolher a técnica a ser adotada depende do tipo de sistema de gerenciamento de conteúdo adotado e dos objetivos da organização. A principal vantagem das organizações na aplicação de técnicas de IA em seus atuais sistemas de gerenciamento de conteúdo consiste na agregação de valor ao seu capital intelectual. Para a nova economia, isso significa vantagem competitiva.

### **Documento 2: Inteligência Artificial Aplicada a Ambientes de Engenharia de Software: Uma Visão Geral**

RENATO A FONSO C OTA S ILVA<sup>1</sup> Departamento de Informática – Universidade Federal de Viçosa CEP 36570-000 Viçosa, MG 1 renatoacs@dpi.ufv.br  
 Resumo. A Inteligência Artificial sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana. Softwares são produtos intangíveis e utilizam no seu processo de construção recursos intelectuais humanos, que vão desde sua especificação até sua distribuição e pleno funcionamento. Como meio de auxiliar o processo de Engenharia de Software, foram criados os ambientes de Engenharia de Software centrados no processo, que possuem um conjunto de ferramentas integradas. Baseado neste contexto, este artigo vem mostrar alguns ambientes existentes que utilizam técnicas de Inteligência Artificial e propor o uso de outras técnicas para melhorar os Ambientes de Engenharia de Software, trazendo uma maior facilidade de construção de softwares e uma maior qualidade para os mesmos. Palavras-Chave: Inteligência Artificial, Ambientes de Engenharia de Software, Processo de desenvolvimento de Software



### Artificial Intelligence in Software Engineering Environments: A Roadmap

**Abstract.** The Artificial Intelligence organizes and automates intellectual task and, therefore, is potentially relevant for any sphere of the human intellectual activity. Softwares are immaterial products and in their construction process use human intellectual resources, that go since its specification to its distribution and full operation. As a mean of helping out the software engineering process, Process-centered Software Engineering Environment were created, which has a set of integrated tools. Presented that, this paper is going to show some existing environments that use techniques of Artificial Intelligence and proposes the use of other techniques in order to improve the Software Engineering Environments, developing a easier technique of software construction and improving its quality. **Keywords:** Artificial Intelligence, Software Engineering Environments, Software Process

(Received August 03, 2005 / Accepted November 18, 2005) 1 Introdução

A Engenharia de Software refere-se à aplicação disciplinada de princípios e métodos no projeto e construção de software de qualidade de forma economicamente viável [2]. Alguns métodos procuram apoiar o gerente de desenvolvimento de software na tarefa de observar

e controlar o desenvolvimento de software em termos de seus recursos, prazos, alocação de tarefas e orçamento. Entretanto, a natureza multidisciplinar da área demanda cooperação entre profissionais e ferramentas que apoiem o desenvolvimento de software. Informalmente, o processo de desenvolvimento de software pode ser compreendido como o conjunto de

todas as atividades necessárias para transformar os requisitos do usuário em software [13, 21]. O processo de desenvolvimento de software é formado por um conjunto de passos parcialmente ordenados, relacionados com conjuntos de artefatos, pessoas, recursos, estruturas organizacionais e restrições, tendo como objetivo produzir e manter os software requisitados [17, 7]. A partir de um esboço dos requisitos iniciais para o problema a ser resolvido, através de software, um modelo de processo de desenvolvimento de software será adotado, o que resultará no software para atender os requisitos dos usuários. Com a tecnologia de processo de desenvolvimento de software, surgiu nos últimos anos um novo conceito de ambiente de trabalho que dá suporte de forma integrada aos processos de gestão e produção: Ambientes de Engenharia de Software Centrados no Processo, os chamados PSEEs (do inglês, Process-Centered Software Engineering Environments) [16], que constituem um tipo especial de ambiente de desenvolvimento de software que apoia a definição rigorosa de processos de software, objetivando a automação da gerência do desenvolvimento. Tais ambientes geralmente provêem serviços para análise, simulação, execução e reutilização das definições de processos, que cooperam no aperfeiçoamento contínuo de processos. Dentro deste ambiente, existem os agentes, que estão relacionados com as atividades de um processo, podendo ser pessoas ou ferramentas automatizadas. Agentes diferentes terão percepções diferentes acerca do que acontece durante o processo de desenvolvimento de software. Um gerente, por exemplo, perceberá os aspectos de controle e alocação de recursos e cronogramas para atividades,

enquanto um desenvolvedor perceberá as suas atividades como atribuições que devem ser feitas para produzir um resultado [7]. Este artigo apresenta, na Seção 2, conceitos sobre ambientes de Engenharia de Software. Na Seção 3 são apresentadas algumas técnicas de Inteligência Artificial que podem ser aplicadas em conjunto com os PSEEs e em seguida, na Seção 4, são mostrados alguns PSEEs que utilizam alguma técnica de IA. Logo após, na Seção 5, é apresentada uma avaliação desses ambientes do ponto de vista do uso de técnicas de IA e algumas propostas são apresentadas, tendo em vista a restrição observada do uso dessas técnicas nos ambientes atuais.

## 2 Ambientes de Engenharia de Software

Environment) é bastante recente. Um SEE é definido como uma coleção de ferramentas que fornece apoio automático, parcial ou total, às atividades de Engenharia de Software. Normalmente essas atividades são executadas dentro de uma estrutura de projeto de software, e se referem a aspectos tais como especificação, desenvolvimento, reengenharia ou manutenção de software [25]. O termo SEE pode ser aplicado a vários sistemas de alcances bem diferentes: desde um conjunto de poucas ferramentas executando sobre um mesmo sistema, até um ambiente totalmente integrado capaz de gerenciar e controlar todos os dados, processos e atividades do ciclo de vida de um software. Graças a automatização total ou parcial das atividades, um SEE pode contribuir com importantes benefícios para uma organização: redução de custos, aumento da produtividade, melhora da gestão e maior qualidade do software final. Por exemplo, automatização de atividade repetitivas como execução de casos de teste que não apenas melhoram a produtividade, mas também ajuda a garantir o término e a consistência das atividades testadas [25]. Normalmente, um SEE gerencia informações relacionadas com:

- Desenvolvimento ou manutenção do software (especificações, dados de projeto, códigos fonte, dados de teste, planos de projeto, ...);
- Recursos do projeto (custos, recursos de informática, pessoal, responsabilidades e obrigações, ...);
- Aspectos organizacionais (política da organização, padrões e metodologias empregadas, ...).

Um SEE oferece suporte às atividades humanas mediante uma série de serviços que descrevem as capacidades do ambiente. Os serviços proporcionam uma correspondência entre um conjunto de processos selecionados, relativos ao ciclo de vida do software, e sua automatização mediante o uso de ferramentas. Na maioria dos casos as funcionalidades de uma ferramenta estão relacionadas com um ou mais serviços.

### 2.1 Ambientes de Engenharia de Software Centrados no Processo

Embora o uso de ferramentas para auxiliar os desenvolvedores na produção de softwares venha sendo aplicado há algum tempo, o conceito de ambiente de Engenharia de Software - SEE (do inglês, Software Engineering

Os ambientes de Engenharia de Software centrados no processo, os chamados PSEEs (do inglês, Process-centred Software Engineering Environment) [16] constituem um tipo especial de ambientes de Engenharia de Software que surgiu nos últimos anos para apoiar a definição rigorosa de processos de software, objetivando automatizar a gerência do desenvolvimento. Tais ambientes geralmente provêem serviços para análise, simulação,

execução e reutilização das definições de processos, que cooperam no aperfeiçoamento contínuo de processos. A modelagem de processos de software

não consiste apenas em escrever programas que automatizem completamente o processo de desenvolvimento de software e nem descrevem tudo o que os atores do processo devem fazer [23]. Enquanto os programas de computador são escritos para definir o comportamento de uma máquina determinística, os programas de processo são escritos para definir possíveis padrões de comportamento entre elementos não-determinísticos (atores) e ferramentas automatizadas [26]. Como consequência, um PSEE deve ainda permitir que os atores envolvidos no processo recebam orientação automatizada e assistência na realização de suas atividades, sem interferência no processo criativo [20]. Além disso, processos ainda podem ser modificados dinamicamente, em resposta a estímulos organizacionais ou mudança nos requisitos do software em desenvolvimento. A literatura especializada define três tipos principais de modelos [9, 6, 8]:

- Modelos Abstratos (patterns ou templates), que fornecem moldes de solução para um problema comum, em um nível de detalhe que idealmente não está associado a uma organização específica. Um processo abstrato é um modelo de alto nível que é projetado para regular a funcionalidade e interações entre os papéis de desenvolvedores, gerentes, usuários e ferramentas em um PSEE [27];
- Modelos Instanciados (ou executáveis) são modelos prontos para execução, podendo ser submetidos à execução por uma máquina de processo. O modelo instanciado é considerado uma instância de um modelo abstrato, com objetivos e restrições específicos, envolvendo agentes, prazos, orçamentos, recursos e um processo de desenvolvimento;
- Modelos em Execução ou Executados registram o passado histórico da execução de um processo, incluindo os eventos e modificações realizadas no modelo associado.

A arquitetura de um PSEE usualmente define como componente central a máquina de processo [6] que auxilia na coordenação das atividades realizadas por pessoas e por ferramentas automatizadas, sendo responsável pela interpretação/execução dos modelos de processos descritos com PMLs (Process Modelling Language). Uma máquina de processos é responsável por: ativar automaticamente atividades sem intervenção humana através de uma integração com as ferramentas do

ambiente; apoiar o envolvimento cooperativo dos desenvolvedores; monitorar o andamento do processo e registrar o histórico da sua execução [16]. A máquina de processo também deve garantir a execução das atividades na sequência definida no modelo de processo; a repetição de atividades; a informação de feedback sobre o andamento do processo; a gerência das informações de processo (incluindo gerência de versões); a coleta automática de métricas; a mudança do processo durante sua execução; a interação com as ferramentas do ambiente e a gerência de alocação de recursos [16]. O mecanismo de execução de processos de um SEE pode conter diversas instâncias simultâneas de máquinas de execução. Isto é necessário porque o SEE pode estar sendo utilizado para desenvolvimento em diversos projetos em uma organização. Portanto, de forma geral um mecanismo de execução consiste de uma ou várias máquinas de execução. As máquinas de execução possuem componentes que trabalham na execução de processos e na integração com o restante do ambiente pois a execução envolve desde a interface com o usuário até a gerência dos objetos no banco de dados. Dentro de um PSEE, o mecanismo de execução pode ser tratado como mais uma ferramenta ou pode ser um componente básico do ambiente.

### 3 Técnicas de Inteligência Artificial

Inteligência Artificial é uma das ciências mais recentes, que atualmente abrange uma variedade enorme de subcampos, que vão desde áreas de uso geral, como

aprendizado e percepção, até tarefas mais específicas, como jogos de xadrez [24]. A IA sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana, que neste caso, será abordada técnicas de IA para auxiliar o processo de desenvolvimento de softwares através dos PSEEs.

### 3.1 Sistemas Especialistas

Sistemas especialistas (SE's) são uma classe de software que atuam como colaboradores na tomada de decisão em áreas da ciência dominadas por especialistas humanos. Um sistema especialista condensa o conhecimento de um ou mais especialistas e utiliza este conhecimento armazenado para auxiliar na resolução de problemas do usuário. Os SE's são Sistemas Baseados em Conhecimento (SBC's) que atuam em áreas e em tarefas bem definidas. Estruturalmente, todo SE é constituído de duas partes principais: a Base de Conhecimento (BC), que contém o conhecimento heurístico e fatorial sobre o do-

mínio de aplicação do SE, e a Máquina de Inferência (MI), que usa o conhecimento da BC para construir a linha de raciocínio que leva à solução do problema. O conhecimento obtido dos especialistas pode ser representado através de formalismos distintos: lógica, regras de produção, redes semânticas, frames e raciocínio baseado em casos. Embora tenham abordagens diferentes ao problema da representação do conhecimento, nada impede que sejam utilizadas combinações entre duas abordagens, como por exemplo, regras e frames para a representação do conhecimento de um determinado sistema.

### 3.2 Sistemas Baseados em Regras

O paradigma de regras é um dos paradigmas mais populares para representação do conhecimento em sistemas de IA. Conforme [12] isso se deve, principalmente, à natureza modular das regras, as suas facilidades de explanação e por modelarem o conhecimento de uma forma muito próxima ao processo cognitivo humano. Uma regra consiste de uma parte SE, o lado esquerdo da regra, e de uma parte ENTÃO, o seu lado direito. A parte SE lista um conjunto de condições combinadas de forma lógica e a parte ENTÃO representa a ação a ser executada ou a conclusão a ser deduzida, caso todas as condições da parte SE tenham sido satisfeitas. Sintaticamente, as regras podem ser representadas como :

ou onde medidas de certeza estatísticas, probabilísticas ou ad-hoc sejam usadas para desenvolver parâmetros." Os sistemas fuzzy combinam a flexibilidade e representação de conhecimento de alto nível dos sistemas especialistas convencionais com a habilidade de tratar problemas complexos e não lineares com o mínimo de regras. Segundo [4], os sistemas baseados em lógica fuzzy tem como principal benefício a redução dos custos de desenvolvimento, execução e manutenção. Entretanto não são adequados para todos os tipos de sistemas em que os sistemas convencionais se aplicam [4]. Diferente da lógica booleana, que possui os estados verdadeiro ou falso, a lógica fuzzy trata de valores verdade que variam continuamente de 0 a 1. Dessa forma um fato pode ser meio verdade 0,5, quase verdade 0,9 ou quase falso 0,1. O uso da lógica fuzzy em sistemas de raciocínio traz impacto não somente na máquina de inferência, mas também na representação do conhecimento. A lógica fuzzy permite expressar conhecimento em um formato de regra que é bastante parecido com linguagem natural.

### 3.4 Raciocínio Baseado em Casos

ENTÃO

SE

ENTÃO

A coleção de predicados é chamada de Memória de Trabalho (MT). A parte ENTÃO da regra especifica novos predicados a serem colocados na MT. O sistema baseado em regras é dito ser um sistema dedutivo quando a parte ENTÃO somente contém predicados. Algumas vezes o ENTÃO especifica ações. Neste caso o sistema é dito reativo.

### 3.3 Sistemas Fuzzy

O interesse nos sistemas fuzzy tem aumentado nos últimos anos. A palavra fuzzy tornou-se adjetivo usado para descrever tudo o que não é absolutamente preciso, porém, nem sempre o termo é usado adequadamente. Devido ao uso indiscriminado deste termo, [4] define sistemas fuzzy como: “Qualquer sistema computacional onde os valores não são precisamente predeterminados ou onde a confiança no sistema ou em seus dados possa ser discutida

Case-Based Reasoning (CBR) é uma técnica que utiliza a experiência passada para resolver problemas. A idéia de CBR é descrever e acumular casos significativos para a área de conhecimento especializado e tentar descobrir, por analogia, quando determinado problema é “similar” a um outro resolvido, aplicando a solução armazenada ao novo problema semelhante que surgiu. Um caso é um pedaço de conhecimento contextualizado representando uma experiência. Ele contém a solução do caso e o contexto onde essa solução pode ser usada. Um caso pode ser a descrição de um evento, uma história ou algum registro contendo tipicamente o problema no momento em que o caso ocorreu mais a solução para esse problema [28]. Uma maneira de visualizar um sistema de CBR é em termos de espaços de problemas e espaços de solução. A descrição de um novo problema a ser resolvido é posicionada no espaço do problema. O caso com a descrição mais similar é recuperado e sua solução é encontrada. Se necessário, pode ser feita uma adaptação e uma nova solução pode ser criada. Comparando os sistemas CBR com sistemas baseados em regras pode-se observar que para a criação de regras é sempre necessário saber como resolver o problema, sendo esta tarefa complexa e consumidora de muito tempo. Em sistemas CBR não é necessário saber como resolver os problemas, e sim apenas reconhecer se um problema similar foi resolvido no passado. Por isso, para problemas bem compreendidos que não se

modificam com o tempo, um sistema de regras é mais adequado. Porém, quando o problema é pouco conhecido e dinâmico, o CBR deve ser utilizado. Apesar das diferenças, alguns sistemas híbridos (CBR + regras) têm sido desenvolvidos, tais como em [5], onde é apresentado um sistema para a área jurídica que recupera casos para auxiliar a denúncia de homicídios. Os sistemas CBR e os sistemas de redes neurais possuem como similaridade apenas o fato de se basearem em casos passados. As redes neurais são boas em domínios onde os dados não podem ser representados simbolicamente, tais como reconhecimento de voz e interpretação de

sinais. CBR trata de forma adequada dados estruturados simbolicamente e complexos mas não é tão bom para tratar dados puramente numéricos.

### 3.5 Agentes

Um agente é tudo o que se pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por meio de atuadores [24]. Dado um determinado sistema, denomina-se agente cada uma de suas entidades ativas. O conjunto de agentes forma uma sociedade. As entidades passivas serão designadas pelo termo ambiente. Um agente raciocina sobre o ambiente, sobre os outros agentes e decide racionalmente quais objetivos deve perseguir, quais ações deve tomar, etc. O agente pode ser uma entidade real ou virtual que é capaz de agir em seu ambiente, podendo se comunicar com outros agentes, comportando-se de forma autônoma. Os sistemas multiagentes dividem-se em duas classes principais: Agentes Reativos e Agentes Cognitivos [24]. A abordagem reativa baseia-se na idéia de que agentes com ações elementares podem realizar trabalhos complexos. Segundo [1], nos sistemas multiagentes reativos não há representação explícita do conhecimento; não há representação do ambiente; não há memória das ações; a organização é etológica, ou seja, similar a dos animais; e existe grande número de membros. Os sistemas multiagentes cognitivos são baseados em modelos organizacionais humanos, como grupos, hierarquias e mercados. Nestes sistemas os agentes mantêm uma representação explícita de seu ambiente e dos outros agentes da sociedade; podem manter um histórico das interações e ações passadas; a comunicação entre os agentes é direta, através de mensagens; seu mecanismo de controle é deliberativo, ou seja, os agentes raciocinam e decidem seus objetivos, planos e ações; seu modelo de organização é sociológico; uma sociedade contém poucos agentes. Apesar da classificação, os sistemas multiagentes

podem não ser totalmente cognitivos ou reativos. Um sistema pode ser uma mistura dos dois para atender a solução de um determinado problema. Um tipo de aplicação de agentes que pode também ser usado na Engenharia de Software são os agentes que reduzem o trabalho e o excesso de informação [18]. Uma solução para o excesso de informação está no uso de Assistentes Pessoais, que são agentes autônomos capazes de atuar cooperativamente com o usuário para reduzir trabalho e excesso de informação. Um tipo de assistente pessoal é o agente de interface, capaz de realizar tarefas pelo usuário, treiná-lo, auxiliar na colaboração entre usuários e de monitorar eventos. Os agentes de interface podem ser usados para filtragem de informações, gerência de correio eletrônico, agenda de compromissos, seleção de entretenimento, dentre outros. Em [18] é apresentada uma abordagem para construção de agentes de interface. As principais questões a serem tratadas são quanto a competência do agente, ou seja, como garantir que ele vai adquirir o conhecimento necessário; e a confiança do usuário, ou seja, como garantir que o usuário vai delegar tarefas para o agente. O agente adquire competência através da observação das ações do usuário, através do feedback do usuário quando o agente realiza uma ação, através de exemplos do usuário (treinamento do agente), pedindo conselho a outros agentes e aprendendo quais são os agentes mais confiáveis.

### 3.6 Aprendizado

O elemento central do comportamento inteligente é a habilidade de se adaptar ou aprender a partir de experiências. Existem várias técnicas de aprendizado [24], tais

como:

- por implantação direta do conhecimento: através da inclusão de novas regras diretamente na base de conhecimento;
- por indução: perante um conjunto de exemplos ou dados particulares, o sistema procura inferir conceitos e leis gerais. Este aprendizado pode ser através de exemplos ou por observação e descoberta;
- por analogia: o sistema chega a conclusões sobre uma nova situação a partir de um modelo que foi construído como resultado de experiências anteriores;
- por casos: dados alguns casos de referência, para descobrir uma propriedade de uma situação ou dado particular, são encontrados os casos mais similares de acordo com as propriedades conhecidas;

A máquina de execução do Merlin constrói e atualiza os working contexts dos usuários. Cada atividade

- por redes neurais: Existem vários tipos de redes possui pré-condições que definem, por exemplo, os neurais. Elas podem ser usadas em aprendizado pois que podem desempenhar a atividade, a pessoa supervisionado (backpropagation) ou não supervisionado, os direitos de acesso aos documentos ou a visão (mapa de Kohonen). Em uma rede de dependência com outras atividades. Quando as preneurônios artificiais, existem somadores, multiplicadores de uma atividade são verdadeiras, a atividade cadores e limites. Cada neurônio é disparado quando é inserida no working context. O processo de encadeia influência coletiva de suas entradas atinge um limiar é parecido com o do ambiente Marvel apresenta um mínimo. As entradas possuem pesos que são atualizados anteriormente. O Merlin suporta mudanças dinâmicas no processo, permitindo que um processo não ajustados no processo de aprendizagem. seja totalmente definido antes de iniciar. A linguagem escolhida para o ambiente é baseada em PROLOG.

4 PSEEs que utilizam técnicas de IA

Articulador - [19] é um ambiente baseado em conhecimento para processo de desenvolvimento de software seguir são descritos brevemente alguns ambientes de Engenharia de Software que utilizam técnicas de Inteligência Artificial.

Ele provê um meta-modelo de processo de desenvolvimento de software, uma linguagem baseada em inteligência Artificial.

Marvel - Marvel [11] é um ambiente centrado em objetos e um mecanismo de simulação automática. Para processo resultante de um projeto de mesmo nome na simulação a execução de processos, o Articulador usa uma Columbia University desde 1986. A ideia do projeto era abordagem multiagentes onde os desenvolvedores são desenvolver um ambiente centrado em processo que orientados como agentes cognitivos. A arquitetura do sistema e assiste os usuários que trabalham em um ambiente contém cinco subsistemas: base de conhecimento de grande escala. Para isso, juntaram as áreas de conhecimento, simulador de comportamento, mecanismo de consulta, gerenciador de instanciação e gerenciador de Engenharia de Software e Inteligência Artificial. aquisição de conhecimento. Marvel segue o paradigma de orientação a objetos O meta-modelo do Articulador consiste de recursos, e está baseado em uma linguagem de regras. Ele representa agentes e tarefas. Os recursos são objetos nas tarefas uma ajuda automatizada pela aplicação das estratégias forward e backward chaining sobre as regras, informações dos agentes. As tarefas consomem e produzem revocando as atividades que são parte de uma fase do processo. Um agente representa uma coleção de conhecimento de desenvolvimento. A sua base de objetos é representados por portamentos e atributos associados. O comportamento sistêmico. O processo sendo modelado é decomposto em do agente emerge durante a execução das tarefas (passos de processo. Cada passo é encapsulado em uma incluindo comunicação, acomodação e negociação) levando em consideração a situação do agente. Os agentes mais regras. estes são modelos gerais de

desenvolvedores, times de deMerlin - Merlin [14, 10] é um ambiente de desenvolvimento e organizações. Ferramentas de desenvolvimento de software centrado em processo construído envolvendo são modeladas como subclasse de agentes. dentro de um projeto na University of Dortmund. O As tarefas são representadas através de uma rede de protótipo do ambiente usa o paradigma de regras para ações que os agentes realizam. Como não existe exedescrever e executar processos de software. Uma definição de processo no Merlin possui atividade real de processos neste ambiente, os agentes são dados, papéis (roles), documentos (qualquer objetos) e agentes cognitivos que simulam execução de tarefas e recursos (pessoas, ferramentas, etc.). Um documento é várias situações para auxiliar a geração da descrição do ligado a um conjunto de atividades e a um conjunto de processo. A simulação é importante para detectar falhas ferramentas que suportam as atividades. Nesta aborda- como alocação de recursos, cronograma, dentre outras. Pandora - Pandora [15] é uma máquina de processamento os usuários recebem todas as informações relevantes baseada em programação em lógica e conceitos de tempo em um espaço de trabalho chamado working context associado ao papel do usuário. Este espaço de trabalho lógica temporal. Todos os eventos são registrados e o contém os documentos a serem manipulados, suas descrições possui um algoritmo de aplicação de regras que pendências com outros documentos e as atividades que otimiza os passos de execução. Além disso, o sistema devem ser realizadas. Quando uma atividade realizada possui um mecanismo de sincronização que garante que influencia um working context, acontece uma atualização- as atividades cooperativas ou que devem ser executadas dinâmica nos working contexts dependentes deste em alguma ordem serão sincronizadas. evento. Pandora integra os paradigmas de eventos para mo-

- por construção de árvores de identificação;

delar interatividade entre mensagens e de regras para representar o conhecimento do processo. O conhecimento pode ser declarativo ou procedural. Conhecimento declarativo descreve o domínio do discurso, no caso, o modelo de processo enquanto conhecimento procedural descreve o comportamento que o processo pode assumir durante o desenvolvimento, ou seja, as regras e eventos que disparam evolução de processo. O sistema reage aos eventos externos e dispara as ações internas. As regras e eventos são especificados em uma linguagem de lógica de primeira ordem acrescida de operadores de lógica temporal (Linguagem Pandora) onde o tempo é caracterizado por uma linha seqüencial simples de eventos. Isto permite expressar quantitativamente o tamanho dos intervalos temporais, a distância temporal entre os eventos e a viabilidade das atividades modeladas quanto às proposições lógicas estabelecidas. É possível estabelecer não apenas quais atividades são permitidas a cada momento, mas também como as atividades interagem (ou seja, sincronização entre atividades paralelas e disparo de atividades engatilhadas). Pandora é implementado em PROLOG e consiste de dois módulos principais: compilador de regras e interpretador de regras. O compilador checa as regras quanto a sintaxe e as transforma em uma representação interna. O interpretador permite ativação das ferramentas do sistema operacional, permite rastreamento da aplicação de regras e observação dos estados internos. 5 Avaliação e Propostas



o paradigma. As causas para essa adoção podem estar no requisito de modificação dinâmica durante execução do processo, sendo o paradigma de regras apontado como flexível a mudanças, ou ainda a necessidade de raciocínio baseado em conhecimento. O conhecimento sobre processos pode ser armazenado em uma base de conhecimento e regras podem ser utilizadas para obter novas informações ou auxiliar no acesso a esse conhecimento. Os SEEs encontrados não se comportam como sistemas especialistas. O conhecimento sobre modelos de processo fica embutido no ambiente e os modelos construídos são desenvolvidos por projetistas de processo, que o fazem como um roteiro do desenvolvimento de software. Este roteiro pode ser modificado a qualquer momento. Portanto o conhecimento adquirido não provém de um especialista em processo de desenvolvimento de software visando resolver ou diagnosticar uma situação e sim de experiências em modelagem de processos. O PSEE estudado que mais utiliza técnicas de IA é o Articulator. Além de ser baseado em regras para modelagem e execução do processo, ele utiliza o paradigma de multiagentes para modelar o comportamento dos agentes do processo. Como o ambiente permite somente simulação do processo, os agentes são modelados para agirem da mesma forma que agiriam agentes humanos. Neste caso, o conhecimento a ser modelado é complexo e baseia-se em perícia, experiência, estilo de trabalho, dentre outras características dos agentes. A este ambiente foi integrada uma abordagem para diagnóstico, replanejamento e re-escalonamento chamada Articulation. Esta abordagem é muito útil, pois o conhecimento sobre as falhas da execução dos processos vai sendo armazenado e gera heurísticas para a solução de novas falhas. Outra ferramenta incorporada ao Articulator é a biblioteca de processos SPLib. Ela permite reutilização de processos e possui uma base de conhecimento de processos que organiza os modelos e suas dependências. A forma de acesso a um processo desta biblioteca é similar à técnica case-based reasoning uma vez que o usuário deve fornecer algumas características desejáveis do processo e a biblioteca faz o matching dos processos mais adequados. Pode-se observar que nem todas as técnicas de Inteligência Artificial são utilizadas pelos PSEEs, como redes neurais por exemplo. Isto se justifica porque o modelo de processo e sua execução necessitam de dados simbólicos e bem estruturados e o uso de redes neurais é mais adequado quando os dados não podem ser representados simbolicamente, como reconhecimento de voz por exemplo. Porém existe um caso de utilização de re-

Esta seção apresenta a avaliação do uso das técnicas de IA em PSEEs levando em consideração aspectos de modelagem, reutilização de processo, execução, simulação de processo, diagnóstico e recuperação de falhas na execução, arquitetura do SEE e interface com os usuários do ambiente. Em seguida são apresentadas algumas propostas para utilização de técnicas de IA em PSEE visando facilitar o uso e aumentar a produtividade tanto do desenvolvedor quanto do gerente.

### 5.1 Avaliação

Os ambientes pesquisados permitem modelagem e execução de processos de software, sendo que somente o Articulator permite simulação. A adoção de notações gráficas para modelagem de processos foi observada no ambiente Merlin. O paradigma de regras como forma de representação do processo ou como modelo de execução foi totalmente adotado pela tecnologia de processos face à grande quantidade de ambientes existentes que utilizam

des neurais em [3] para geração automática de modelos formais do comportamento de processos de software a partir da execução utilizando-se uma técnica chamada process discovery, que foi testada usando redes neurais, cadeias de Markov e um método algorítmico, mas o teste com redes neurais não foi satisfatório. A seguir é apresentada uma tabela comparativa dos ambientes estudados.

a recuperação de processos de software adequados à situação que se quer modelar, podem ser usados sistemas de case-based reasoning como na biblioteca SPLib do sistema Articulator.

### 5.2.2 Execução

Figura 1: Comparação do uso de técnicas de IA pelos ambientes centrados em processo

## 5.2

### Propostas

As técnicas de IA ainda são muito pouco exploradas na Engenharia de Software. As principais utilizações se restringem ao paradigma de regras para modelar sistemas baseados em conhecimento que auxiliam o desenvolvimento de software. Este artigo procurou características de utilização de Inteligência Artificial em PSEEs. Foram apresentadas algumas técnicas de IA e alguns ambientes, com o objetivo de avaliar a utilização das técnicas nos ambientes. Entretanto o uso ainda é muito restrito, o que nos leva a propor algumas utilizações. Nas seções a seguir são apresentadas algumas idéias a respeito das técnicas de IA que podem ser utilizadas em modelagem, reutilização, execução, simulação e interface com o usuário de um PSEE [22].

#### 5.2.1 Modelagem e Reutilização

Na fase de execução do meta-processo de desenvolvimento de software, o componente principal é a máquina de processo. Este componente centraliza o controle do desenvolvimento de software e é responsável por inúmeras tarefas, como verificar consistência, direitos de acesso, próximas atividades a serem executadas, agendas dos usuários, estados das atividades, dentre outras. Para distribuir essas tarefas poderiam ser utilizados agentes da IA. Um agente poderia cuidar das agendas dos usuários, outro das atividades e seus estados e assim por diante. Uma vantagem dessa abordagem é a capacidade de aprendizado dos agentes, os quais poderiam agir de forma mais apropriada às peculiaridades do ambiente e seus usuários. Neste caso utilizaria sistemas multiagentes cognitivos. Os agentes reativos também poderiam ser usados para coletar métricas, por exemplo. Eles observariam o que acontece no ambiente e aumentariam a base de métricas com informações importantes sobre a execução do processo. Há também a possibilidade de se utilizar lógica fuzzy na execução de processos. A máquina de execução, ao observar o atraso em uma atividade, pode detectar que o desenvolvedor que causou o atraso é “preguiçoso”. Assim, pode existir uma regra fuzzy que verifica se a atividade está atrasada e o desenvolvedor é preguiçoso, então delegar automaticamente a tarefa para um agente mais responsável. A variável preguiçoso seria um conjunto fuzzy derivado de várias características [22].

Case-based reasoning também seria aplicável na fase de execução pois em casos de parada na execução, o sistema poderia buscar um caso parecido e resolver o problema com uma solução testada.

### 5.2.3 Simulação

Uma técnica de IA que aumentaria o poder de expressão de um modelo de processo seria a lógica fuzzy. Através de regras fuzzy, situações do tipo “se nível de qualidade da tarefa anterior é alto, então faça operação 1”. Neste caso, o nível de qualidade depende de uma avaliação que não é precisa, mas que é importante para o desenvolvimento do processo. A reutilização de processos de software é altamente necessário e tem sido proposto em vários trabalhos. Para

O SEE deve suportar validação e verificação de modelos de processo. Através da simulação de processos de software o ambiente e os usuários podem detectar falhas, inconsistências e comportamentos anômalos na descrição do processo. Conflitos no cronograma e alocação de recursos são comuns. Além disso, o usuário pode prever o progresso do desenvolvimento e comparar modelos [22]. Na simulação de processos não existe o desenvolvedor humano. Portanto é preciso simular o comportamento dele. A abordagem de agentes tem sido utilizada

para essa finalidade como apresentado pelo ambiente Articulator [19]. A simulação de processos pode usar também lógica fuzzy da mesma forma que foi proposta na execução (seção 5.2.2). Para que a simulação atinja seus objetivos de apontar os problemas da modelagem e da execução é necessário que ela seja baseada em situações reais. Portanto um sistema de case-based reasoning poderia ser usado para prover conhecimento real sobre situações de execução de processo e como são resolvidas essas situações. Na simulação também poderia se utilizar características reais dos agentes humanos para modelar os agentes cognitivos. Dessa forma, a simulação poderia auxiliar a resolver conflitos entre os desenvolvedores e a selecionar atividades para os mesmos.

### 5.2.4 Interface

identificar isso e filtrar as informações de acordo com o nível do usuário facilitando assim o aumento da produtividade no desenvolvimento de software.

## 6 Conclusão

A interface com o usuário de um PSEE depende do papel que ele executa no desenvolvimento. Um gerente precisa ver resultados quantitativos e qualitativos do trabalho dos desenvolvedores. As suas ações visam ajustar o processo para que não atrase e atinja os objetivos, sendo que para isso ele necessita interagir com os desenvolvedores. O desenvolvedor precisa ver as tarefas a serem desenvolvidas e os objetos a serem manipulados. Precisa também priorizar e delegar tarefas quando possível. Os agentes de interface poderiam auxiliar esses usuários. Da mesma forma que um agente pode adquirir competência no tratamento das mensagens de seus usuários, ele pode adquirir competência no tratamento das tarefas que o desenvolvedor deve executar e tornar-se seu assistente pessoal de processo. O agente pode auxiliar um usuário novato indicando as tarefas mais importantes e fornecendo todas as informações necessárias para o desenvolvimento das mesmas. Por outro lado, com um usuário mais experiente, o agente pode deduzir que o mesmo não necessita de tantas informações para realizar seu trabalho e apresentar somente as informações necessárias. Adquirindo competência, o agente pode

automaticamente delegar tarefas para outros desenvolvedores, marcar reuniões e trocar informações com outros agentes para solucionar os problemas que surgirem [22]. Acredita-se que esta abordagem pode aumentar a utilização de PSEEs, pois um dos principais problemas da execução automatizada é a obrigação do seguimento de alguns passos do processo. Um desenvolvedor experiente não fica satisfeito com um sistema que o obriga a seguir passos programados e informa a todo instante como fazer seu trabalho. Um agente de interface pode

As técnicas de Inteligência Artificial tem sido cada vez mais utilizadas em várias áreas da ciência da computação. Seu uso se justifica pelo auxílio fornecido ao raciocínio humano em tarefas que se beneficiam da experiência e do conhecimento de especialistas. Neste artigo foi apresentada a tecnologia de processos de software e suas características com o objetivo de encontrar utilização ou necessidade de utilização de técnicas de IA. Os PSEEs Marvel, Merlin, Articulator e Pandora foram estudados e avaliados. Com isso, podemos dizer que o ambiente que mais utiliza técnicas de IA dentre os apresentados é o Articulator [19], pois, permite simulação de processos e usa um modelo multiagentes para representar o comportamento dos desenvolvedores. Também foram apresentadas algumas propostas de utilização de técnicas de IA em modelagem, execução, simulação, reutilização e interface do ambiente. As propostas foram derivadas do estudo das técnicas de Inteligência Artificial e do estudo da arquitetura e funcionamento dos PSEEs, sendo que essas propostas podem vir a ser implementadas em um futuro próximo, de forma a contribuir para o aumento da qualidade do software. A utilização de técnicas de IA na Engenharia de Software ainda necessita de estudos e experiências, porém estão surgindo ambientes que se preocupam com os fatores psicológicos do desenvolvimento de software e sua facilidade de uso. As pesquisas na área apontam para o objetivo de tornar o PSEE um gerente de auxílio que não influencie a capacidade e a vontade dos desenvolvedores, mas ao mesmo tempo controle os recursos e os direitos de acesso, colete métricas e disponibilize análises sobre o desempenho do processo de desenvolvimento de software. Acredita-se que as técnicas de IA ainda tem muito a contribuir para o desenvolvimento dessa área.

### **Documento 3: Inteligência Artificial e Aprendizado**

Este tutorial tem por objetivo apresentar uma introdução ao aprendizado artificial e automatizado (machine learning), focalizando-se sobre os aspectos referentes a uma técnica em particular, as redes neurais artificiais –R.N.A. Na primeira seção vamos discutir sobre a Inteligência Artificial, sobre a aquisição de conhecimentos e sobre a importância do aprendizado na construção de sistemas inteligentes. Na segunda seção iremos abordar as redes neurais artificiais (modelos conexionistas), onde vamos destacar: os diferentes tipos de redes e de algoritmos de aprendizado existentes; a representação do conhecimento neural; as características e limitações de uso deste tipo de técnicas, bem como mostraremos alguns exemplos de aplicações das RNAs. Para concluir, iremos discutir sobre os caminhos da pesquisa atual nesta área e tendências futuras no que diz respeito ao desenvolvimento dos sistemas inteligentes.

#### **1. Inteligência Artificial e Aprendizado**

Para podermos falar em Inteligência Artificial e Aprendizado Artificial, precisamos antes saber o que é “inteligência” e o que entendemos por “aprendizado”? Apesar de termos uma noção básica do que significam estas duas palavras, inteligência e aprendizado, temos uma grande dificuldade de defini-las em termos práticos e de forma bastante precisa.

### 1.1. O que é Inteligência?

O que é inteligência? O que é um ser inteligente? Estas duas questões devem ser respondidas (ao menos de forma superficial) se quisermos então partir para a implementação dos sistemas inteligentes. A idéia de implementação dos sistemas inteligentes parte do pressuposto que iremos “copiar” a inteligência humana e colocá-la a nosso serviço através de implementações que automatizem este tipo de processo/comportamento.

## REPRODUZIR A INTELIGÊNCIA HUMANA

Figura 1.1 – Inteligência: Natural e Artificial O termo “Artificial Intelligence” (A.I. – I.A. em português) foi usado pela primeira vez em 1956 por McCarthy (e desenvolvido por grande pesquisadores como Marvin Minsky e Herbert Simon), e nada mais é do que uma tentativa de formalizar o eterno sonho da criação de um “cérebro eletrônico” (termo muito usado na ficção científica e mesmo na época inicial do desenvolvimento dos computadores). Encontramos também algumas definições interessantes na literaturas, tais como: (1) Definição de Inteligente: Dotado de inteligência, capaz de compreender, esperto, habilidoso [Larousse 99];

(2) Definição de Inteligência: Faculdade de conhecer, de aprender, de conceber, de compreender: a inteligência distingue o homem do animal [Larousse 99]; (3) Definição de Inteligência Artificial: Conjunto de teorias e de técnicas empregadas com a finalidade de desenvolver máquinas capazes de simular a inteligência humana [Larousse99]; (4) Definição de Inteligência Artificial: A Inteligência Artificial é uma área de estudos da computação que se interessa pelo estudo e criação de sistemas que possam exibir um comportamento inteligente e realizar tarefas complexas com um nível de competência que é equivalente ou superior ao de um especialista humano [Nikolopoulos 97]. Destas definições acima, podemos fazer os seguintes comentários: As definições são usualmente recursivas, ou seja, um ser inteligente é aquele que possui inteligência, e a inteligência é a característica dos seres inteligentes; As definições possuem contradições, onde aparece que a inteligência distingue o homem do animal, pois só o homem é capaz de aprender, conhecer, compreender. Isto não é muito preciso/correto, pois os animais também podem aprender (a reconhecer o dono), conhecer (o caminho de volta para casa) e compreender (um comando de seu mestre). E podemos mesmo dizer que estas tarefas realizadas por animais são tipicamente tarefas consideradas na atualidade como “tarefas inteligentes” que os computadores ainda não realizam com um desempenho satisfatório. Uma pessoa com limitações (analfabeta, deficiente mental e/ou físico, com QI reduzido) poderá ser qualificada, pelos critérios acima, como inteligente? E no entanto ela é um ser inteligente com capacidades superiores as de um computador, ou você acha que, por exemplo, um computador conseguiria ir até uma farmácia comprar um remédio? (isto inclui tarefas de: planificação de trajetória, reconhecimento do caminho, tratamento de imprevistos, etc).

Esta discussão poderia se prolongar aqui de forma indefinida, mas acredito que um ponto importante a ser considerado é: se queremos reproduzir a inteligência humana, devemos ter consciência de quais aspectos da inteligência que gostaríamos que fossem reproduzidos, e que em nossa opinião, permitiriam o desenvolvimento de um sistema com propriedades ditas “inteligentes”. Esta pergunta foi feita em sala de aula para um conjunto de alunos da Unisinos, e o resultado foi bastante interessante, onde reproduzo aqui as respostas de forma a complementar esta discussão: O que é Inteligência? Associação de idéias e conceitos; Concluir coisas; Capacidade de aprendizado; Acúmulo de conhecimentos; Raciocínio: lógico, abstrato, dedução, analogia, indução, inferência, síntese, análise, ...; Uso de experiências e conhecimentos passados; Uso prático de conhecimentos e experiências; Tomada de decisões; Criar coisas novas (criatividade); Saber o que eu sei (capacidade de explicar algo); Saber que sei / Saber que não sei; Interação; Comunicação. (espaço reservado para você continuar esta reflexão!)

Vamos aqui usar a definição de inteligência (que nos convêm) que é baseada no seguinte ditado: “Errar é humano, mas repetir o erro é burrice”. Logo o conceito de inteligência (por oposição à burrice, e devido ao fato desta ser uma propriedade humana), indica que o ser humano (ou não) inteligente deve ser capaz de: adaptar-se, aprender, evoluir, corrigir seus erros. E assim estamos passando da questão “o que é um ser inteligente” para a questão seguinte “o que é aprendizado”.

### 1.1. O que é Aprendizado?

O aprendizado natural é também um conceito difícil de ser definido, mas também de grande importância para que possamos partir para a construção de sistemas inteligentes dotados da capacidade de aprendizado. Sugerimos ao leitor que faça aqui o uso de sua inteligência natural a fim de definir o que você entende por aprendizado natural e quais são as propriedades que estão associadas a este conceito. Depois sugerimos ao leitor que, de posse dessa sua definição/descrição, passe a analisar os próximos itens abordados neste tutorial, com um olhar crítico sobre o mesmo no que diz respeito a presença ou não destas propriedades associadas aos métodos que iremos apresentar. Vamos tentar definir aqui brevemente o que entendemos (o que o autor entende) por aprendizado. Aprendizado é a capacidade de se adaptar, de modificar e melhorar seu comportamento e suas respostas, sendo portanto uma das propriedades mais importantes dos seres ditos inteligentes, sejam eles humanos ou não. A capacidade de aprender está ligada diretamente aos seguintes itens: Adaptação e mudança de comportamento de forma a evoluir (melhorar segundo algum critério). Um sistema, biológico ou artificial, que não seja capaz de evoluir ou de mudar seu comportamento diante de novas situações que lhe são propostas é um sistema sem inteligência; Correção dos erros cometidos no passado, de modo a não repeti-los no futuro. Este item está diretamente relacionado ao anterior, o sistema deve modificar seu comportamento caso o comportamento atual não satisfaça a algum tipo de exigência (onde a sobrevivência deve ser um dos quesitos mais importantes a serem considerados nos seres vivos); Otimização: melhoria da performance do sistema como um todo. O aprendizado pode implicar em uma mudança do comportamento que busque: a economia de energia gasta para realizar uma tarefa, a redução do tempo gasto numa tarefa, etc. Quando falamos em otimização, devemos lembrar que podemos ter quesitos contraditórios e opostos, onde teremos que maximizar ou minimizar custos de acordo com algum tipo de critério; Interação com o meio, pois é através deste contato com o mundo que nos cerca que podemos trocar experiências

e/ou realizar experiência, de forma a adquirir novos conhecimentos; Representação do conhecimento adquirido. O sistema deve ser capaz de armazenar uma massa muito grande de conhecimentos, e isto requer uma forma de representar estes conhecimentos que permita ao sistema explorá-los de maneira conveniente. Como nossos recursos são limitados, devemos ter uma maneira de guardar conhecimentos e regras gerais, pois guardar tudo seria muito difícil (exige muita memória, dificulta o uso pela lentidão da consulta aos conhecimentos).

O aprendizado é um tema bastante interessante e amplo, onde insisto que o autor deste tutorial irá centrar a noção de inteligência no que diz respeito a capacidade de um sistema aprender. Visto que este tema é muito amplo, sugerimos a leitura de obras como a Sociedade da Mente [Minsky 85], Regras da Mente [Anderson 93] e Como a Mente Funciona [Pinker 99] como bases para uma discussão mais profunda sobre os questionamentos relativos a inteligência humana e aprendizado. Como já foi dito, iremos nos concentrar no tema “aprendizado”, uma característica básica das redes neurais, mas antes de passarmos ao aprendizado conexionista (neural), achamos importante dar uma visão histórica sobre os métodos e conceitos básicos da Inteligência Artificial Simbólica (clássica).

### 1.3. Inteligência Artificial – Processamento Simbólico

A I.A. simbólica ficou conhecida nas décadas de 70/80 pela aparição dos sistemas especialistas, dos sistemas baseados em conhecimentos (KBS – Knowledge based systems), e da expansão do uso de linguagens como Prolog (Programação em Lógica). Nesta mesma época um dos grandes desafios à “inteligência” dos computadores eram jogos, destacando-se o jogo de Xadrez. Os japoneses também entraram na corrida pela criação de ferramentas de inteligência artificial, e propuseram a criação dos computadores de 5a. Geração (computadores inteligentes, capazes de escutar, falar e raciocinar). Do outro lado do oceano, os americanos também criavam seus mega-projetos, onde um dos mais famosos foi o projeto CYC, que visava dotar um computador de conhecimentos, muitos conhecimentos de forma a torná-lo um grande “cérebro eletrônico” (o nome CYC vem da palavra Encyclopaedia). Em meados dos anos 80 e no início dos anos 90 a I.A. atravessou (e ainda atravessa?) uma grande crise de identidade. Os resultados práticos obtidos com a I.A. simbólica, de certa maneira, eram decepcionantes. Os computadores progrediram em grandes passos (memória, velocidade, softwares, etc), e a I.A. não parecia estar acompanhando este progresso. A linguagem Prolog tinha seus atrativos, mas faltava alguma coisa (até a Borland chegou a lançar o seu Turbo Prolog, e o que houve com o TProlog?). O projeto CYC esbarrou em problemas de gerência de uma grande base de conhecimentos, com muitas contradições, e até hoje está tentando representar o “senso comum” com um “bom senso”. O computador japonês de 5a. Geração até sabe falar e escutar, mas ainda não compreende a linguagem humana. A síntese e o reconhecimento de voz são talvez os maiores avanços deste final de século, mas ainda enfrentam muitos problemas e não passam de apenas mais uma forma de comunicação homem-máquina, sem no entanto haver uma verdadeira interação do ponto de vista lingüístico. E não podemos esquecer o Xadrez, pois o maior especialista do mundo, Kasparov, foi derrotado recentemente pelo Deep Blue da IBM. Mas o que significa realmente esta derrota? A I.A. atingiu os seus objetivos? O Deep Blue é uma máquina inteligente, e mais ainda, mais inteligente que toda a raça humana pois ganhou de nosso melhor representante? Não, de forma alguma podemos dizer que o Deep Blue é uma máquina inteligente... ele apenas possui uma capacidade de

cálculo enorme, o que lhe deu uma grande vantagem sobre os seres humanos, mas seus conhecimentos, estes provêm dos seres humanos que lhe programaram. O Deep Blue é como uma grande calculadora, tão precisa e veloz que não tem ser humano que possa ganhar dele, mas tão burro quanto a calculadora. E como ele fez para ganhar, se o seu opositor era tão inteligente? Simples, ele podia prever todas as combinações possíveis de jogadas, em um número muito superior a capacidade humana (simples problemas combinatorial) e além disso armazenava em sua memória uma base de dados enorme de jogos clássicos de Xadrez já disputados por Kasparov e outros (simples banco de dados). Onde está a inteligência e a capacidade de aprendizado? Em uma das partidas o Deep Blue perdeu (entrou em loop), mas ele não se recuperou, foi resetado e reprogramado para evitar explicitamente aquela situação que lhe causou problemas.

Em 1999 o homem chegou lá, desenvolveu máquinas que sintetizam e reconhecem a voz e que ganham no Xadrez do maior especialista humano, mas que não apresentam muitas daquelas propriedades que foram discutidas nos itens 1.1 e 1.2. Deixamos para o leitor dar o seu veredito sobre a I.A. do final dos anos 90. Mas não seja tão cruel em seu veredito, pois muito, muito foi feito nestes últimos 15 anos. Vejamos como evoluíram os sistemas especialistas. Inicialmente os Sistemas Especialistas (de 1a. Geração [Nikolopoulos 97]) se baseavam em uma arquitetura como mostra a figura 1.2. O problema destes sistemas logo apareceu, a aquisição de conhecimentos não era automática e dependia do especialista e/ou engenheiro de conhecimento para que fossem adicionados novos conhecimentos ao sistema. Surgia assim o conhecido “gargalo da aquisição de conhecimentos” (Knowledge Acquisition bottle-neck) dos sistemas especialistas.

Sistemas Especialistas - Esquema Global Usuário Módulo de Explicação

Base de Conhecimentos Motor de Inferência Interface com o usuário Módulo de Aquisição de Conhecimentos Expert

Figura 1.2 – Arquitetura dos Sistemas Especialistas Os sistemas especialistas de 2a. Geração introduziram a aquisição automática de conhecimentos, e então começamos a ouvir falar de aprendizado de máquinas simbólico (symbolic machine learning).

Base de Conhecimentos (regras + fatos) Conversão para um formato de representação interno Aquisição Explicação Automática Conhecimentos sobre uma área de aplicação: • Conhecimentos do especialista • Teorias sobre o domínio de aplicação • Resultados de experiências praticas (casos conhecidos)

Figura 1.3 – Aquisição de Conhecimentos: Explicação e Aprendizado Automático

A Inteligência Artificial começava a ter uma nova “cara” (vide fig. 1.4) onde deu-se mais importância a representação de conhecimentos e ao aprendizado, e não mais somente aos métodos de raciocínio automatizados (motor de inferência, processo de unificação, etc).

Inteligência Artificial

Sistemas Inteligentes Aprendizado de Máquinas Representação de Conhecimentos Sistemas Especialistas KBS, robótica, visão artificial, ... CBR, ILP, árvores de decisão/indução, redes neurais algoritmos genéticos, ... Métodos Simbólicos Métodos Conexionistas (RNA)



Figura 1.3 – Inteligência Artificial: Uma visão moderna Como pode ser visto na figura 1.3, a representação de conhecimentos é uma espécie de divisor, que irá separar de um lado os métodos simbólicos e de outro os métodos conexionistas (redes neurais). Na realidade esta divisão não deve ser encarada como uma separação, mas sim como dois tipos de métodos que possuem cada qual suas peculiaridades e que devem ser entendidos e estudados de maneira a tirar o maior proveito de cada um. Fala-se muito atualmente em sistemas híbridos, multi-agentes ou múltiplas inteligências, como sendo uma direção para onde deve se orientar a I.A. do novo milênio [Osório 99].

#### 1.4. Inteligência Artificial – Aprendizado Simbólico

As ferramentas de I.A. simbólica, em sua evolução, também foram dotadas de mecanismos de aquisição automática de conhecimentos, onde podemos citar o exemplo da linguagem Prolog e dos sistemas especialistas (2a. Geração) que foram dotados de mecanismos de aprendizado de máquinas. Os principais métodos (ou pelos menos os mais conhecidos) de aprendizado simbólico são: Aprendizado por analogia/por instâncias. Exemplo: Sistemas baseados em casos - CBR (Case Based Reasoning) [Mitchell 97, Kolodner 93]; Aprendizado por Indução. Exemplos: Árvores de Decisão - ID3, C4.5, CN2 (IDT - Induction of Decision Trees) [Quinlan 92], e ILP - Inductive Logic Programming (Prolog) [Nilsson 98]. Aprendizado por evolução/seleção. Exemplo: Algoritmos Genéticos - GA e GP (Genetic Algorithms / Genetic Programming) [Goldberg 89, Mitchell 97]; Outros tipos de aprendizado: por reforço (reinforcement learning), não supervisionado, bayesiano e por explicações (explanation based) [Mitchell 97, Nilsson 98].

Estas ferramentas de aprendizado possuem fortes limitações: (1) usualmente assumem que os conhecimentos a serem adquiridos/manipulados e as informações disponíveis são corretos e devem estar completos em relação ao problema (o que dificilmente ocorre); (2) são orientadas para manipular informações simbólicas (informações qualitativas), onde os valores numéricos e contínuos (informações quantitativas) representam um problema difícil de ser tratado. Em relação a este último item, existem tentativas de integrar ao processamento simbólico probabilidades (regras bayesianas) e incerteza (regras fuzzy), como forma de expandir as potencialidades deste tipo de sistemas.

#### 1.4. Inteligência Artificial – Alternativas ao aprendizado e processamento simbólico

O processamento e aprendizado simbólico, devido as suas características básicas, possuem algumas limitações no que diz respeito a manipulação: de incertezas, de valores aproximados, de informações contraditórias, e de uma maneira geral, de informações quantitativas [Minsky 90]. O mundo que nos cerca é extremamente “impreciso” do ponto de vista computacional, e sendo assim, as ferramentas computacionais de I.A. devem portanto ser bastante robustas para poderem trabalhar neste ambiente de informações imprecisas (informações incorretas) e cheio de problemas imprevisíveis (informações incompletas). Muitos pesquisadores, na sua busca da implementação de ferramentas e sistemas inteligentes, se orientaram para os sistemas baseados “realmente” no funcionamento do cérebro humano. Este tipo de enfoque levou muitos pesquisadores (entre eles Marvin Minsky, Seymour Papert e John Von Neumann) a estudarem novas formas de processamento de informações baseadas nos estudos neuro-fisiológicos do cérebro. Esta corrente de pesquisas tentava reproduzir os neurônios como elementos básicos do

processamento de novas arquiteturas de máquinas inteligentes, ao invés de usar portas lógicas, bits e bytes controlados por uma Unidade Central. Esperava-se que de elementos de processamento baseados em neurônios, conectados entre si com um grande número de ligações, e operando em paralelo, pudessem “emergir comportamentos inteligentes”. Este ramo da I.A. foi denominado de Inteligência Artificial Conexionista.

## 2. Redes Neurais Artificiais

### 2.1. Introdução

As Redes Neurais Artificiais (RNA), também conhecidas como métodos conexionistas, são inspiradas nos estudos da maneira como se organiza e como funciona o cérebro humano. Este tipo de método possui características peculiares de representação e de aquisição de conhecimentos, sendo considerado um método de nível sub-simbólico (em oposição aos métodos simbólicos, citados na seção anterior: árvores de decisão, CBR, KBS, etc). Inicialmente vamos discutir sobre a representação de conhecimentos utilizada pelas Redes Neurais, para depois analisarmos a parte referente ao aprendizado destas. É importante salientar que existem diferentes tipos de RNAs e que cada um destes modelos diferentes possui suas características próprias quanto a forma de representar e de adquirir conhecimentos. Em função disto vamos primeiramente apresentar uma visão geral, classificando os diferentes modelos de RNAs para em seguida nos concentrarmos em um modelo mais específico: as redes neurais artificiais do tipo multi-nível baseada em Perceptrons (conhecidas como Multi-Layer Perceptron Nets ou Back-Propagation Nets). A figura 2.1 apresenta um exemplo deste modelo de rede conexionista.

Figura 2.1. Exemplo de Rede Neural Artificial do tipo Multi-Nível

### 2.2. Conceitos Básicos, Origem e Evolução das Redes Neurais

Vamos discutir inicialmente como se estruturam as redes neurais e seus neurônios, passando depois para uma apresentação da origem e evolução dos estudos nesta área.

#### 2.2.1. Representação de Conhecimentos

A representação de conhecimentos nas redes conexionistas, como diz o próprio nome, é fortemente ligada a noção de conexão entre neurônios (elementos processadores de informação) que interagem uns com os outros através destas ligações. O modelo conexionista possui sua origem nos estudos feitos sobre as estruturas de nosso cérebro – sofrendo uma grande simplificação do modelo original – onde encontramos no modelo artificial, que é simulado, elementos como os neurônios e as suas conexões, chamadas de sinapses. A proximidade e fidelidade dos modelos artificiais em relação ao modelo real é um tema polêmico que não vamos nos aventurar a discutir aqui, deixando apenas o registro da origem dos conceitos básicos que norteiam este campo de pesquisas. O conhecimento de uma RNA está codificado na estrutura da rede, onde se destacam as conexões (sinapses) entre as unidades (neurônios) que a compõe. Nestas redes artificiais obtidas por simulação em computadores, associamos a cada conexão um peso sináptico (valor numérico) que caracteriza a força da conexão entre dois neurônios. O aprendizado em uma RNA é realizado por um processo de adaptação dos seus pesos sinápticos. As figuras 2.2 e 2.3 mostram a relação entre os neurônios naturais e o modelo de neurônio artificial.

Sinapse Núcleo Corpo Celular Segmento Inicial Dendrito Axônio Sinapse

### Figura 2.2. Exemplo de Neurônio Natural

Uma vez que os conhecimentos da rede estão codificados na estrutura de interconexões entre os neurônios e nos pesos associados a estas conexões, fica muito difícil para um ser humano realizar uma análise e interpretação dos conhecimentos adquiridos por uma RNA. Os conhecimentos das redes se resumem a um conjunto de valores numéricos descrevendo as conexões, e por consequência, estes valores descrevem também o comportamento da rede. Entretanto, para um ser humano estes dados não fazem muito sentido. Na seção 2.3 vamos apresentar mais em detalhes os diferentes tipo de redes, onde a escolha do tipo de neurônio artificial utilizado é fundamental para se definir como será a representação interna dos conhecimentos da rede.

### Pesos Sinápticos

(Considera o conjunto de valores de entradas e os seus respectivos pesos sinápticos)

Figura 2.3. Exemplo de Neurônio Artificial Apesar dos estudos sobre as redes neurais artificiais serem considerados como pertencentes à uma área “jovem”<sup>1</sup> de pesquisa, encontramos atualmente uma série de referências e exemplos de importantes aplicações práticas deste tipo de método de aprendizado, onde podemos citar alguns exemplos de obras relevantes na área, como por exemplo [Fiesler 97, Ripley 96, Bishop 95, Arbib 95, Krose 93, Freeman 92, Simpson 90, Faq 99a].

**2.2.2. Origem e Evolução** Os primeiros estudos sobre as Redes Neurais Artificiais e propostas de modelos destas redes surgiram nos anos 40. Os primeiros modelos evoluíram bastante, onde alguns deles se destacaram e tornaram-se famosos, mas mesmo assim até hoje continuam sendo propostos novos modelos de redes neurais. O estudo dos primeiros modelos e de sua evolução nos ajuda a entender melhor as redes neurais, e seu estágio atual de evolução.

Os primeiros estudos sobre as redes neurais artificiais remontam aos anos 40 (McCulloch-Pitts), mas foi somente na década de 80 que houve um grande desenvolvimento nesta área (Back-Propagation, Hopfield, Kohonen SOFM, ...)

O começo do estudo das rede neurais artificiais pode ser atribuído à criação do Psychon em 1943 por McCulloch e Pitts [McCulloch 43], sendo que alguns anos mais tarde, em 1949 D. O. Hebb publicava uma importante obra, o livro “The Organization of Behaviour” [Hebb 49], que influenciou vários modelos de RNAs de destaque na atualidade. Em 1959, Frank Rosenblatt criou o Perceptron [Rosenblatt 59] que, como será visto neste trabalho, tem até hoje uma grande influência sobre os estudos das redes neurais, mostrando que apesar desta área de estudos ter crescido muito na atualidade, suas bases foram estruturadas juntamente com a criação dos fundamentos da ciência da computação. Alguns outros modelos similares ao Perceptron foram também desenvolvidos nesta época, como é o caso do Adaline (Adaptive Linear Element), criado por Bernard Widrow em 1962 [Widrow 62, 88, 90]. Os modelos do tipo Perceptron, incluindo o Adaline, são baseados no aprendizado supervisionado por correção de erros, uma classe muito importante de redes neurais artificiais, que possui uma larga aplicação na atualidade. Em 1969 os modelos baseados no Perceptron receberam uma dura crítica feita por Minsky e Papert através de sua obra “Perceptrons: An Introduction to Computational Geometry” [Minsky 69]. Através deste livro, Minsky e Papert provaram matematicamente que os modelos de redes neurais baseados no Perceptron (redes

de um só nível, o que na época era o tipo de rede de Perceptrons utilizado), não eram capazes de aprender uma simples função lógica do tipo “ou-exclusivo” (XOR = Exclusive Or). A função XOR possui um padrão de valores de entrada e de saída cuja associação não podia ser aprendida pelos modelos de redes baseados em Perceptron disponíveis naquela época. O impacto da publicação deste livro abalou profundamente as pesquisas realizadas nesta área de estudos. O Madaline (Many Adaline), também criado por Widrow [Widrow 88, 90], podia de certa forma resolver o problema, mas o aprendizado não podia ser realizado de uma forma muito “natural” e automatizada, pois requeria a intervenção humana na construção da rede. Devido as críticas feitas e a falta de uma solução para os problemas apresentados, as redes neurais ficaram “esquecidas” por um certo tempo... Somente na década de 80, surgiram novos modelos que deram um novo impulso as redes neurais. Em 1982 surgia um modelo importante de rede criado por J. Hopfield [Hopfield 82], onde este modelo começou a dar um novo impulso as redes neurais. O modelo que Hopfield criou era baseado em um tipo de rede diferente dos modelos baseados no Perceptron, sendo uma rede com conexões recorrentes e com um comportamento baseado na competição entre os neurônios, onde o aprendizado era não supervisionado. Outros modelos similares ao modelo de Hopfield surgiram pouco depois, onde podemos citar alguns como por exemplo: a máquina de Boltzmann [Hinton 84] e o BAM (Binary Associative Memory) [Kosko 87, 87a]. A década de 80 ficou também marcada profundamente pelo reaparecimento das redes baseadas em Perceptrons. Isto deveu-se ao desenvolvimento dos computadores, que eram mais velozes e permitiam realizar melhores simulações das redes neurais, bem como o desenvolvimento de modelos matemáticos que permitiram a solução do problema apontado por Minsky e Papert. Também podemos associar em parte este renascimento das redes neurais ao suposto desencanto com a I.A. clássica. O modelo que permitiu o ressurgimento das redes baseadas em Perceptrons foi o das redes multi-nível, onde o novo algoritmo de aprendizado chamado Back-Propagation resolveu em grande parte os problemas de aprendizado existentes até então. Este modelo foi desenvolvido por diferentes pesquisadores quase ao mesmo tempo, como D. Parker [Parker 82] e D. Rumelhart [Rumelhart 85], mas foi Rumelhart e Hinton que o tornaram este algoritmo famoso com a sua obra “Parallel Distributed Processing - PDP” [Rumelhart 86]. Este algoritmo, o Back-Propagation permitia realizar o aprendizado por correção de erros em uma rede com múltiplas camadas (níveis) e consequentemente resolveria o problema do XOR.

Além dos modelos de Hopfield e do modelo de redes multi-nível com Back-Propagation (chamado de Multi-Layer Perceptron – MLP), outro modelo importante que surgiu nesta década foi o modelo de Teuvo Kohonen [Kohonen 82, 87]. O modelo de Kohonen é muito interessante pois permite o aprendizado competitivo com uma auto-organização da rede neural, criando os chamados “mapas de atributos auto-organizáveis” (self-organizing feature maps). Por fim, o último modelo de destaque neste período, foi o modelo ART (Adaptive Resonance Theory) criado por Gail Carpenter e Stephen Grossberg [Carpenter 93]. Este modelo possui um aprendizado do tipo não supervisionado, criando protótipos (clusters) dos padrões aprendidos. O modelo ART teve diversas versões posteriores, entre elas versões do tipo semi-supervisionado e com uso de conceitos da lógica nebulosa (Fuzzy-ART). Os estudos sobre as redes neurais sofreram uma grande revolução a partir dos anos 80, conforme foi demonstrado acima. E, a partir dos anos 80, cada vez mais, esta área de estudos tem se destacado, seja pelas promissoras características

apresentadas pelos modelos de redes neurais propostos, ou seja pelas condições tecnológicas atuais de implementação que permitem desenvolver arrojadas implementações de arquiteturas neurais paralelas em hardwares dedicados, obtendo assim ótimas performances destes sistemas (bastante superiores aos sistemas convencionais).

## 2.3. Modelos Conexionistas

**2.3.1. Definição de uma Rede Neural Artificial** As redes conexionistas são formadas por um conjunto de unidades elementares de processamento de informações fortemente conectadas, que denominamos de neurônios artificiais. Uma RNA é constituída por um grafo orientado e ponderado. Os nós deste grafo são autômatos simples, os chamados neurônios artificiais, que formam através de suas conexões um autômato mais complexo, a rede neural, também conhecida como rede conexionista. Cada unidade da rede é dotada de um estado interno, que nós vamos denominar de estado de ativação. As unidades podem propagar seu estado de ativação para as outras unidades do grafo, passando pelos arcos ponderados, que nós chamamos de conexões, ligações sinápticas ou simplesmente de pesos sinápticos. A regra que determina a ativação de um neurônio em função da influência vinda de suas entradas, ponderadas pelos seus respectivos pesos, se chama regra de ativação ou função de ativação. As mudanças realizadas nos valores dos pesos sinápticos ou na estrutura de interconexão das unidades de uma rede, são responsáveis pelas alterações no comportamento de ativação desta rede. Estas alterações nas conexões e na estrutura da rede é o que nos permite realizar o aprendizado de um novo comportamento. Desta maneira vamos poder modificar o estado de ativação na saída da rede em resposta a uma certa configuração de entradas. Portanto, a rede é capaz de estabelecer associações de entrada-saída (estímulo e resposta) a fim de se adaptar a uma situação proposta. No caso de uma rede com aprendizado supervisionado (vide item sobre tipos de aprendizado), a rede deve adaptar os seus pesos de maneira à passar a responder de acordo com o exemplo dado, ou seja, gerando na sua saída um estado de ativação compatível para com o esperado. O método utilizado para modificar o comportamento de uma rede é denominado de regra de aprendizado.

A grande quantidade de modelos de redes conexionistas existentes torna difícil para nós a descrição exaustiva de todos eles. Se o leitor assim desejar, poderá se aprofundar em maiores detalhes sobre os diferentes modelos de RNAs em obras como o “Handbook of Neural Computation” [Fiesler 97]. Nós iremos nos concentrar aqui em diferenciar estes modelos, tomando como base as suas principais características.

**2.3.2. Classificação e Propriedades** A grande quantidade de modelos existentes nos leva à uma análise de suas principais propriedades e diferenças em detrimento de uma análise caso à caso mais detalhada. Este estudo das principais propriedades das redes neurais nos permite compreender melhor as vantagens e/ou inconvenientes da escolha de um modelo em detrimento de um outro. Consideramos que não existe apenas uma maneira de classificar todos os modelos, mas de um modo geral devem ser considerados grupos de atributos, tais como: tipo de aprendizado, arquitetura de interconexões, forma interna de representação das informações, tipo de aplicação da rede, etc. Caso o leitor tenha o interesse de buscar uma proposta mais formal de classificação das redes neurais, esta pode ser encontrada em [Fiesler 97].

**2.3.2.1. Aprendizado Conexionista** O aprendizado conexionista é em geral um processo gradual e iterado, onde os pesos são modificados várias vezes, pouco à pouco, seguindo-se uma regra de aprendizado

que estabelece a forma como estes pesos são alterados. O aprendizado é realizado utilizando-se um conjunto de dados de aprendizado disponível (base de exemplos). Cada iteração deste processo gradativo de adaptação dos pesos de uma rede neural, sendo feita uma apresentação completa do conjunto de dados, é chamada de época de aprendizado. Os métodos de aprendizado neural podem ser divididos em três grandes classes, segundo o grau de controle dado ao usuário

Aprendizado supervisionado: o usuário dispõe de um comportamento de referência preciso que ele deseja ensinar a rede. Sendo assim, a rede deve ser capaz de medir a diferença entre seu comportamento atual e o comportamento de referência, e então corrigir os pesos de maneira a reduzir este erro (desvio de comportamento em relação aos exemplos de referência). O aprendizado supervisionado utiliza conhecimentos empíricos, habitualmente representados por um conjunto de exemplos etiquetados, ou seja, exemplos com pares de dados de entrada com a respectiva saída associada. A tabela verdade de uma operação booleana do tipo AND poderia ser considerada como um conjunto de exemplo de aprendizado, pois indica os valores de entrada e também a saída desejada. Nos casos de problemas de classificação, a saída é a classe à qual cada exemplo está associado. Exemplo de aplicação: reconhecimento de caracteres em uma aplicação do tipo OCR (Optical Character Recognition) [Osório 91]. Aprendizado semi-supervisionado: o usuário possui apenas indicações imprecisas (por exemplo: sucesso/insucesso da rede) sobre o comportamento final desejado. As técnicas de aprendizado semi-supervisionado são chamadas também de aprendizado por reforço (reinforcement learning) [Sutton 98]. Para ser mais exato, neste tipo de aprendizado nós dispomos apenas de uma avaliação qualitativa do comportamento do sistema, sem no entanto poder medir quantitativamente o erro (desvio do comportamento em relação ao comportamento de referência desejado). Exemplo: aplicações em robótica autônoma, onde supondo uma situação hipotética, sabemos que seguir em frente não é possível pois existe um obstáculo, mas em compensação não temos uma medida numérica que indique para que lado seguir e exatamente como devemos proceder para desviar deste obstáculo.

Aprendizado não-supervisionado: os pesos da rede são modificados em função de critérios internos, tais como, por exemplo, a repetição de padrões de ativação em paralelo de vários neurônios. O comportamento resultante deste tipo de aprendizado é usualmente comparado com técnicas de análise de dados empregadas na estatística (e.g. clustering). Exemplo: diferenciar tomates de laranjas, sem no entanto ter os exemplos com a sua respectiva classe etiquetada (e.g. self-organizing feature maps [Kohonen 87]).

O aprendizado conexionista em geral precisa de uma grande quantidade de dados, que nós agrupamos em uma base de aprendizado. Exemplos de bases de aprendizado podem ser encontrados na Internet no UCI-ML repository [UCI 99]. De acordo com a técnica de aprendizado utilizada, outros conjuntos de dados podem também ser necessários, principalmente para que se possa medir a validade do aprendizado realizado pela rede (e.g. cross-validation [Krogh 95]). Este conjunto de dados complementar é usualmente chamado de conjunto de teste de generalização. A figura 2.4 apresenta um gráfico típico da evolução do erro durante o aprendizado de uma rede neural, comparando a curva do erro (aprendizado supervisionado) referente à base de aprendizado com a curva do erro da base de teste de

generalização. Nós chamamos de generalização a capacidade de um modelo de aprendizado responder corretamente aos exemplos que lhe são apresentados, sendo que estes exemplos NÃO devem estar presentes na base de aprendizado. Um modelo que tem uma boa generalização é aquele modelo que responde corretamente aos exemplos contidos na base de aprendizado, mas também a outros exemplos diferentes daqueles da base de aprendizado, e que estão contidos em uma base de teste. A capacidade de generalizar é a principal capacidade buscada nas tarefas que envolvem aprendizado.

Erro na Saída da Rede

Figura 2.4. Aprendizado: Erro em relação ao conjunto de aprendizado e de teste. Uma rede pode se especializar demasiadamente em relação aos exemplos contidos na base de aprendizado. Este tipo de comportamento vai nos levar a um problema de aprendizado conhecido como super-aprendizado (over-training / over-fitting). Normalmente o over-fitting pode ser detectado/evitado através do uso de um teste de generalização por validação cruzada (cross-validation). O aprendizado de um conjunto de dados pode ser realizado de diferentes formas, se considerarmos a maneira pela qual a rede é alimentada por estes dados:

Aprendizado instantâneo: o conjunto de dados de aprendizado é analisado uma única vez e com isto o conjunto de pesos da rede é determinado de maneira imediata em uma única passagem da base de exemplos. Este modo de aprendizado também é conhecido como: one single epoch learning / one shot learning.

Aprendizado por pacotes: o conjunto de dados de aprendizado é apresentado à rede várias vezes, de modo que possamos otimizar a resposta da rede, reduzindo os erros da rede e minimizando o erro obtido na saída desta. Este modo de aprendizado é caracterizado por trabalhar com uma alteração dos pesos para cada época, ou seja, para cada passagem completa de todos os exemplos base de aprendizado. O algoritmo de aprendizado deve reduzir pouco à pouco o erro de saída, o que é feito ao final de cada passagem (análise) da base de exemplos de aprendizado. Neste tipo de processo, podemos apresentar os exemplos na ordem em que se encontram, ou de modo mais usual, apresentar os dados em uma ordem aleatória. Outros tipos de seleção de exemplos para análise pelo algoritmo de aprendizado nos levam a métodos como a aprendizagem ativa (vide mais abaixo). Este método é conhecido pelo nome de batch-learning e constitui-se de um dos métodos mais utilizados. Aprendizado contínuo: o algoritmo de aprendizado leva em consideração continuamente os exemplos que lhe são repassados. Se o conjunto de dados é bem delimitado, chamamos este método de aprendizado on-line, e caso o conjunto de dados possa ir aumentando (sendo adicionados novos exemplos no decorrer do tempo), então chamamos este método de aprendizado incremental. O aprendizado on-line se opõe ao aprendizado por pacotes, pois ao contrário deste, para cada novo exemplo analisado já se realiza uma adaptação dos pesos da rede, com o objetivo de convergir na direção da solução do problema. O aprendizado contínuo incremental deve ser analisado sob o ponto de vista da aquisição dos dados (adição de novos exemplos na base de aprendizado), onde devemos prestar atenção para não confundir este tipo de aprendizado com o aprendizado incremental em relação a estrutura da rede (adição de novos neurônios no decorrer da simulação). O principal problema do aprendizado contínuo é a dificuldade de achar um bom compromisso entre a plasticidade e a estabilidade da rede. Uma rede com uma grande facilidade de adaptação pode “esquecer” rapidamente os conhecimentos anteriormente adquiridos e uma rede com uma grande estabilidade

pode ser incapaz de incorporar novos conhecimentos. Aprendizado ativo: este modo de aprendizado assume que o algoritmo de adaptação da rede pode passar de uma posição passiva (apenas recebendo os dados do jeito como lhe são passados), para uma posição ativa. Sendo assim, assumimos que este algoritmo poderá vir a intervir sobre a forma como os dados lhe são repassados. Neste caso, a rede pode intervir e determinar assim quais dados que serão considerados e/ou desconsiderados, além também de determinar a ordem em que estes dados deverão ser considerados. A rede pode também vir a solicitar novos dados que julgue necessários para o bom aprendizado do problema proposto. Esta é uma área que vem sendo investigada com mais destaque recentemente.

A adaptação/otimização dos pesos também pode ser implementada por diferentes métodos, segundo o tipo de regra de aprendizado que for empregado. As regras de aprendizado mais usadas são [Jodoin 94, Caudill 92, Simpson 90, Faq 99]:

Maiores detalhes sobre a implementação de algoritmos de aprendizado neural podem ser encontradas nas seguintes obras [Osório 91, 92, 98] e na Internet em <http://www.inf.unisinos.br/~osorio/neural.html>

Métodos de correção do erro, tais como a descida de uma superfície de erro baseada no gradiente. Exemplos de modelos deste tipo: Adaline, Perceptron, Back-Propagation, CascadeCorrelation; Métodos de aprendizado por reforço. Exemplos: Driver-Reinforcement Learning, AHC; Métodos de aprendizado por competição ou por auto-organização. Exemplos: Kohonen SelfOrganizing Feature Maps, ART1; Métodos de aprendizado através da criação de protótipos ou clusters. Exemplos: RBF, ART1, ARN2; Métodos de aprendizado baseados em memórias associativas (auto-associativas ou heteroassociativas). Exemplos: Modelo de Hopfield, BAM. Métodos de aprendizado de seqüências temporais (redes recorrentes). Exemplos: SRN, BPTT, RTRL.

Existem alguns métodos que podem pertencer a duas categorias ao mesmo tempo, por exemplo, as redes com aprendizado do tipo ARN2 [Giacometti 95] que inclui neste modelo técnicas de aprendizado não-supervisionado, aprendizado supervisionado, adaptação por competição, e também através do uso de um método de criação de protótipos. Nas seções seguintes, nós vamos focar com mais atenção os modelos baseados no Perceptron: que possuem aprendizado supervisionado com descida do gradiente.

**2.3.2.2. Tipos de Unidades** As unidades de uma rede – os neurônios artificiais – podem ser de diferentes tipos, de acordo com a função interna utilizada para calcular o seu estado de ativação. As principais diferenças são relativas ao tipo de função de ativação utilizada (e.g. linear, sigmoide assimétrica (exp), sigmoide simétrica (tanh), gaussiana, etc) [Jodoin 94a, Jodoin 94b]. Outro elemento importante diz respeito a forma como os neurônios armazenam as informações: unidades baseadas em protótipos, unidades do tipo Perceptron. Vamos diferenciar aqui estes dois tipos de maneiras de representar o conhecimento nas unidades de uma rede.

**Redes à base de protótipos:** este tipo de rede utiliza neurônios que servem para representar protótipos dos exemplos aprendidos – as unidades tem uma representação interna que agrupa as características comuns e típicas de um grupo de exemplos [Orsier 95]. As redes baseadas em protótipos tem normalmente um aprendizado não supervisionado (com um ou mais protótipos associados à cada classe). Uma das vantagens deste tipo de redes é a possibilidade de fazer um aprendizado contínuo e incremental, uma vez que não é muito difícil de conceber um



algoritmo capaz de aumentar a rede neural através da adição de novos protótipos. Os protótipos são também denominados de clusters, onde apresentamos um exemplo de rede a base de protótipos na figura 2.5. Este tipo de redes vão gerar uma representação dita localista de conhecimentos.

Redes à base de Perceptrons: as unidades do tipo “Perceptron” foram criadas por Frank Rosenblatt em 1950. Este é um dos modelos de neurônios mais utilizados na atualidade. Ele é a base de diversos tipos de RNA com aprendizado supervisionado utilizando uma adaptação por correção de erros (usualmente baseada na descida da superfície de erro usando o gradiente). O modelo do Perceptron de múltiplas camadas (MLP – Multi-Layer Perceptron) tornou-se muito conhecido e aplicado, sendo na maior parte das vezes associado a regra de aprendizado do Back-Propagation [Jodoin 94, Widrow 90, Rumelhart 86]. A figura 2.6 apresenta um esquema da representação de conhecimentos nas redes baseadas em Perceptrons, e como este tipo de redes é capaz de classificar padrões, gerando planos (ou hiper-planos) de divisão do espaço em que se situam os exemplos.

Reta, Plano ou Hiper-plano de separação das classes Entrada Y

2.3.2.3. Tipos de Arquiteturas de Conexão das Redes As unidades de uma rede neural podem se conectar de diferentes modos, resultando em diferentes arquiteturas de interconexão de neurônios. A figura 2.7 apresenta alguns exemplos de possíveis maneiras de conectar os componentes de uma RNA. As arquiteturas de redes mais importantes são:

Redes com uma única camada: as unidades estão todas em um mesmo nível. Neste tipo de arquitetura, as unidades são conectadas diretamente às entradas externas e estas unidades servem também de saídas finais da rede. As redes de uma única camada possuem normalmente conexões laterais (entre os neurônios de uma mesma camada). Um exemplo deste tipo de arquitetura de redes são as redes do tipo “Self-Organizing Feature Maps” [Kohonen 87]. Redes com camadas uni-direcionais: as unidades são organizadas em vários níveis bem definidos, que são chamados de camadas ou layers. Cada unidade de uma camada recebe suas entradas vindas à partir de uma camada precedente, e envia seus sinais de saídas em direção a camada seguinte. Estas redes são conhecidas como redes feed-forward. A Figura 2.7(a) mostra um exemplo de uma rede de três camadas uni-direcionais. Esta arquitetura de três camadas (entrada, camada oculta e saída) é muito usada em aplicações práticas das redes neurais. O modelo MLP [Widrow 90, Rumelhart 86] é composto em geral de uma arquitetura deste tipo, ou seja, com apenas uma camada oculta (hidden layer), mas nada nos impede de colocar mais de uma camada oculta entre a camada de entrada e a camada de saída de uma rede. Um outro tipo de interconexão utilizado em redes uni-direcionais são os atalhos (shortcuts) que permitem a conexão de uma unidade à outra em uma camada posterior, passando por cima de outras camadas intermediárias. O uso desta técnica vai nos permitir “saltar” por cima de uma camada até uma outra camada (vide figura 2.7(b)), à condição de não introduzir uma recorrência na rede, o que descaracterizaria esta rede como sendo do tipo feed-forward.

Redes recorrentes: as redes recorrentes podem ter uma ou mais camadas, mas a sua particularidade reside no fato de que temos conexões que partem da saída de uma unidade em direção a uma outra unidade da mesma camada ou de uma camada anterior à esta. Este tipo de conexões permitem a criação de modelos que levam em consideração aspectos temporais e comportamentos dinâmicos, onde a saída de uma unidade depende de seu estado em um tempo anterior. Os laços internos ao mesmo tempo que dão características interessantes de memória e

temporalidade as redes, tornam este tipo de redes muito instáveis, o que nos obriga a usar algoritmos específicos (e usualmente mais complexos) para o aprendizado destas redes. Um tipo particular de redes recorrentes são as redes totalmente conectadas, e um exemplo de modelo recorrente de uma única camada e totalmente conectado são as redes de Hopfield, representadas na figura 2.7(d). Redes de ordem superior: as unidades deste tipo de rede permitem a conexão direta entre duas ou mais de suas entradas, antes mesmo de aplicar a função de cálculo da ativação da unidade [Fiesler 94a]. Este tipo de rede serve para modelar “sinapses de modulação”, ou seja, quando uma entrada pode modular (agir sobre) o sinal que vem de uma outra entrada. Um modelo particular de rede de ordem superior são as redes tipo Sigma-Pi que foram apresentadas no livro PDP – Parallel Distributed Processing [Rumelhart 86], e que são representadas na figura 2.7(e).

A arquitetura de uma rede também pode ser classificada de acordo com a evolução desta no decorrer de sua utilização e desenvolvimento do aprendizado. Em função deste critério podemos ter os seguintes grupos:

**Redes com estrutura estática:** a rede tem a sua estrutura definida antes do início do aprendizado. A quantidade de neurônios, assim como a sua estrutura de interconexões, não sofrem alterações durante a adaptação da rede. As únicas mudanças se realizam à nível dos pesos sinápticos, que são modificados durante o processo de aprendizado. Este tipo de modelo impõe uma dificuldade maior ao usuário: a determinação do número ideal de neurônios e de conexões a ser utilizado em uma determinada aplicação. Uma rede com poucas unidades e conexões tem forte chance de não ter sucesso em uma tarefa de aprendizado, não tendo condições de alcançar o melhor desempenho possível por falta de capacidade de representação de todos os conhecimentos envolvidos no problema tratado. Uma rede com muitas unidades pode ter também problemas de convergência e principalmente de generalização, pois quando se tem muita capacidade de armazenamento de informações em uma rede, esta tem uma tendência a decorar os exemplos no lugar de “aprendê-los” (generalizar os conhecimentos sobre o problema) [Fiesler 97, Krogh 95]. No caso deste tipo específico de redes, não existe um método formal que permita determinar o número exato e ótimo de unidades e conexões à serem empregadas no aprendizado de um determinado problema. As redes do tipo MLP com BackPropagation, de acordo com o modelo proposto por Rumelhart, são redes do tipo estático.

**Redes com estrutura dinâmica:** as redes que possuem uma estrutura dinâmica são redes onde o número de unidades e conexões pode variar no decorrer do tempo. Estas redes são também chamadas de ontogênicas [Fiesler 94b]. As modificações na estrutura da rede podem ser do tipo generativo (incremental) ou do tipo destrutivo (reduzidor por eliminação/simplificação). A escolha entre estes dois tipos de métodos é bastante polêmica: devemos começar com uma rede pequena e ir aumentando ela, ou devemos começar com uma rede bastante grande e ir reduzindo o seu tamanho posteriormente? Alguns autores defendem a idéia de uma criação construtiva de conhecimentos [Elman 93, Osório 98]. Do ponto de vista relacionado à carga de processamento de dados necessária para as simulações neurais, a opção por uma rede pequena que adiciona pouco à pouco novas unidades e conexões é sem dúvida a de melhor performance, pois nas redes do tipo destrutivo uma grande parte do esforço de aprendizado acaba sendo depois destruído ao ser realizada a simplificação da rede. Apesar desta discussão, sobre qual dos dois tipos de redes com estrutura dinâmica que seria melhor usar, não possuir um consenso, podemos dizer que uma grande parte dos pesquisadores concorda que as redes ontogênicas em geral são um dos melhores métodos que

existem para se escolher uma boa arquitetura para uma rede neural e assim resolver melhor um certo problema proposto. As redes do tipo CascadeCorrelation (CasCor [Fahlman 90]) são redes do tipo dinâmico e incremental. O último ponto relevante que vamos abordar em relação a arquitetura das redes neurais está relacionado à modularidade [Ronco 96, Ronco 95, Amy 96, Rouzier 98, Jacobs 91]. As redes neurais podem trabalhar com arquiteturas modulares: elas podem ser constituídas por blocos com uma maior ou menor dependência entre eles. Existem diferentes maneiras de integrar e fazer cooperar os diferentes módulos de uma rede neural. Um primeiro método consiste em decompor o problema e obter assim módulos especializados para cada sub-problema. Um exemplo de aplicação deste tipo de método é o caso das aplicações de classificação em múltiplas classes, onde o problema de identificação de cada classe pode ser tratado por módulos separados, e então no lugar de ter um único classificador para os  $N$  exemplos em  $M$  classes, temos um classificador para cada uma das  $M$  classes. Outro tipo de método usado pelas redes modulares, mas mais complexo de ser implementado, é aquele onde os diferentes módulos vão tentar cooperar entre si a fim de juntos resolverem um problema. Neste tipo de método não são impostas tarefas particulares à módulos pré-especificados, deixando para a rede a tarefa de distribuir os conhecimentos e gerenciar a interação entre os módulos. A modularidade é um problema relativo à escolha de uma arquitetura de rede, mas ela também pode ser ligada ao problema de particionamento dos dados de aprendizado (em um esquema semelhante ao usado na aprendizagem ativa, onde cada módulo poderia escolher que informações iria tratar). Para concluir, devemos salientar que a modularidade pode se tornar um aspecto muito importante a ser considerado segundo o tipo e a complexidade do problema a ser tratado.

#### 2.3.2.4. Tipos de Aplicações das Redes Neurais

As RNA podem ser aplicadas à diferentes tipos de tarefas, tais como: o reconhecimento de padrões (e.g. reconhecimento de faces humanas), a classificação (e.g. reconhecimento de caracteres OCR), a transformação de dados (e.g. compressão de informações), a predição (e.g. previsão de séries temporais, como as cotações da bolsa de valores, ou também, uso para o diagnóstico médico), o controle de processos e a aproximação de funções (e.g. aplicações na área da robótica). Um grande número de exemplos de aplicações pode ser encontrado no UCI-ML repository [UCI 99]. Todas estas tarefas podem ser reagrupadas em dois grupos principais, segundo o tipo de saída fornecido pela rede neural e o comportamento que é buscado. Estes dois grupos são:

**Redes para a aproximação de funções:** este tipo de redes devem ter uma saída com valores contínuos e usualmente são empregadas para realizar aproximações de funções (interpolação) [Chentouf 97]. Neste tipo de aplicações, as funções são representada por um conjunto de pontos-exemplo desta. Este tipo de redes é capaz de aprender uma função de transformação (ou de associação) de valores de entrada em valores de saída, usualmente estimando por interpolação as respostas para os casos que não aparecem na base de exemplos. Este tipo de problemas de aprendizado neural de funções é conhecido por ser uma aplicação de um problema de regressão [Bishop 97]. Em geral as funções a serem aprendidas pelas redes possuem tanto as entradas como as saídas indicadas através de valores contínuos (variáveis não discretas).

**Redes para a classificação de padrões:** este tipo de rede deve atribuir uma classe para cada exemplo que lhe é fornecido. Portanto, a saída da rede é a classe associada ao exemplo e por consequência, as classes são valores discretos e não contínuos. A classificação é um caso particular da aproximação de funções onde o

valor de saída da rede é discretizado e pertence a um conjunto finito de classes. No caso do aprendizado supervisionado, o conjunto de classes é bem definido e conhecido antes de ser iniciado o processo de aprendizado. Uma rede utilizada para fins de classificação deve possuir saídas discretas, ou então, deve implementar métodos de discretização de suas saídas (e.g. aplicação de um limiar de discriminação – activation threshold). As entradas da rede podem ser tanto contínuas, como também podem ser discretas, o que não deve interferir no fato desta rede ser usada para uma aplicação classificação.

Seria muita pretensão de nossa parte se tentássemos classificar todos os diferentes modelos de redes neurais em apenas uma destas duas classes descritas acima. A maioria dos modelos pode ser adaptado para ser utilizado em um ou em outro tipo de aplicação, entretanto, alguns modelos são claramente mais adaptados a um tipo de tarefa que ao outro, como é o caso do Cascade-Correlation que foi desenvolvido basicamente apenas para tarefas de classificação.

#### Discussão sobre as Redes Conexionistas

As redes conexionistas utilizam métodos de aprendizado à partir de exemplos que possibilitam o ajuste dos pesos de suas conexões, resultando em um comportamento próximo ou até mesmo exatamente igual ao esperado. Esta modificação dos pesos da rede é feita de forma que a rede generalize os conhecimentos contidos na base de exemplos de aprendizado. Uma boa definição das redes conexionistas é dada por Giacommetti [Giacometti 92]: as redes são capazes de aprender e representar o “saber fazer algo” (savoir-faire), que se traduz pelo seu comportamento após o processo de aprendizado de uma tarefa; este “saber fazer”, representado pelos conhecimentos práticos (empirical knowledge ~ practical examples) adquiridos pela rede, aparece aqui em oposição ao “saber sobre algo” (savoir-que) que representa os conhecimentos teóricos sobre um determinado assunto (theoretical knowledge ~ symbolic rules). Esta distinção entre o saber fazer uma tarefa e o conhecimento sobre a tarefa, é um dos pontos mais importantes da discussão sobre os sistemas híbridos, pois ambos os conhecimentos se completam um ao outro. Antes de prosseguir em nossa análise sobre as redes conexionistas, cabe ressaltar que não seria possível fazer aqui uma análise relativa a todos os modelos de redes neurais, por isso nosso estudo se concentra principalmente em dois tipos principais de redes: as redes à base de protótipos com aprendizado supervisionado e as redes à base de Perceptrons (MLP) com aprendizado supervisionado. Nosso objetivo é fazer uma discussão orientada principalmente para estes dois tipos de redes, dada a sua importância. Devemos ressaltar também que nosso interesse é voltado às redes usadas para aplicações de classificação, visto que nosso estudo aborda aplicações do tipo sistemas especialistas, onde a aproximação de funções não é uma característica típica deste tipo de sistemas.

As redes conexionistas, em particular aquelas que tem sido aplicadas na construção de sistemas inteligentes, possuem as seguintes vantagens:

**Conhecimento empírico:** o aprendizado à partir de exemplos é feito de uma maneira bastante simples e permite uma aquisição de conhecimentos de forma automática, muitas vezes de maneira bem mais fácil e confiável do que através de outros métodos de aquisição de conhecimentos (ajuda a resolver em parte o problema do “gargalo da aquisição de conhecimentos” bem conhecido dos sistemas especialistas);

Degradação progressiva: as respostas dadas por uma rede se degradam progressivamente na presença de “perturbações e distorções” dos dados de entradas. Em geral, as redes obtêm uma boa generalização dos conhecimentos presentes na base de aprendizado e sendo assim são menos sensíveis a “perturbações” do que os sistemas simbólicos; Manipulação de dados quantitativos: o fato de se trabalhar com uma representação numérica dos conhecimentos implica que as redes são melhor adaptadas para a manipulação de dados quantitativos (valores contínuos). Grande parte dos problemas de nosso mundo real, necessitam do tratamento de informações medidas de forma quantitativa, onde uma representação qualitativa muitas vezes implica na perda de informações. As redes neurais são menos vulneráveis aos dados aproximativos e a presença de dados distorcidos ou incorretos que possam estar presentes na base de aprendizado. Esta capacidade de manipular dados aproximados e até mesmo inexatos é mais difícil de ser encontrada em outros métodos de aprendizado do tipo simbólico; Paralelismo em larga escala: as redes neurais são compostas de um conjunto de unidade de processamento de informações que podem trabalhar em paralelo. Apesar da maioria das implementações de RNAs serem feitas através de simulações em máquinas seqüenciais, é possível de se implementar (softwares e hardwares) que possam explorar esta possibilidade de ativação simultânea das unidades de uma rede. A maior parte das implementações de redes neurais simuladas em máquinas seqüenciais pode ser facilmente adaptada em uma versão paralela deste sistema.

As redes conexionistas apresentam um certo número de inconvenientes, do mesmos modo que os outros tipos de métodos de aprendizado. No caso específico das redes, temos limitações tais como: Arquitetura e parâmetros: não existe um método totalmente automático para que se possa escolher a melhor arquitetura possível para um problema qualquer. É bastante difícil de se encontrar uma boa topologia de uma rede, assim como os bons parâmetros de regulação do algoritmo de aprendizado. A evolução do processo de aprendizado é bastante influenciada por estes dois elementos: a arquitetura da rede e os parâmetros de regulação do algoritmo. O sucesso da rede depende bastante de uma boa escolha destes elementos, que variam muito de um problema para outro. Uma simples troca do conjunto de exemplos de uma base de aprendizado pode nos obrigar a reconfigurar toda a rede; Inicialização e codificação: os algoritmos de aprendizado conexionista são em geral muito dependentes do estado inicial da rede (devido a inicialização aleatória dos pesos) e da codificação dos dados da base de aprendizado. Uma má escolha dos pesos iniciais da rede, do método de codificação dos dados de entrada, ou mesmo, a ordem de apresentação destes dados, pode levar ao bloqueio do processo de aprendizado (e seu conseqüente fracasso), ou então, pode dificultar bastante o processo de convergência da rede na direção de uma boa solução;

Caixa preta: os conhecimentos adquiridos por uma rede estão codificados no conjunto de valores dos pesos sinápticos, assim como pela maneira pela qual estas unidades se conectam. É extremamente difícil para um ser humano conseguir interpretar diretamente estes conhecimentos. As redes conexionistas são “caixas pretas” onde os conhecimentos ficam codificados de tal forma que estes são ininteligíveis para o utilizador ou até mesmo para um especialista. Uma rede não possui a capacidade de explicitar o tipo de raciocínio que lhe levou a obter uma certa resposta, ao contrário dos sistemas baseados em regras, que por sua vez podem facilmente mostrar a seqüência de regras aplicadas na resolução de um problema; Conhecimentos teóricos: as redes neurais clássicas não permitem que se utilize os conhecimentos teóricos que possam estar disponíveis sobre um determinado

problema que estejamos tratando. Como as árvores de decisão, as redes neurais são orientadas para a aquisição de conhecimentos empíricos (baseados em exemplos). Um modo simplista de se aproveitar algum conhecimento teórico pré-existente, consiste em se converter regras em exemplos (“protótipos” representativos destas regras). Entretanto, este tipo de método não nos garante que a rede será capaz de aprender corretamente estes exemplos, sendo assim, não podemos garantir que ao final do aprendizado todos os conhecimentos teóricos disponíveis estarão bem representados internamente na rede.

Estes tópicos listados acima não cobrem exaustivamente todas vantagens e desvantagens das redes conexionistas, mas permitem que se tenha uma idéia das principais características deste tipo de sistemas. Podemos encontrar uma análise complementar a citada acima em outras obras da área, como por exemplo [Orsier 95, Towell 91, Fahlman 88]. No que diz respeito mais especificamente ao aprendizado usando redes neurais baseadas em MLP com Back-Propagation (um dos modelos de redes neurais mais utilizados na atualidade), podemos listar alguns dos pontos inconvenientes deste modelo: Paralisia do aprendizado: as redes do tipo MLP com Back-Propagation, devido à maneira como o algoritmo ajusta os pesos, tem a tendência à não mais corrigir estes pesos uma vez que a saída das unidades da rede forneçam valores próximos à 0 ou à 1 (isto se deve ao uso da aplicação da sigmoide e de sua derivada pelo algoritmo Back-Propagation). Este comportamento do algoritmo de aprendizado permite dar uma maior estabilidade ao processo de adaptação dos pesos, mas em compensação pode paralisar o aprendizado da rede. Este problema também é conhecido como o “flat spot problem”. Instabilidade e esquecimento catastrófico: de maneira inversa ao problema da paralisia, as redes também podem sofrer de um problema de instabilidade com a conseqüente perda dos conhecimentos anteriormente adquiridos. Uma vez que as redes realizam a minimização do erro de uma maneira não coordenada, ou seja, as unidades competem entre si a fim de reduzir o erro, isto pode levar a uma constante concorrência entre as unidades. Não importa se uma unidade se adaptou a fim de realizar uma pequena, mas importante tarefa, esta unidade vai continuar sempre tentando alterar os seus pesos a fim de minimizar o erro global ao máximo possível. Este tipo de “comportamento competitivo” pode nos levar à duas situações: (1) as unidades mudam constantemente de “opinião” durante o aprendizado (conhecido como o moving target problem), ou, (2) a rede perde grande parte dos conhecimentos já adquiridos ao tentarmos aprender um novo conjunto de exemplos de aprendizado (conhecido como o esquecimento catastrófico). Portanto este problema envolve a busca de um ponto de equilíbrio entre a grande plasticidade (capacidade de se adaptar) e a estabilidade (necessidade de manter as informações) das redes neurais. Redes neurais com uma grande plasticidade estão sujeitas a ficarem alterando indefinidamente os seus pesos, ou então, destruir uma boa configuração de pesos ao tentar adquirir novos conhecimentos.

Escolha dos parâmetros do algoritmo de aprendizado e a velocidade de convergência: na maior parte das aplicações, a velocidade de convergência de uma rede em direção à um mínimo (local ou global) de erro é realizada de maneira muito lenta. As alterações dos pesos da rede devem ser feitas pouco à pouco de modo a garantir que não se “ultrapasse” o ponto ótimo de mínimo da curva de erro. O algoritmo de Back-Propagation é dotado de dois parâmetros –  $a$  e  $b$  – que controlam respectivamente a velocidade de aprendizado (learning speed) e a inércia na descida da curva de erro (momentum). Estes dois parâmetros permitem que o processo de adaptação dos pesos da rede seja acelerado ou retardado, mas é

preciso que eles sejam ajustados precisamente para que se obtenha bons resultados. Estes dois parâmetros, que devem ser fornecidos pelo usuário, são essenciais para o bom desempenho do processo de aprendizagem. O problema é que não possuímos métodos precisos de estimar estes valores. Além disso, os valores de  $a$  e  $b$  são bastante dependentes do tipo de aplicação e da base de exemplos de aprendizado utilizada, devendo ser reconfigurados novamente caso o problema tratado seja alterado. Um valor de  $a$  ou  $b$  que não seja muito bem escolhido pode levar ao fracasso toda a tentativa de se aprender uma base de dados. É por isso que normalmente o aprendizado de uma base de exemplos é feito com o uso de  $N$  conjuntos de configurações de parâmetros do algoritmo, para que se possa ter uma melhor chance de encontrar os valores adequados destes parâmetros.

Para concluir sobre as redes MLP, o algoritmo de Back-Propagation não é um algoritmo incremental, nem ao nível da base de exemplos, e muito menos ao nível da estrutura da rede. A arquitetura da rede é estática e consequentemente este continua à ser um problema a mais no que se refere ao aprendizado: como fazer para estimar o número ideal de neurônios para uma dada aplicação? Apesar de todos estes problema, o algoritmo Back-Propagation ainda é um dos métodos mais usados junto as redes neurais. Alguns pesquisadores, conscientes dos problemas deste algoritmo, propuseram técnicas para resolver ou reduzir estes problemas [Schiffmann 93, 94]. Podemos citar aqui alguns destes métodos aperfeiçoados de aprendizado: o RPROP [Riedmiller 93], o QuickProp [Fahlman 88], o Gradiente Conjugado (Scaled Conjugated Gradient) [Moller 90] e o Cascade-Correlation [Fahlman 90], bem como as técnicas ontogênicas de aprendizado [Fiesler 94].

Redes Neurais: A busca de uma solução ótima

As redes neurais possuem algumas limitações e problemas que citamos na seção anterior. Vamos listar aqui alguns pontos que devem ser considerados e discutidos no que se refere as redes neurais e a busca de uma solução ou melhoria do aprendizado neural: Aprendizado incremental: a rede neural deve ser capaz de adquirir novos conhecimentos, sem no entanto destruir os conhecimentos anteriormente adquiridos. A rede neural deve ser também capaz de adequar a sua estrutura aos requisitos do problema, sendo que esta deveria poder aumentar de tamanho (quantidade de neurônios) à medida que fosse aumentando a complexidade do problema tratado; Estimativa da topologia: deve-se buscar métodos que permitam ao usuário criar uma topologia de rede adequada para tratar um determinado problema. O ideal seria dotar as redes neurais de mecanismos que possibilitem que esta ajuste automaticamente a sua estrutura em função do problema tratado;

Estimativa dos parâmetros de aprendizado: um bom algoritmo de aprendizado não deve ser muito dependente de parâmetros externos, e idealmente, a rede deveria poder de maneira automática ajustar todos os seus parâmetros para conseguir um resultado ótimo no aprendizado; Introdução de conhecimentos a priori: as redes neurais devem permitir que conhecimentos a priori sobre o problema possam ser inseridos de maneira a facilitar e adiantar o aprendizado de um determinado problema; Instabilidade e velocidade: um bom algoritmo de aprendizado deve ser o menos instável possível, com ótimas chances de adaptar os pesos da rede e convergir em direção a uma boa solução, minimizando o mais possível o erro na saída da rede. Este algoritmo deve permitir um aprendizado rápido e eficiente; Abrir a “caixa preta”: devemos buscar uma solução para o problema da falta de

mecanismos para analisar os conhecimentos adquiridos pela rede. Devemos ser capazes de representar os conhecimentos adquiridos pela rede em um formato mais compreensível para os seres humanos; Aprendizado ativo: a rede neural deve ser capaz de dar um retorno sobre o processo de aprendizado, indicando quais os exemplos que devem ser tratados com uma maior prioridade, ou até mesmo, indicando a necessidade de mais exemplos de uma categoria específica para que o problema possa ser corretamente tratado; Tratamento de informações temporais e contexto: as redes neurais devem ser capazes de considerar o contexto (possuir memória) e assim poder também tratar informações que evoluem no decorrer do tempo, como por exemplo as séries temporais. As redes recorrentes parecem ser um caminho importante a seguir nesta direção, mas ainda restam problemas a serem resolvidos no que se refere a instabilidade e confiabilidade deste tipo de redes.

Estes itens citados acima seguem sendo pesquisados atualmente, e muitas propostas tem sido apresentadas a fim de solucionar (ou minimizar) os problemas ainda enfrentados pelas redes neurais. Apesar de termos problemas relacionados ao aprendizado neural sem serem completamente solucionados, este tipo de técnica tem adquirido uma importância cada vez maior junto à aplicações que necessitem de uma aquisição automática de conhecimentos. As redes neurais superam em muitos casos os demais métodos automáticos de aquisição de conhecimentos.

#### Conclusão e Perspectivas

Neste trabalho apresentamos uma visão geral sobre os sistemas de I.A. e a necessidade do aprendizado para que um sistema inteligente possa ser considerado como tal. Enfocamos o aprendizado neural como sendo uma forma de aquisição de conhecimentos, que dadas as suas peculiaridades, possui um interesse particular na área de inteligência Artificial.

As principais características consideradas foram: a representação de conhecimentos, o paralelismo inerente as unidades da rede, a sua capacidade de adaptação, entre outros aspectos. As redes neurais também se apresentam como uma alternativa ao processamento simbólico de informações, podendo manipular informações do tipo quantitativo e qualitativo sem maiores problemas. Entretanto as redes neurais possuem ainda alguns pontos fracos a serem estudados, principalmente no que diz respeito a explicitação dos conhecimentos adquiridos e na dificuldade de convergência em relação a uma solução ótima. Procuramos apresentar neste trabalho uma visão bastante ampla do assunto, levantando questionamentos e pontos em aberto para estudos futuros, de forma que o leitor possa ter ao mesmo tempo uma visão global da área, e também uma noção dos temas de pesquisa na atualidade neste domínio. Este trabalho visa ser uma fonte de questionamentos e idéias para novos trabalhos, onde a extensa relação de bibliografias remetem o leitor as demais obras da área que podem complementar os temas aqui abordados. Concluindo, acreditamos que as pesquisas futuras nos levam em direção aos sistemas com múltiplas formas de aquisição e representação de conhecimentos, onde os sistemas híbridos, sistemas multi-agentes e sistemas com múltiplas inteligências são uma tendência. Devemos buscar a integração dos métodos simbólicos com os métodos conexionistas de forma a expandir as potencialidades dos “sistemas inteligentes”, para quem sabe assim poderemos realmente ter sistemas com as características (múltiplas!) relacionadas a inteligência que foram levantadas na primeira seção deste trabalho.



## ANEXO B - SAÍDAS NO PROCESSAMENTO DE PALAVRAS

symbol	category	examples
S @SUBJ> @<SUBJ	subject sujeito subjekt	Ninguém gosta de chuva <u>Retomar</u> o controle foi difícil. A cidade era toda de vidro. Seja quem for. Tem gente morrendo de fome no Brasil. Fugiram do zôo um <u>hipopótamo</u> e um <u>crocodilo</u> .
@SUBJ>>	pre-matrix verb subject	Nunca o vi jogar futebol. As <u>inspeções</u> , recorde-se, <u>começaram</u> em Abril
Od, Oacc @ACC> @<ACC	direct (accusative) object objeto direto (acusativo) direkte (akkusativ) objekt	Liga a luz! Para combater as doenças do inverno, come vitaminos. Não tem onde <u>morar</u> . Sempre come um <u>monte</u> de folhas.
@ACC>>	pre-matrix verb object	ela se deixou <u>levar</u> . ..., o que é quase dispensável a <u>dizer</u>
@ACC>-PASS @<ACC-PASS	passive 'se'- construction	
Oi, Odat @DAT> @<DAT	dative object objeto indireto pronominal indirekte (dativ) objekt	Lhe dou um presente. Preste-me a sua caneta, por favor! Me mostre seu hipopótamo!
Op, Opiv @PIV> @<PIV	prepositional object objeto preposicional preæpositionsobjekt	Não me lembro <u>dele</u> . Falamos <u>sobre</u> a sua proposta. Gostava muito <u>de</u> passear ao longo do rio. Não sabe <u>de</u> nada. Pode contar comigo. Chamamos <u>de</u> objeto preposicional complementos indiretos não substituíveis por pronomes adverbiais.
As @ADVS>/@SA> @<ADVS/@<SA Ao @ADVO>/@OA> @<ADVO/@<OA	argument adverbial complemento adverbial adverbialargument [can be substituted by adverbial pronoun, valency	Durava muito <u>tempo</u> . (As) A vase caiu <u>no</u> chão. (As) Não mora mais aqui. Mora <u>em</u> São Paulo. (As) Voltamos <u>ao</u> nosso assunto. (As) Mandaram-nos <u>para</u> Londres. (Ao) Costuma custar mais de mil <u>coroas</u> . (As)

	bound, unlike adjuncts]	
Cs @SC> @<SC	subject complement predicativo do sujeito subjektsprædikativ	Está doente. Está <u>com</u> febre. A moça parece muito <u>cansada</u> . Nadava nua no mar. Andava zangado todo dia.
Co @OC> @<OC	object complement predicativo do objeto objektsprædikativ	O acho muito <u>chato</u> . Tê-lo feito de propósito o faz um <u>delito</u> .
P	predicator predicador prædikator	Hipopótamo come folhas. Hipopótamo tem que dormir muito.
Vm @FMV, @IMV <mv>	main verb verbo principal hovedverb	Bebe muita cerveja. Todo dia mandava (1) o filho comprar (2) leite. Hipopótamo tem que dormir muito.
Vaux @FAUX, @IAUX <aux>	auxiliary verbo auxiliar hjælpeverb	A interface foi feito por uma equipe da WinSoft. Estou lendo um romance português. Hipopótamo tem que dormir muito.
@NPHR	top node noun phrase (especially headlines)	Clinton ainda no poder
@ADVL	top node adverbial (especially headlines)	Seis milhões <u>em</u> dez anos

symbol	category	examples
fA @ADVL> @<ADVL	adjunct adverbial adjunto adverbial adverbialadjunkt	Sempre comiam cedo. As crianças jogavam no parque. Feito o trabalho temos tempo para mais uma cerveja. Entraram na vila quando amanheceu. O outro dia fugiu do zôo um hipopótamo.
@ADVL>AS<	adjunct adverbial in averbal clause	..., ainda que agora de pouco valor
fC	adjunct predicative	Sempre nada nua.

@PRED> @<PRED	adjunto predicativo prædikativadjunkt	Cansado, se retirou.
fApass @<PASS	passive adjunct adjunto do passivo passivadjunkt	Era o herói do dia e foi elogiado pelo chefe do jardim zoológico.
fCsta @S<	statement predicative (sentence apposition) aposto da oração sætningsprædikativ	Morreu o cachorro da velha, o que muito a entristece.
@>S	complementizer adject	Só quando ele lhe surge na frente, se compenetra que era mesmo verdade
fCvoc @VOK	vocative adjunct constituente vocativo vokativadjunkt	Me ajuda, Pedro!
FOC @FOC> @<FOC	focus marker marcador de foco fokusmarker	É a dançar que ela se entende. (focus bracket) Foi de sua música que gostei. (focus bracket) Gosta é de briga.
TOP @TOP	topic constituent constituente de tópico topic-konstituent	A Maria, não quero convidá-la. (object topic) Esse rapaz, ele sabe dançar. (subject topic)

symbol	category	examples
SUB @SUB	subordinator subordinador subordinator	Acho que um jardim zoológico sem hipopótamos não merece subsídios.
SUBcom @COM	comparative subordinator subordinador comparativo komparator	Esta fofqueira fala como uma cachoeira.
SUBprd @PRD	predicative subordinator (role complementizer) subordinador predicativo rolleindleder	Trabalha como guia.

SUBaux @PRT-AUX<	auxiliary subordinator subordinador auxiliar (partículo auxiliar) auxiliarpartikel	Voltou a molestá-la no escritório. O outro ano acabou de ensinar inglês. Hipopótamo tem que dormir muito.
Oaux (special cases) @ICL-AUX< (also @#)	auxiliary complement complemento auxiliar auxiliarkomplement	Hipopótamo tem que dormir muito.
SUB< @AS<	[averbal] clause body tronco de oração [averbal] sætningsstamme (indlederkomplement)	Quando em Roma, faça como os romanos.
CO @CO	coordinator coordenador koordinator	Fugiram do zôo um hipopótamo e um crocodilo.
CJT	conjunct (elemento) conjunto konjunkt	Fugiram do zôo um hipopótamo e um crocodilo.

symbol	category	examples
H - D	head <-> dependent núcleo <-> dependente hoved <-> dependent	uma grande árvore sem dinheiro devagar demais
D DN DNmod DNarg @>N, @N<	adnominal adjunct adjeto adnominal adnominaladjekt (H: noun or pronoun)	o (1) seu (2) grande (3) carro novo (4) (modifiers) a (1) mulher do amigo (2) (modifiers) um tanto (modifier) cacique Jerônimo (modifier) Manoel Neto (1) da Silva (2) (modifiers) a proposta de lhe ajudar (argument) combinaram a venda da casa. (argument) predisposição para diabetes (argument)
DNapp @APP	(adnominal) apposition aposição (do substantivo) [epíteto de identidade] (nominal-) apposition	O grande cacique, Jerônimo, conhecia o seu país como mais ninguém.

DNc @N<PRED	predicative adjunct adjeto predicativo [epíteto predicativo] preædikativadjekt	Jerônimo, um grande cacique, temia ninguém. com a mão na bolsa
DA DAmo DAarg @>A, @A<	adverbial adjunct adjeto adverbial adverbialadjekt (H: adjective, adverb or determiner)	muito devagar (modifier) devagar demais (modifier) rico em ouro (argument) receoso de lhe ter ofendido (argument)
@ADVL>A @A<ADVL @A<ADV @A<PASS @A<PIV @A<SC	functions in postnominal participial "clause"	o número de famílias já (ADVL>A) abrangidas pelo projecto (A<PASS) um equipamento periférico denominado "Audioman" (A<SC) um período destinado a testar a aplicação (A<PIV) uma discoteca situada em Albufeira (A<ADV)
@NUM<	numeral chain constituent	de 1 a 9 de Junho passado a criação de 30 mil a 50 mil postos de emprego
DAcom @KOMP<	argument of comparative complemento comparativo komparativkomplement	é mais bonito do que um hipopótamo.
DP DParg DPmod @P<, @>P	argument of preposition argumento de preposição præpositionsargument [styrelse]	sem dinheiro nenhum (argument) quase sem dinheiro (modifier)
Dfoc	focus dependent dependente focalizador fokus dependent	Até o rei gostava da peça. Comeu nem a sobremesa.

symbol	category	examples
UTT STA QUE COM EXC	utterance enunciado ytring	Não faz nada. [statement] Já vás embora? [question] Espera! [command] Pobre de mim! [exclamation]

STA	statement enunciado declarativo udsagn	A terra é redonda. Gosta muito de elefantes. Sua vez. Às sete. Obrigado.
QUE	question enunciado interrogativo spørsmål	Quem quer uma cerveja? Já ligou para o ministério? Quando?
COM	command enunciado imperativo ordre	Pára com isso! Venha pra cá! Fora!
EXC	exclamation enunciado exclamativo udråb	Deus! Que beleza! Quanta gente!

symbol	category	examples
np np propp pronp	noun phrase sintagma nominal nominalsyntagme (H: noun or pronoun)	Era um homem como um forte. (np) A velha avó dormia na rede. (np) Vou fazê-lo eu mesmo. (pronp) O seu nome era Mário Moreno dos Santos. (propp)
ap adjp advp detp	adpositional phrase sintagma adposicional adpositionssyntagme (H: adjective, adverb or determiner)	As árvores no jardim eram muitovelhas. (adjp) Foi um presidente um poucoiconoclasta. (adjp) Nesta saia, parece mais jovem do que as amigas. (adjp) Costuma falar muito devagar. (advp) Ainda hoje vivem de caça e pesca. (advp) Era muito mais vinho do que imaginava. (detp)
vp	verb phrase sintagma verbal verbalsyntagme (H: main verb [semantically] or auxiliary [dependency grammar])	Ele continua mexendo nas tarefas dos outros.  Vem de lhes propor um acordo.  Temos que dar-lhe mais dinheiro.
pp	prepositional phrase sintagma preposicional præpositionssyntagme (H: preposition)	Abriu a janela da sala Gostou do que viu. Pedro da Silva Mudamos para São Paulo.

symbol	category	examples
cl fcl @#FS- ...	finite (sub)clause oração finita finit (led)sætning	Não acredito que seja verdade
icl @#ICL- ...	non-finite (sub)clause oração infinita infinít (led)sætning	Consertar um relógio não pode ser fácil
acl @#AS- ...	averbal (sub)clause oração averbal averbal (led)sætning	Ajudou onde possível
cu	compound unit paratagma paratagme	ver Roma e viver a história era o seu sonho.

symbol	category	examples
n N	noun nome substantiv (nomen)	árvores n(F P) um oitavo n(<num> M S)
prop PROP	proper noun nome próprio proprium (egenavn)	Estados=Unidos prop(M P) Dinamarca prop(F S)
adj ADJ	adjective adjetivo adjektiv	belas adj(F P) terceiros adj(<num> M P)
v V	v-fin VFIN	fizessem v-fin(IMPf 3P SUBJ)
	v-inf INF	fazermos v-inf(1P)
	v-pcp PCP	comprados v-pcp(M P) [attributive] tem comprado v-pcp [verbal]

	v-ger GER	gerund gerúndio gerundium	correndo v-ger
art DET <artd> DET <arti>		article artigo artikel	os membros art(<artd> M P) [definite] uma criança art(<arti> F S) [indefinite]
pron	pron-pers PERS	personal pronoun pronome pessoal personligt pronomen	mim pron-pers(1S PIV) tu pron-pers(2S NOM)
	pron-det DET	determiner pronoun pronome determinativo determinativt pronomen (adjektivisk pronomen)	estas pron-det(<dem> F P) [demonstrative] muita pron-det(<quant> F S) [indefinite] cujos pron-det(<rel> M P) [relative] quantos pron-det(<interr> M P) [interrogative] minhas pron-det(<poss 1P> F P) [possessive]
	pron-indp SPEC	independent pronoun pronome independente independent pronomen (substantivisk pronomen)	isto pron-indp(<dem> M S) [demonstrative] algo, nada pron-indp(<quant> M S) [indefinite] os=quais pron-indp(<rel> M P) [relative] quem pron-indp(<interr> M S) [interrogative]
adv ADV		adverb advérbio adverbium	facilmente, devagar adv [modals] aqui, lá adv [pronominals] muito, imensamente adv [intensifiers] onde, quando, como adv [relatives, interrogatives] não, até, já adv [operators]
num NUM		numeral numeral numeralia	duas num(F P) 17 num (<cif> M/F P)
prp PRP		preposition preposição preposition	contra prp em=vez=de prp
intj IN		interjection interjeição interjektion	oi! in
conj	conj-s KS	subordinating conjunction conjunção subordinativa underordnende konjunktion	que conj-s embora conj-s



conj-c KC	coordinating conjunction conjunção coordenativa sideordnende konjunktion	e conj-c ou conj-c
pu (unused)	punctuation pontuação tegnsetningstegn	,pu [ komma]

### Inflexion potential of inflecting word classes

	gender	number	case	person	tense, mode	mode
	M, F,-	S, P,-	NOM, GEN, ACC, PIV	1, 2, 3,-	PR, IMPF, PS, FUT, IMP	IND, SUBJ
noun	+	+				
proper noun	+	+				
adjective	+	+				
pronoun personal (PERS)	+	+	+	+		
determiner (DET)	+	+				
independent (SPEC)	M*	S*				
verb finite (VFIN)				+	+	+
infinitive (INF)				+		+
past participle (PCP)	+	+				

+ = inflects (word form category), +\* = lexeme category

Tag	Definition	occurs with
<advl>	adverbial function (header of averbal clause/pp)	@AS-... (PRP)
<artd>	definite article	DET
<arti>	indefinite article	DET
<asarg>	argument in averbal clause	@P<

	rewritten as pp	
<aux>, cf. <mv>	auxiliary	V
<card>, cf. <NUM-ord>	cardinal number	NUM
<cjt>, cf. <co-...>	conjunct	any word
<cjt-head>, cf. <co-...>	first conjunct in coordination	any word
	what a co-ordinator co-ordinates	
*<co-acc>, <co-advl>, <co-app>, <co-dat>, <co-fmc>, <co-ger>, <co-inf>, <co-oa>, <co-oc>, <co-pcv>, <co-postad>, <co-postnom>, <co-pred>, <co-prenom>, <co-prparg>, <co-sa>, <co-sc>, <co-subj>, <co-vfin>, <co-vp>	(used internally for tree-generation) @ACC, @ADVL, @APP, @DAT, main clauses, GER, INF, @OA/@ADVO, @OC, PCP-@IMV, @A<, @N<, @PRED, @>N, @P<, @SA/@ADVS, @SC, @SUBJ, VFIN, VFIN	KC
<com>	comparator function (header of averbal clause/pp)	@AS-... (PRP)
<dem>	demonstrative	DET, SPEC
<DERP>	derivation by prefixation	N, ADJ, V, ADV
<DERS>	derivation by suffixation	N, ADJ, V, ADV
<det>	determiner usage/inflection of adverb	ela estava toda nua
<diff>	differentiator	DET (o mesmo, o outro)
*<fmc>	finite main clause (used internally for tree-generation)	VFIN --> @FMV @#FS-STA (statement)
*<foc>	focus marker (redundant --> @FOC)	ADV (eis, eis=que, é=que, foi ... que, é)
<hyfen>	hyphenated word (left-marked)	acreditá-lo --> 'acreditar' V <hyfen> + 'o' PERS
<ident>	identifier	DET (ele mesmo, ele próprio)
<interr>, cf. <rel>	interrogative	DET (quanto, que), SPEC (quem, o=que), ADV (onde, porquê)
*<kc> (not disambiguated)	can be used as "conjunctual adverb"	ADV (pois, entretanto, mais)
<KOMP>	comparative "hook" (for @KOMP<)	DET, ADV (mais, menos, pior, tal, mesmo)
<ks>, cf. <prp>	used like a "subordinating conjunction"	ADV <rel> (como, onde, quando)

<mv>, cf. <aux>	main verb	V
<NUM-ord>, cf. <card>	ordinal number (subclass of adjective)	ADJ (terceiro, quinto)
<n>	used as noun (head in np)	ADJ
<np-close>	close attachment in np	@N<, @N<PRED, @APP
<np-long>	long attachment in np	@N<, @N<PRED, @APP
<obj>, cf. <si>	direct object reflexive	PERS (e.g. "se" = "a si mesmo")
<parkc-1>	first part in paired coordinator	KC @CO
<parkc-2>	second part in paired coordinator	KC @CO
<poss 1S>, <poss 1P>, <poss 2S>, <poss 2P>, <poss 3S/P>	possessive	DET
<pp>	prepositional phrase tagged as polylexical adverb or adjective	ADV, ADJ
<prd>	predicator function (header of averbal clause/pp)	@AS-... (PRP)
<predco>	predicate coordination (i.e. with shared subject)	VFIN
<prop>	other word class used as "proper noun", i.e. with capital initial	N, ADJ, PCP
<prp>, cf. <ks>	used like a "preposition"	ADV <rel> (como, onde, quando)
<quant>	quantifier (indefinite)	DET, SPEC
<quote>, cf. <v-quote>	head word in quoted utterance	mostly VFIN
<reci>, cf. <si>	reciprocal	PERS (se, nos, vos, si)
<refl>, cf. <si>	reflexive	PERS (se, me, te, nos, vos, si)
<rel>, cf. <interr>	relative	DET (cujo), SPEC (quem, que, o=qual), ADV (onde, quando, como)
<sam->	1. part in contracted word	nisto --> em
<-sam>	2. part in contracted word	nisto --> isto
<sub>	subordinator function (header of averbal clause)	@AS-...
<si>, cf. <refl> and <poss>	reflexive usage of 3.person possessive	DET <poss> (seu, sua, seus, suas)
<topkc>	top level coordinator (without	KC (e.g. "Mas

<SUP>	conjunction) superlative	quando ...")
<vpcjt>, cf. <vpheadcjt>	conjunct in vp-level coordination	V (e.g. "deverá poder descolar e <i>aterrar</i> em porta- aviões."
<vpheadcjt>, cf. <vpcjt>	first conjunct in vp-level coordination	V (e.g. "deverá poder <i>descolar</i> e aterrar em porta- aviões."
<v-quote>, cf. <quote>	quoting verb	VFIN

\* internal or redundant secondary tag

## ANEXO C - MACROS NO EXCEL

Sub NormalizarLista()

' Macro responsável por eliminar caracteres indesejados da lista de sintagmas

```
Cells.Replace What:="~*", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~_", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~[", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~]", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~{", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~}", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~?", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~;", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~(", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~)", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~:", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~--", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~-", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

```
Cells.Replace What:="~.", Replacement:="", LookAt:=xlPart, SearchOrder _
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False
```

Cells.Replace What: "~.", Replacement: "", LookAt: xlPart, SearchOrder \_  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:="~0", Replacement:="", LookAt:=xlPart, SearchOrder \_  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:="~1", Replacement:="", LookAt:=xlPart, SearchOrder \_  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:="~2", Replacement:="", LookAt:=xlPart, SearchOrder \_  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:="~3", Replacement:="", LookAt:=xlPart, SearchOrder \_  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:="~4", Replacement:="", LookAt:=xlPart, SearchOrder \_:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:="~5", Replacement:="", LookAt:=xlPart, SearchOrder \_  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:="~6", Replacement:="", LookAt:=xlPart, SearchOrder \_:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:="~7", Replacement:="", LookAt:=xlPart, SearchOrder \_  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:="~8", Replacement:="", LookAt:=xlPart, SearchOrder \_  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:="~9", Replacement:="", LookAt:=xlPart, SearchOrder \_:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:="~'s", Replacement:="", LookAt:=xlPart, SearchOrder \_:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What: "~", Replacement: "", LookAt:=xlPart, SearchOrder \_  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What: "~+", Replacement: "", LookAt:=xlPart, SearchOrder \_  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What: "~/" , Replacement: "" , LookAt: xlPart, SearchOrder \_  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What: "~!", Replacement: "", LookAt:=xlPart, SearchOrder \_:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

Cells.Replace What:=" """, Replacement:="", LookAt:=xlPart, SearchOrder\_:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False

```
Cells.Replace What:="", Replacement:="", LookAt:=xlPart, SearchOrder _  
:=xlByRows, MatchCase:=False, SearchFormat:=False, ReplaceFormat:=False  
End Sub
```

Sub RemoveQualificadorQuantificador()

```

For coluna = 1 To 2
    linha = 1
    While Cells(linha, coluna).Value <> ""
        'Localizando o quantificador
        If Mid(Cells(linha, coluna).Value, 1, 2) = "A " Then Cells(linha, coluna).Value =
Mid(Cells(linha, coluna).Value, 3, 500)
        If Mid(Cells(linha, coluna).Value, 1, 2) = "O " Then Cells(linha, coluna).Value =
Mid(Cells(linha, coluna).Value, 3, 500)
        If Mid(Cells(linha, coluna).Value, 1, 2) = "o " Then Cells(linha, coluna).Value =
Mid(Cells(linha, coluna).Value, 3, 500)
        If Mid(Cells(linha, coluna).Value, 1, 2) = "a " Then Cells(linha, coluna).Value =
Mid(Cells(linha, coluna).Value, 3, 500)

        If Mid(Cells(linha, coluna).Value, 1, 3) = "As " Then Cells(linha, coluna).Value =
Mid(Cells(linha, coluna).Value, 4, 500)
        If Mid(Cells(linha, coluna).Value, 1, 3) = "Os " Then Cells(linha, coluna).Value =
Mid(Cells(linha, coluna).Value, 4, 500)
        If Mid(Cells(linha, coluna).Value, 1, 3) = "as " Then Cells(linha, coluna).Value =
Mid(Cells(linha, coluna).Value, 4, 500)
        If Mid(Cells(linha, coluna).Value, 1, 3) = "os " Then Cells(linha, coluna).Value =
Mid(Cells(linha, coluna).Value, 4, 500)

        If Mid(Cells(linha, coluna).Value, 1, 3) = "Um " Then Cells(linha, coluna).Value
= Mid(Cells(linha, coluna).Value, 4, 500)
        If Mid(Cells(linha, coluna).Value, 1, 3) = "um " Then Cells(linha, coluna).Value =
Mid(Cells(linha, coluna).Value, 4, 500)
        If Mid(Cells(linha, coluna).Value, 1, 4) = "uma " Then Cells(linha, coluna).Value
= Mid(Cells(linha, coluna).Value, 5, 500)
        If Mid(Cells(linha, coluna).Value, 1, 4) = "Uma " Then Cells(linha, coluna).Value
= Mid(Cells(linha, coluna).Value, 5, 500)

        If Mid(Cells(linha, coluna).Value, 1, 5) = "Essa " Then Cells(linha, coluna).Value
= Mid(Cells(linha, coluna).Value, 6, 500)
        If Mid(Cells(linha, coluna).Value, 1, 5) = "Esse " Then Cells(linha, coluna).Value
= Mid(Cells(linha, coluna).Value, 6, 500)
        If Mid(Cells(linha, coluna).Value, 1, 6) = "Essas " Then Cells(linha,
coluna).Value = Mid(Cells(linha, coluna).Value, 7, 500)
        If Mid(Cells(linha, coluna).Value, 1, 5) = "essa " Then Cells(linha, coluna).Value
= Mid(Cells(linha, coluna).Value, 6, 500)
        If Mid(Cells(linha, coluna).Value, 1, 5) = "esse " Then Cells(linha, coluna).Value
= Mid(Cells(linha, coluna).Value, 6, 500)
        If Mid(Cells(linha, coluna).Value, 1, 6) = "essas " Then Cells(linha,
coluna).Value = Mid(Cells(linha, coluna).Value, 7, 500)

```



```
    If Mid(Cells(linha, coluna).Value, 1, 5) = "Esta " Then Cells(linha, coluna).Value  
    = Mid(Cells(linha, coluna).Value, 6, 500)  
    If Mid(Cells(linha, coluna).Value, 1, 5) = "Este " Then Cells(linha, coluna).Value  
    = Mid(Cells(linha, coluna).Value, 6, 500)  
    If Mid(Cells(linha, coluna).Value, 1, 6) = "Estas " Then Cells(linha,  
coluna).Value = Mid(Cells(linha, coluna).Value, 7, 500)  
    If Mid(Cells(linha, coluna).Value, 1, 5) = "esta " Then Cells(linha, coluna).Value  
    = Mid(Cells(linha, coluna).Value, 6, 500)  
    If Mid(Cells(linha, coluna).Value, 1, 5) = "este " Then Cells(linha, coluna).Value  
    = Mid(Cells(linha, coluna).Value, 6, 500)  
    If Mid(Cells(linha, coluna).Value, 1, 6) = "estas " Then Cells(linha,  
coluna).Value = Mid(Cells(linha, coluna).Value, 7, 500)  
  
    linha = linha + 1  
  
Wend  
Next  
  
End Sub
```

## ANEXO D - LISTA DE SINTAGMAS APÓS A EXTRAÇÃO DOS QUANTIFICADORES DO DOCUMENTO 1

adaptação	especialistas	reconhecimento de voz
análise	estes sistemas	redes neurais
analogia	estudos	referência
animais	exemplos	regra
aplicação	experiência	resultado
Aprendizado	experiências	seus objetivos
aprendizagem	ferramentas	similaridade
área	forma	simulação
atributos	fuzzy	sistema
capacidade	indução	sistemas
caso	inferência	sistemas convencionais
casos	influência	sistemas especialistas
CBR	informação	sistemas multiagentes
ciência de a	informações	softwares
computação	Inteligência Artificial	solução
classificação	Kohonen	solução de o problema
comportamento	Lógica	suas entradas
computação	métodos	tarefa
comunicação	modelo	tarefas
conceitos	modelos	técnica
conhecimento	neurônios artificiais	técnicas
conhecimento adquirido	objetivo	tempo
consideração	padrões	teste
contexto	passado	uso
custos	problemas	usuário
dados	processo	valores
decisão	processo de	voz
determinado problema	aprendizagem	Xadrez
diferenças	processos	
entradas	Prolog	

## **ANEXO E - SINTAGMAS OU SENTENÇAS QUE FORAM REMOVIDAS MANUALMENTE**

A figura

A figura N (onde N é o número da figura)

A seção

A seção N (onde N é o número da seção)

Introdução

Resumo

Abstract

Título

Tabela N (Onde N é o número da tabela)

Pg N (Onde N é o número de uma página)

Pg

## ANEXO F - BIBLIOTECA DE COMANDOS DO CLASDOC

Identificação do conceito do sintagma

[http://www.clasdoc.com/clasdoc/proces.php?retorno=conceito&param1=\\*Homem bala](http://www.clasdoc.com/clasdoc/proces.php?retorno=conceito&param1=*Homem bala)

Identificação do pré-qualificador

[http://www.clasdoc.com/clasdoc/proces.php?retorno=pre&param1=\\*Homem bala](http://www.clasdoc.com/clasdoc/proces.php?retorno=pre&param1=*Homem bala)

Identificação do pós-qualificador

[http://www.clasdoc.com/clasdoc/proces.php?retorno=pos&param1=\\*Homem bala](http://www.clasdoc.com/clasdoc/proces.php?retorno=pos&param1=*Homem bala)

Identificação do sinônimo

[http://www.clasdoc.com/clasdoc/proces.php?retorno=sinonimo&param1=\\*Homem](http://www.clasdoc.com/clasdoc/proces.php?retorno=sinonimo&param1=*Homem)

Calculo do *stemming*

[http://www.clasdoc.com/clasdoc/proces.php?retorno=stemming&param1=\\*Homem](http://www.clasdoc.com/clasdoc/proces.php?retorno=stemming&param1=*Homem)

## **ANEXO G - BIBLIOTECA DE COMANDOS DO PALAVRAS**

```
#!/bin/bash
```

```
cat eng/1.txt | /opt/palavras/por.pl --dep> eng/1.dep.txt
```

```
cat eng/1.dep.txt | /caminhodopendrive/extract_np.pl > sintagmast1.txt > eng1.txt
```

## ANEXO H - CÓDIGO FONTE DO PROCESSAMENTO PELO SISTEMA CLASSDOC.COM

```

<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Classdoc.com</title>
</head>
<?php
    set_time_limit(0);

    function my_file_get_contents($site_url){
        $sch = curl_init();
        $timeout = 10;
        curl_setopt ($sch, CURLOPT_URL, $site_url);
        curl_setopt ($sch, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt ($sch, CURLOPT_CONNECTTIMEOUT, $timeout);
        $file_contents = curl_exec($sch);
        curl_close($sch);
        return $file_contents;
    }

    function normaliza($frase) {
        $caracteres = array("_", "[", "]", "{", "}", "?", ";", "(", ")", ":", "--", " - ", " -", " -",
        ", .", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "s", " ", "+", "/", "!", "", "", "Figura",
        "Resumo", "Abstract", "Título", "Seção", "Gráfico", "Tabela", "a", "o", "*", "%");
        $texto = str_ireplace($caracteres,"",$frase);
        $caracteres = array(" ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ");
    );
        $texto = str_ireplace($caracteres," ", $texto);
        $texto = trim($texto);
        return $texto;
    }

    function retiraQuantificador($frase) {
        $palavra = $frase;
        $posicaoEspaco = strpos($palavra, " ");
        $quant = substr($palavra, 0, $posicaoEspaco);
        $quantificadores = array("a", "o", "as", "os", "um", "uma", "uns", "umas",
        "aquele", "aquela", "aqueles", "aquelas", "este", "esta", "essa", "esse", "essas",
        "esses");
        if (in_array(strtolower($quant), $quantificadores))
            $palavra = substr($palavra, $posicaoEspaco + 1,
            strlen($palavra) - $posicaoEspaco);
    }

```

```

        if ($palavra != "")
            return $palavra;
        return $frase;
        break;
    }

    function geraStemming($frase) {
        $vetor = explode(" ", $frase);
        $retorno = "";
        for ($x=0; $x<count($vetor); $x++) {
            if ($vetor[$x] == strtoupper($vetor[$x]) && strlen($vetor[$x]) > 1) {
//Manter as siglas
                $retorno .= strtoupper($vetor[$x]) . " ";
            }
            elseif (strlen($vetor[$x]) > 3) {
                $url
                "http://www.pmsb.com.br:8071/StemmerService.svc/GetStemmer/" . $vetor[$x];
                $stemming
                my_file_get_contents("http://www.google.com.br");
                $retorno .= substr($stemming,1,strlen($stemming)-2) . " ";
                //$retorno .= $vetor[$x] . " ";
            }
        }
        return $retorno;
    }

    function limpa($frase) {
        return retiraQuantificador(normaliza($frase));
    }

    if ($_POST["enviar"] != "") {
        $i = 0;
        foreach ($_FILES["uploads"]["error"] as $key => $error) {
            move_uploaded_file($_FILES["uploads"]["tmp_name"][$i], "textos/"
$_FILES["uploads"]["name"][$i]);
            ++$i;
        }
        echo "<script>alert('Arquivos enviados com sucesso!');</script>";
    }

    if ($_POST["excluir"] != "") {
        $path = "textos/";
        $diretorio = dir($path);

        while($arquivo = $diretorio -> read()){
            if ($arquivo != "." && $arquivo != "..")
                unlink($path.$arquivo);
        }
        $diretorio -> close();
    }

```

```

        echo "<script>alert('Arquivos excluídos com sucesso!');</script>";
    }

    if ($_POST["limpar"] != "") {
        $nomesArquivos = array();

        //Armazena os nomes dos arquivos em um vetor
        $path = "textos/";
        $diretorio = dir($path);
        while($arquivo = $diretorio -> read()){
            if ($arquivo != "." && $arquivo != "..")
                $nomesArquivos[] = $arquivo;
        }
        $diretorio -> close();

        //compara os arquivos entre si
        for ($x=0; $x<count($nomesArquivos); $x++) {
            //abre o arquivo apenas para leitura
            $arquivo = fopen($path.$nomesArquivos[$x],"r");
            $texto = fread($arquivo,20000000);
            $novoTexto = "";
            $vetor1 = explode("\n",$texto); //$vetor1 = explode("\r\n",$texto);

            //limpa o texto
            for($i=0; $i<count($vetor1); $i++) {
                $temp = limpa($vetor1[$i]);
                if ($temp != "" && $temp != "*" && $temp != " " &&
strlen($temp) >= 4)
                    $novoTexto .= limpa($vetor1[$i]) . "\n";
            }

            fclose($arquivo);

            //abre o arquivo novamente apagando o conteúdo
            $arquivo = fopen($path.$nomesArquivos[$x],"w");
            fwrite($arquivo, $novoTexto);
            fclose($arquivo);
        }
    }

    if ($_POST["comparar"] != "") {
        $conteudo = array();
        $matriz = array();
        $nomesArquivos = array();
    }

```





```

        <form          name="form1"          action="compara.php"          method="post"
enctype="multipart/form-data">
    <div style="position:absolute; left:10px">
        <input name="uploads[]" type="file" multiple />
        <input type="submit" value="Enviar" name="enviar" />
    </div>
    <div style="position:absolute; left:500px">
        <div>
            <input type="submit" value="Limpar" name="limpar" />
        </div>
    </div>
    <div style="position:absolute; left:700px">
        <div>
            <input type="submit" value="Comparar" name="comparar" />
        </div>
    </div>
    <div style="position:absolute; left:900px">
        <div>
            <input type="submit" value="Excluir Arquivos" name="excluir" />
        </div>
    </div>
</form>
<div style="position:absolute; left:10px; top:50px;">
    <?php if ($_POST["comparar"] != "") { ?>

        <?php for ($x=0; $x<count($nomesArquivos); $x++) { ?>
            <div style="margin-top:50px;">
                <?php echo "<strong>" . $nomesArquivos[$x] . " :</strong>"; ?>
                <?php for ($y=0; $y<count($conteudo);
$y++) { ?>
                    <?php if (in_array($conteudo[$y],
$matriz[$x])) echo $y + 1 . " :1 "; else echo $y + 1 . " :0 ";?>
                <?php } ?>
            </div>
        <?php } ?>

        <div style="margin-top:50px;">
            <strong>Palavras</strong>
            <ul>
                <?php for ($y=0; $y<count($conteudo); $y++) { ?>
                    <li style="list-style:none"><?php echo $y + 1 . " - " . $conteudo[$y];
?></li>
                <?php } ?>
            </ul>
        </div>
    <?php } ?>
</div>
</div>
</body></html>

```