Escola Politécnica da Universidade de São Paulo

Alice Shimada Bacic

Extração de Informação e Documentação de Laudos Médicos

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção de título de
Doutora em Engenharia.

Área de Concentração: Sistemas Eletrônicos
Orientador: Prof. Dr. Sérgio Shiguemi Furule

São Paulo
2007

Escola Politécnica da Universidade de São Paulo

Alice Shimada Bacic

Extração de Informação e Documentação de Laudos Médicos

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção de título de
Doutora em Engenharia.

Área de Concentração: Sistemas Eletrônicos
Orientador: Prof. Dr. Sérgio Shiguemi Furuie

São Paulo
2007

31200034575

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

São Paulo, 03 de junho de 2007.

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Bacic, Alice Shimada

**Extração de informação e documentação de laudos médicos
/ A.S. Bacic. -- São Paulo, 2007.
p. 165**

**Tese (Doutorado) - Escola Politécnica da Universidade de
São Paulo. Departamento de Engenharia de Telecomunicações e
Controle.**

**1.Sistemas de informação em radiologia 2. Tórax 3. Lingua-
gem natural (Processamento) I. Universidade de São Paulo.
Escola Politécnica. Departamento de Engenharia de
Telecomunicações e Controle II. t.**

**Escola Politécnica da USP
Divisão de Biblioteca**

FT. 2394
Ed. rev.

Dedico este trabalho aos meus avós
Lola, Yoshiharu, Frana e Gregório.

Agradecimentos

Agradeço a todos os amigos e colegas pelo suporte e companheirismo durante o desenvolvimento deste trabalho. Em especial agradeço ao meu orientador Prof. Dr. Sérgio Shiguemi Furuie, a André Castilla, a Carlos da Silva dos Santos, ao Prof. Dr. Roberto Hirata, ao Prof. Dr. João Eduardo Kögler, ao Prof. Dr. Marco Antônio Gutierrez, a Fábio Antero Pires e a todos meus colegas da informática do InCor, Marina Rebelo, Ramon Moreno, Marcelo Santos, Luiz Kobayashi, Alexandre Murakami e Nivaldo Bertozzo.

Agradecimentos especiais a Glauco Corrêa da Cruz Bacic Fratric, pelo trabalho de rotular o *Corpus* de treinamento e corrigir a rotulação do *Corpus* de teste.

Resumo

Os sistemas de informação hospitalares geram diariamente uma quantidade significativa de dados em formato de texto livre, principalmente através de laudos médicos. Os laudos geralmente são recuperados do sistema através de informações associadas como identificação do paciente, por datas ou por profissional responsável. A recuperação da informação a partir do conteúdo descritivo é uma tarefa não trivial, pois os sistemas hospitalares em geral não são capazes de verificar o conteúdo de um texto livre em uma busca.

Não havendo uma estrutura básica de organização, categorização ou indexação do texto livre armazenado nas bases hospitalares, uma grande quantidade de informação deixa de estar disponível para profissionais que necessitam delas, pois não sabem como recuperá-las. A capacidade de recuperação do conhecimento armazenado nestas bases de dados seria de grande valia para pesquisadores, estudantes ou mesmo para o estudo de casos clínicos.

Segundo o contexto descrito, este trabalho propõe a criação de uma ferramenta de documentação automática que tem por objetivo gerar uma formatação associada ao texto livre de laudos em radiologia através da adição de informações obtidas a partir de sistemas de terminologias médicos padronizados. Com este procedimento, pretende-se facilitar a pesquisa pelo conhecimento armazenado em uma base de dados médicos através da informação adicional gerada. Para tanto o trabalho envolve pesquisas nas áreas de Ontologias e Extração de Informação, uma sub-área do Processamento de linguagem Natural.

As ontologias são importantes neste trabalho por tratarem o problema da padronização das terminologias usadas na escrita dos laudos, bem como para fornecer a organização e formatação necessária para que os laudos passem a ser partes de uma base de conhecimento.

A Extração de Informação fornece os algoritmos e técnicas necessárias para que os laudos sejam documentados de forma automática, minimizando a necessidade de intervenção humana, normalmente muito custosa em termos de trabalho manual e tempo.

Como resultado final obteve-se um conjunto de metodologias e ferramentas capazes de receber um laudo em texto livre e gerar um documento XML rotulado com códigos de conceitos definidos em um sistema de terminologias médico, como o UMLS ou o Radlex. Em todas as fases de processamento, até a obtenção do arquivo XML de saída. Obteve-se valores de precisão superiores a 70%, um resultado bastante satisfatório se considerado que os algoritmos de PLN utilizados são todos baseados em regras.

Em adição às ferramentas de PLN desenvolvidas, cita-se como resultados, os trabalhos desenvolvidos para avaliação de ontologias médicas segundo uma área médica predefinido, a organização das ontologias em um formato útil para a utilização por algoritmos de PLN, a criação de um *Corpus* de laudos de Raio-X de Tórax em português para treinamento e testes de aplicações de PLN e um modelo de informação para documentação dos laudos.

Abstract

Hospital Information Systems generate each day a significant amount of data in free text format, mainly as medical reports. Normally the reports are recovered from the system through associated information like patient identification, dates or responsible identification, for example. To recover a report by its content is not a trivial task since hospital systems are not capable of searching the free text content.

Without a basic organizational structure, some categorization and indexing the free text stored on the hospital database is not accessible, since it cannot be recovered in the right context when it is needed. The ability of recovering the knowledge stored on these databases would be valuable for researchers, students or for the study of clinical cases.

According to the described context, this work considers the creation of a tool for automatic documentation of medical reports written in free text. The main objective is to format radiological reports to achieve a more efficient way of recovering the knowledge stored in medical report's databases. To achieve this goal, information from medical terminology systems is added to the original report automatically. Such task requires some research in the field of Ontologies and Information Extraction, a sub field of Natural Language Processing.

Ontologies are important in this work because they provide the standardization needed for the terminologies used in the written reports. It is important too for supplying the organization necessary to format the reports in an adequate way to be stored on the knowledge base.

Information Extraction supplies the algorithms and the necessary techniques to register in an automatic way the radiological reports, minimizing the human intervention, normally with a high cost in terms of handwork and time.

The final result achieved was a set of methodologies and tools used to process a free text report, generating a XML document tagged with codes extracted from a medical terminology system. Considering all process steps, it was achieved a precision of at least 70%, in each step, a good score, if we consider that all the algorithms are rule based.

In addition to the NLP tools results, there are results concerning to medical ontologies evaluation for a pre-defined medical area, the organization need to make the ontologies usable by the NLP tools, the creation of a x-ray Corpus of reports in Portuguese and an information model used to document the reports. The Corpus could be used on the evaluation and test of NLP tools.

Lista de Figuras

FIGURA 1: FLUXO TÍPICO DE PROCESSAMENTO DE EI [JACKSON 2002]	31
FIGURA 2: DIAGRAMA DE CLASSES REPRESENTANDO OS IDENTIFICADORES DO <i>METATHESAURUS</i>	46
FIGURA 3: REPRESENTAÇÃO EM ÁRVORE DA ESTRUTURA DE UMA LAUDO EM FORMATO XML.	70
FIGURA 4: MÓDULOS DO SISTEMA PROPOSTO	73
FIGURA 5: EXEMPLO DE MAPEAMENTO ENTRE RÓTULOS DOS LAUDOS E UMA ONTOLOGIA	75
FIGURA 6: FLUXO DO PROCESSAMENTO DE LAUDOS	75
FIGURA 7: HIERARQUIA DO RADLex VISUALIZADA COM A FERRAMENTA <i>PROTÉGÉ</i>	79
FIGURA 8: ESTRUTURA RELACIONAL DE ARMAZENAMENTO DO RADLex	81
FIGURA 9: PASSOS DA NORMALIZAÇÃO DE UM <i>STRING</i>	83
FIGURA 10: COMPARAÇÃO E BUSCA POR CONCEITOS	87
FIGURA 11: DIAGRAMA ER PARA O ARMAZENAMENTO DOS LAUDOS	165

Lista de Tabelas

TABELA 1: CONSTANTES USADAS EM EXPRESSÕES REGULARES.....	19
TABELA 2: OPERADORES USADOS EM EXPRESSÕES REGULARES.....	19
TABELA 3: OPERADORES REDUNDANTES USADOS EM EXPRESSÕES REGULARES.....	19
TABELA 4: VALORES POSSÍVEIS DOS ITENS DA ÁREA ESQUERDA DE UM SN (ADAPTADO DE [PERINI 2005]).	26
TABELA 5: COMPARAÇÃO ENTRE EI E TU.....	30
TABELA 6: CONCEITOS UTILIZADOS NA DEFINIÇÃO DE ONTOLOGIAS.....	39
TABELA 7: REPRESENTAÇÃO DOS RELACIONAMENTOS ENTRE IDENTIFICADORES NO <i>METATHESAURUS</i> ([UMLS 2004AB]).	46
TABELA 8: VOCABULÁRIOS UTILIZADOS.....	50
TABELA 9: CARACTERÍSTICAS DA BASE DE CONCEITOS.....	51
TABELA 10: CATEGORIAS DE TERMOS DO RADLEX [RADLEX 2003].....	52
TABELA 11: GRUPOS DE TERMOS [RADLEX 2003].....	53
TABELA 12: ATRIBUTOS DOS TERMOS [RADLEX 2003].....	53
TABELA 13: RECOMENDAÇÕES PARA O DESENVOLVIMENTO DE APLICAÇÕES DE PLN [FRIEDMAN 1997].	60
TABELA 14: SUMÁRIO DE CONCEITOS DO RADLEX.....	81
TABELA 15: MAPEAMENTO ENTRE ESTRUTURA DE LAUDOS E RADLEX.....	84
TABELA 16: REGRAS USADAS NO <i>CHUNK PARSER</i> DO NLTK.....	91
TABELA 17: DESCRIÇÃO DO <i>CORPUS</i> ANOTADO.....	98
TABELA 18: DESCRIÇÃO DO <i>CORPUS</i> DE TESTE.....	99
TABELA 19: LISTA DE CONCEITOS UTILIZADOS NOS TESTES 1 E 2.....	106
TABELA 20: LISTA DE CONCEITOS UTILIZADOS NO TESTE 3.....	108
TABELA 21: LISTA DE CONCEITOS UTILIZADOS NO TESTE 4.....	109
TABELA 22: CONSTANTES PARA CÁLCULO DE AVALIAÇÃO DA RI.....	110
TABELA 23: RESULTADOS MÉDIOS PARA A RECUPERAÇÃO, PRECISÃO, <i>FALL-OUT</i> E <i>F-MEASURE</i> PARA O TESTE 1.	111
TABELA 24: RESULTADOS MÉDIOS PARA A RECUPERAÇÃO, PRECISÃO, <i>FALL-OUT</i> E <i>F-MEASURE</i> PARA O TESTE 2.	113
TABELA 25: RESULTADOS MÉDIOS PARA A RECUPERAÇÃO, PRECISÃO, <i>FALL-OUT</i> E <i>F-MEASURE</i> PARA O TESTE 3.	114
TABELA 26: CONCEITOS NÃO RECUPERADOS NO TESTE 3.....	115
TABELA 27: RESULTADOS MÉDIOS DA RECUPERAÇÃO, PRECISÃO, <i>FALL-OUT</i> E <i>F-MEASURE</i> PARA O TESTE 3, CONSIDERANDO APENAS OS ITENS ENCONTRADOS MANUALMENTE.....	116
TABELA 28: RESULTADOS MÉDIOS PARA A RECUPERAÇÃO, PRECISÃO, <i>FALL-OUT</i> E <i>F-MEASURE</i> PARA O TESTE 4.	117
TABELA 29: CONCEITOS NÃO RECUPERADOS NO TESTE 4, PARA PONTUAÇÃO DE CORTE IGUAL A 0,80.....	118
TABELA 30: LEGENDAS (AVALIAÇÃO ATRAVÉS DA CURVA ROC).....	119
TABELA 31: DADOS PARA CURVA ROC, TESTE 1.....	120
TABELA 32: DADOS PARA CURVA ROC, TESTE 2.....	122
TABELA 33: DADOS PARA CURVA ROC, TESTE 3.....	123
TABELA 34: RESULTADO DOS TESTES EXECUTADOS PARA AVALIAÇÃO DO UMLS.....	128
TABELA 35: RESULTADOS DO PRIMEIRO PASSO DE AVALIAÇÃO DO <i>BRILL TAGGER</i>	130
TABELA 36: RESULTADOS DO PRIMEIRO PASSO DE AVALIAÇÃO DO <i>BRILL TAGGER</i>	131
TABELA 37: CRITÉRIOS PARA A AVALIAÇÃO DE SEGMENTOS.....	133
TABELA 38: DESCRIÇÃO DOS ARQUIVOS TESTE E PADRÃO (PARA AS FRASES SORTEADAS).....	133
TABELA 39: DESCRIÇÃO DOS ARQUIVOS TESTE E PADRÃO (COMPLETOS).....	133
TABELA 40: RESULTADOS DA SEGMENTAÇÃO PARA O ARQUIVO TESTE.....	134
TABELA 41: RESULTADOS DA SEGMENTAÇÃO PARA O ARQUIVO PADRÃO.....	135
TABELA 42: TESTE DE HIPÓTESE PARA OS ARQUIVOS TESTE E PADRÃO.....	135
TABELA 43: VALORES OBTIDOS PARA A RECUPERAÇÃO E A PRECISÃO NO ALGORITMO DE SEGMENTAÇÃO.....	136
TABELA 44: CARACTERÍSTICAS GERAIS DA AMOSTRA PARA AVALIAÇÃO DA ROTULAÇÃO DE CONCEITOS.....	138
TABELA 45: VALORES OBTIDOS NA AVALIAÇÃO DO ROTULADOR DE TERMOS.....	138
TABELA 46: PROPORÇÃO SEGMENTOS X TERMOS.....	139
TABELA 47: TAXA DE ACERTO/RECUPERAÇÃO DO ROTULADOR DE CONCEITOS PARA A AMOSTRA.....	139
TABELA 48: TESTE DE HIPÓTESE A AVALIAÇÃO DO ROTULADOR DE CONCEITOS.....	139
TABELA 49: HIERARQUIA DE CHOMSKY.....	159
TABELA 50: TABELAS RELACIONAIS DA REDE SEMÂNTICA [UMLS 2004AB].....	160
TABELA 51: EXEMPLO DE DADOS DA TABELA SRSTR [UMLS 2004AB].....	160
TABELA 52: EXEMPLO DE DADOS DA TABELA SRSTR1 [UMLS 2004AB].....	161
TABELA 53: EXEMPLO DE DADOS DA TABELA SRSTR2 [UMLS 2004AB].....	161
TABELA 54: EXEMPLO DE DADOS DA TABELA SRDEF [UMLS 2004AB].....	161
TABELA 55: RÓTULOS DO <i>BROWN CORPUS</i> USADOS PARA O TREINAMENTO DO <i>BRILL TAGGER</i>	162

Lista de Gráficos

GRÁFICO 1: CURVA PARA A RECUPERAÇÃO X PRECISÃO PARA O TESTE 1	111
GRÁFICO 2: PONTUAÇÃO DE CORTE X PRECISÃO / RECUPERAÇÃO E <i>F-MEASURE</i> PARA O TESTE 1	112
GRÁFICO 3: CURVA PARA A RECUPERAÇÃO X PRECISÃO PARA O TESTE 3	114
GRÁFICO 4: PONTUAÇÃO DE CORTE X PRECISÃO / RECUPERAÇÃO E <i>F-MEASURE</i> PARA O TESTE 3.	114
GRÁFICO 5: CURVA PARA A RECUPERAÇÃO X PRECISÃO PARA O TESTE 4	117
GRÁFICO 6: PONTUAÇÃO DE CORTE X PRECISÃO / RECUPERAÇÃO E <i>F-MEASURE</i> PARA O TESTE 4.	117
GRÁFICO 7: CURVA ROC PARA O TESTE 1	120
GRÁFICO 8: ACURÁCIA X CORTE PARA O TESTE 1	121
GRÁFICO 9: FVN x FFN PARA O TESTE 2	122
GRÁFICO 10: ACURÁCIA X CORTE PARA O TESTE 2	123
GRÁFICO 11: CURVA ROC PARA O TESTE 3	124
GRÁFICO 12: ACURÁCIA X CORTE PARA O TESTE 3	124
GRÁFICO 13: CURVA ROC PARA O TESTE 4	125
GRÁFICO 14: ACURÁCIA X CORTE PARA O TESTE 4	126

Lista de Equações

EQUAÇÃO 1: RECUPERAÇÃO [JURAFSKY 2000].....	32
EQUAÇÃO 2: PRECISÃO [JURAFSKY 2000].....	32
EQUAÇÃO 3: <i>F-MEASURE</i> [JURAFSKY 2000].....	32
EQUAÇÃO 4: <i>FALL-OUT</i> [JURAFSKY 2000].	32
EQUAÇÃO 5: SENSIBILIDADE DE UM TESTE (FVP ou S^+) [RANGAYAN 2005].....	34
EQUAÇÃO 6: ESPECIFICIDADE DE UM TESTE (FVN ou S^-) [RANGAYAN 2005].....	34
EQUAÇÃO 7: PROBABILIDADE DE UM ITEM RELEVANTE NÃO SER ENCONTRADO [RANGAYAN 2005].....	35
EQUAÇÃO 8: PROBABILIDADE DE UM ITEM IRRELEVANTE SER CONSIDERADO RELEVANTE [RANGAYAN 2005]... 35	35
EQUAÇÃO 9: PROBABILIDADE DE UM ITEM IRRELEVANTE SER CONSIDERADO RELEVANTE [RANGAYAN 2005]... 35	35
EQUAÇÃO 10: PPV [RANGAYAN 2005].	35
EQUAÇÃO 11: NPV [RANGAYAN 2005].....	35
EQUAÇÃO 12: $P(A T^+)$ [RANGAYAN 2005].	35
EQUAÇÃO 13: CÁLCULO DA PONTUAÇÃO DE SEMELHANÇA PARA COMPARAÇÃO DE <i>STRINGS</i>	88
EQUAÇÃO 14: CÁLCULO DO ERRO DO <i>BRILL TAGGER</i>	131
EQUAÇÃO 15: CÁLCULO DA ESTIMATIVA DE SEGMENTOS NOS ARQUIVOS DE TESTE DO ALGORITMOS DE SEGMENTAÇÃO	134
EQUAÇÃO 16: CÁLCULO DA PROPORÇÃO DO ERRO NA AMOSTRA	135

Lista de Exemplos

EXEMPLO 1: EXEMPLO DE CONVERSA COM O PROGRAMA ELIZA (EXEMPLO EXTRAÍDO DE [JACKSON 2002])	12
EXEMPLO 2: EXEMPLO DE SENTENÇA TOKENIZADA POR PALAVRAS	21
EXEMPLO 3: SENTENÇA ROTULADA	22
EXEMPLO 4: EXEMPLO DE SENTENÇA PROCESSADA POR UM <i>STEMMER</i>	24
EXEMPLO 5: SINTAGMAS NOMINAIS [PERINI 2005]	26
EXEMPLO 6: CONSTITUINTES DA ÁREA ESQUERDA DE UM SN	26
EXEMPLO 7: PRÉ-DETERMINANTE DA ÁREA ESQUERDA	27
EXEMPLO 8: ÁREA DIREITA DE UM SN (ADAPTADO DE [PERINI 2005])	27
EXEMPLO 9: EXEMPLOS DE SINTAGMAS NOMINAIS	28
EXEMPLO 10: ESTRUTURA DE UMA FRASE DIVIDIDA EM <i>CHUNKS</i> QUE REPRESENTAM SINTAGMAS NOMINAIS.	28
EXEMPLO 11: EXEMPLO DE FORMULÁRIO DE EXTRAÇÃO DE INFORMAÇÃO	31
EXEMPLO 12: TEXTO ORIGINAL DE UM LAUDO	69
EXEMPLO 13: LISTA DE QUESTÕES DO DOMÍNIO PARA O PROBLEMA DE ROTULAÇÃO LAUDOS DE RAIO-X DE TÓRAX.	69
EXEMPLO 14: EXEMPLO DE TRECHO ANOTADO SEMANTICAMENTE	70
EXEMPLO 15: EXEMPLO DE LAUDO	72
EXEMPLO 16: MODELO DO DOCUMENTO XML PARA REPRESENTAÇÃO DO LAUDO.	76
EXEMPLO 17: ARQUIVO DTD PARA OS DOCUMENTOS INTERMEDIÁRIO DE SINTAGMAS	77
EXEMPLO 18: ARQUIVO DTD PARA OS DOCUMENTOS INTERMEDIÁRIO COM A INFORMAÇÃO DOS CONCEITOS	78
EXEMPLO 19: SENTENÇAS EXTRAÍDAS DOS LAUDOS DE RAIO-X DE TÓRAX.	89
EXEMPLO 20: APLICAÇÃO DO <i>CHUNK PARSER</i>	91
EXEMPLO 21: SENTENÇAS SEGMENTADAS	92
EXEMPLO 22: PADRÕES DE REAGRUPAMENTO DE <i>CHUNKS</i>	93
EXEMPLO 23: SENTENÇA EXTRAÍDA DE UM LAUDO E ROTULADA SINTATICAMENTE.	98
EXEMPLO 24: DADOS PARA CURVA ROC, TESTE 4	125
EXEMPLO 25: EXEMPLO DE FRASE EXTRAÍDA DE UM LAUDO ANOTADO.	130
EXEMPLO 26: EXEMPLO DE ROTULAÇÃO DE CONCEITOS	140
EXEMPLO 27: OPERAÇÕES E CONSTANTES USADAS EM EXPRESSÕES REGULARES SEGUNDO O PADRÃO POSIX	158

Sumário

1	INTRODUÇÃO	1
1.1	OBJETIVOS	2
1.2	MOTIVAÇÃO.....	3
1.3	JUSTIFICATIVA E RELEVÂNCIA	4
1.4	SISTEMAS DE INFORMAÇÃO EM MEDICINA	5
1.5	BASES DE CONHECIMENTO EM SAÚDE	6
1.5.1	<i>Armazenamento de Texto Livre</i>	9
1.6	ESTRUTURA DA TESE	10
2	REVISÃO BIBLIOGRÁFICA	11
2.1	PROCESSAMENTO DE LINGUAGEM NATURAL	11
2.1.1	<i>Conceitos Principais em PLN</i>	14
2.1.2	<i>Expressões Regulares</i>	18
2.2	FERRAMENTAS PARA ANÁLISE COMPUTACIONAL DE TEXTOS	20
2.2.1	<i>Analizador Léxico</i>	21
2.2.2	<i>Delimitadores de Sentenças</i>	21
2.2.3	<i>Rotuladores</i>	22
2.2.4	<i>Stemmers</i>	24
2.2.5	<i>Sintagmas Nominais e Chunk Parsing</i>	26
2.2.6	<i>NLTK (Natural Language Toolkit)</i>	29
2.3	RECUPERAÇÃO DE INFORMAÇÃO (RI)	30
2.3.1	<i>Extração de Informação</i>	30
2.4	AValiação DA RECUPERAÇÃO DE INFORMAÇÃO	31
2.4.1	<i>Receiver Operating Characteristics (ROC)</i>	33
2.5	ALGUNS TRABALHOS EM PROCESSAMENTO DE LINGUAGEM NATURAL E EXTRAÇÃO DE INFORMAÇÃO	36
2.5.1	<i>MUC (Message Understanding Conferences)</i>	36
2.5.2	<i>Fastus (Finite State Automaton Text Understanding System)</i>	36
2.5.3	<i>MedLEE (Medical Language Extraction and Encoding System)</i>	36
2.6	VOCABULÁRIOS MÉDICOS E ONTOLOGIAS	37
2.6.1	<i>Ontologias</i>	37
2.6.2	<i>Protégé</i>	40
2.7	SISTEMAS DE TERMINOLOGIAS.....	41
2.7.1	<i>SNOMED CT (SNOMED Clinical Terms)</i>	41
2.7.2	<i>MeSh (Medical Subject Headings)</i>	41
2.7.3	<i>UMLS (Unified Medical Language System)</i>	42
2.7.4	<i>RadLex</i>	51
2.8	ALGUNS TRABALHOS EM LINGUA PORTUGUESA NA ÁREA MÉDICA	53
2.8.1	<i>DECs (Descritores de Ciências e Saúde)</i>	53
2.8.2	<i>LILACS - Literatura Latino-americana e do Caribe em Ciências da Saúde</i>	53
2.9	TRABALHOS RELACIONADOS	54
2.9.1	<i>Avaliação da Tradução Automatizada de Relatórios de Radiografias de Tórax</i>	54
2.9.2	<i>Base de Imagens para Avaliação de Algoritmos de Processamento de Imagens</i>	54
3	METODOLOGIA	55
3.1	METODOLOGIA DE PROJETO E DESENVOLVIMENTO	55
3.2	ORGANIZAÇÃO DO TRABALHO	59
3.2.1	<i>Premissas para Organização do Trabalho</i>	59
3.2.2	<i>Metodologia de Avaliação do UMLS</i>	60
3.2.3	<i>Metodologia de Organização do Corpus</i>	62
3.3	CONSTRUÇÃO DE ONTOLOGIAS	64
3.4	DEFINIÇÃO DA ESTRUTURA DE REPRESENTAÇÃO DE CONHECIMENTO DOS LAUDOS	65
3.5	METODOLOGIA DE AVALIAÇÃO	65
4	ANÁLISE E PROJETO.....	67
4.1	CONTEXTO DO TRABALHO	67
4.2	ANÁLISE INICIAL DO PROBLEMA	68

4.2.1	<i>Estrutura dos Laudos</i>	71
4.2.2	<i>Sistemas de Terminologias</i>	72
4.3	ARQUITETURA GERAL.....	73
4.4	ESTRUTURA DE ARQUIVOS DE PROCESSAMENTO	76
4.4.1	<i>Estrutura de Dados para a Saída de Laudos</i>	76
4.4.2	<i>Estrutura do Arquivo de Segmentos</i>	77
4.4.3	<i>Estrutura do Arquivo de Conceitos</i>	78
4.5	ONTOLOGIA E LÉXICO UTILIZADOS.....	79
4.5.1	<i>Estrutura Relacional de Armazenamento do RadLex</i>	81
4.5.2	<i>Sumário de Conceitos do Radlex</i>	81
4.5.3	<i>Índice de Sentenças Normalizadas e Índice de Palavras Normalizadas</i>	82
4.6	MAPEAMENTO ENTRE RÓTULOS DOS LAUDOS E ONTOLOGIA MÉDICA	84
4.7	ALGORITMOS.....	86
4.7.1	<i>Busca de Strings</i>	86
4.7.2	<i>Segmentação de Sentenças (Extração de Sintagmas)</i>	89
4.7.3	<i>Identificação de Conceitos</i>	94
5	IMPLEMENTAÇÃO	96
5.1	ESTRUTURAÇÃO DE UM SISTEMA DE ONTOLOGIA MÉDICA	96
5.1.1	<i>Avaliação de Ontologias</i>	96
5.2	CORPUS DE TREINAMENTO E TESTE	97
5.2.1	<i>Extração dos Laudos da Base Hospitalar e Armazenamento no Banco de Dados do Projeto</i>	97
5.2.2	<i>O Corpus de Treinamento</i>	97
5.2.3	<i>O Corpus de Teste</i>	98
5.3	POS-TAGGER.....	99
5.3.1	<i>Treinamento</i>	100
5.4	ALGORITMOS.....	100
5.4.1	<i>Algoritmo de Comparação e Busca de Sentenças</i>	100
5.4.2	<i>Segmentação de Sentenças</i>	101
5.4.3	<i>Identificação de Conceitos</i>	101
5.5	MAPEAMENTO DE MODELOS	101
5.6	TRABALHOS A SEREM CONCLUÍDOS	102
5.6.1	<i>Anotação Semântica de Textos</i>	102
6	RESULTADOS	104
6.1	ONTOLOGIAS.....	104
6.1.1	<i>Modelo de Representação dos Laudos</i>	104
6.1.2	<i>Organização das Ontologias Médicas</i>	104
6.2	AValiação DO ALGORITMO DE BUSCA.....	105
6.2.1	<i>Conjunto de Testes</i>	106
6.2.2	<i>Avaliação da Recuperação e da Precisão</i>	110
6.2.3	<i>Curva ROC</i>	119
6.2.4	<i>Considerações Gerais Sobre a Avaliação</i>	126
6.3	AValiação DO VOCABULÁRIO UMLS.....	127
6.4	AValiação DO BRILL TAGGER.....	129
6.4.1	<i>Anotação Sintática dos Laudos para Treinamento</i>	129
6.5	AValiação DE ALGORITMOS DE PLN.....	132
6.5.1	<i>Avaliação da Segmentação de Sentenças</i>	132
6.5.2	<i>Avaliação da Identificação de Conceitos</i>	137
6.6	AValiação GERAL.....	141
7	DISCUSSÃO	142
7.1	AValiação DO DESENVOLVIMENTO	142
7.2	SOBRE ONTOLOGIAS	143
7.3	SOBRE FORMATAÇÃO E RESTRIÇÃO DE VOCABULÁRIOS	145
7.4	SOBRE PLN EM PORTUGUÊS	146
7.5	SOBRE AS FERRAMENTAS DESENVOLVIDAS	147
7.6	SOBRE OS OBJETIVOS ALCANÇADOS	147
7.7	CONTINUIDADE DO TRABALHO	148
8	CONCLUSÕES	150

9	REFERÊNCIAS BIBLIOGRÁFICAS.....	155
10	ENDEREÇOS ELETRÔNICOS.....	157
11	ANEXOS.....	158
11.1	CONSTANTES E OPERADORES PARA DEFINIÇÃO DE EXPRESSÕES REGULARES SEGUNDO O PADRÃO POSIX. 158	
11.2	HIERARQUIA DE CHOMSKY	159
11.3	ORGANIZAÇÃO DOS ARQUIVOS/TABELAS DO UMLS	160
11.4	CONJUNTO DE RÓTULOS ADAPTADOS DO <i>BROWN COPUS</i>	162
11.5	PUBLICAÇÕES RESULTANTES DA TESE.....	163
11.6	EXEMPLOS DE LAUDOS ROTULADOS.....	164
11.7	DIAGRAMA ER PARA O ARMAZENAMENTO DOS LAUDOS.....	165

1 INTRODUÇÃO

O setor de Saúde tem usado sistemas computadorizados por muitos anos, porém apenas mais recentemente o potencial dos sistemas informatizados tem sido melhor explorado. Hoje conhecemos as inúmeras possibilidades trazidas pela utilização de computadores em tal setor.

Neste trabalho há um interesse na organização e recuperação de textos livres na área de medicina como fonte de conhecimento para estudantes e profissionais que possam usar a base hospitalar na busca de informação em laudos ou mesmo no prontuário do paciente.

Tal interesse nasce do fato que uma base de dados hospitalar é alimentada diariamente com uma quantidade muito grande de texto livre, visto que os profissionais de saúde sempre estiveram habituados a escrever desta forma, não apreciando a utilização de formulários com campos fixos em muitas situações. Dá-se ao usuário a liberdade de expressão em detrimento da capacidade de busca por informações no banco de dados hospitalar, uma vez que a busca direta em bancos de dados de texto livre é uma tarefa complexa e cara em termos computacionais.

Apesar da deficiência dos métodos de busca em banco de dados em texto livre, a capacidade de busca é tão importante para o profissional de saúde, quanto a liberdade de expressão. Uma solução possível é a formatação automática do texto livre através de algoritmos de Processamento de Linguagem Natural (PLN).

As técnicas de PLN podem ser aplicadas de forma a preencher um modelo de dados, tendo como base o texto livre gerado pelo profissional de saúde. O preenchimento do modelo consiste basicamente em extrair termos chaves dos textos e associar a eles uma codificação de conceito que facilite o processamento dos textos já dentro de uma base de dados.

Montada a base de dados, estudantes e pesquisadores futuramente poderão buscar material de estudo para seus trabalhos acadêmicos e científicos. Enquanto os textos formatados facilitam a busca do ponto de vista de programação, o texto livre, que continua sendo armazenado na base de dados, fornece ao usuário a informação procurada sem as restrições impostas por formulários padronizados. O armazenamento do texto livre é importante para a preservação do conteúdo original.

Definido o contexto, este trabalho propõe a utilização de técnicas de PLN para anotação automática de laudos em radiologia para posterior utilização em sistemas de recuperação de informação. Os laudos são anotados com códigos definidos em um sistema de ontologias médico para então serem armazenados em uma base de dados. Inicialmente optou-se por trabalhar com o sistema UMLS (*Unified Medical Language System*), um sistema multilíngue de terminologias na área de saúde que possui a definição de um vasto conjunto de conceitos pertinentes a área, além de oferecer informações semânticas adicionais e um conjunto de ferramentas destinadas a PLN para língua inglesa. Posteriormente optou-se por trabalhar com o RadLex, uma ontologia definida especificamente para a área de radiologia, porém definida apenas em inglês, o que gerou o trabalho de tradução da mesma.

O trabalho apresenta um estudo das vantagens e desvantagem da utilização de ontologias médicas em ferramentas de processamento de textos. Também é discutido o processo de organização de um sistema de terminologia para a utilização em sistemas de PLN..

Os dados escolhidos para realização do trabalho são formados por um conjunto de cerca de 2000 laudos de Raio-X de Tórax obtidos no InCor (Instituto do Coração / Hospital das Clínicas da Faculdade de Medicina da USP) tendo sido removida qualquer informação que permita a identificação dos pacientes.

1.1 Objetivos

Como discutido anteriormente, as bases de dados hospitalares são alimentadas diariamente com texto livre, apesar de tal prática não ser recomendada ([PEP 2003], pg. 28). Uma forma de contornar a questão do armazenamento de texto livre puro é anotar os textos de forma a facilitar sua manipulação por parte de sistemas computacionais.

O objetivo deste trabalho é anotar de forma automática um conjunto de laudos de radiologia, em língua portuguesa, com rótulos codificados segundo um sistema de terminologias médico. Permite-se assim que uma base de conhecimento seja alimentada com os textos dos laudos, facilitando a busca da informação contida no texto livre dos laudos.

A anotação automática é realizada através do uso de programas de PLN, obtendo-se como saída um documento XML cujos rótulos descrevem a codificação referente a cada

trecho de texto de interesse, além de adicionar informações semânticas contidas no sistema de terminologias utilizado.

Como resultado espera-se obter as ferramentas e as metodologias necessárias para a anotação de textos de forma automática permitindo que o conteúdo anotado seja acrescentado a uma base de conhecimento. Neste trabalho, se entende como anotação a adição de rótulos a um documento de forma a acrescentar alguma marcação ao mesmo, isto é, o sistema produzido recebe um laudo em texto livre e gera um documento XML que permita o processamento computacional posterior do conteúdo do laudo em um sistema de busca.

Neste contexto identificamos os seguintes objetivos específicos:

- ✓ Criação de uma estrutura de representação de conhecimento e mapeamento da mesma para um modelo de dados para armazenamento em um banco de dados;
- ✓ Estudo da adequação do uso de dois sistemas de terminologias médicas distintos, a saber o UMLS e do RadLex, no processamento de laudos de Raio-X de Tórax e organização de uma ontologia em português para uso em processamento de textos médicos;
- ✓ Criação e desenvolvimento de algoritmos e métodos para anotação dos laudos em português;

A união dos objetivos específicos deve resultar na ferramenta de anotação de laudos para alimentação de uma base de conhecimento médico.

1.2 Motivação

Apesar de não se recomendar o uso de texto livre em sistema de saúde, esta prática não pode ser encarada apenas como uma barreira cultural a ser vencida na implantação de novos sistemas, visto que tal prática foi usada por muito tempo e ainda possui alguma motivação para sua existência.

Tendo como peça central neste trabalho a questão do casamento entre requisitos de usuário e computacionais no armazenamento e busca de informação médica, evidenciamos os seguintes aspectos que servem de motivação para este trabalho:

- ✓ A contribuição para a pesquisa e desenvolvimento de ferramentas computacionais voltadas ao processamento de linguagem natural em português;
- ✓ O estudo de ontologias e modelos de representação de conhecimento na área de medicina em português;
- ✓ A validação da utilização de um vocabulário médico padronizado em conjunto com ferramentas de PLN;
- ✓ A criação de ferramentas que permitam adicionar formatação e organização a laudos escritos em texto livre é importante para o processo de aprendizado em medicina, uma vez que facilita o acesso a informação e facilita a recuperação de informações em bases de dados médicas de forma retrospectiva.

1.3 Justificativa e Relevância

O processo de documentação pode se tornar restritivo do ponto de vista do usuário, se feito na forma de simples formulários fixos, muitas vezes não sendo desejável abandonar o uso de texto livre. Este trabalho tenta adicionar alguma documentação, na forma de marcações, ao conhecimento contido em laudos médicos, contribuindo para a obtenção de uma documentação adequada dos laudos. Também é favorecido o acesso a informação de forma mais eficiente, visto que a pesquisa manual por informação médica pode ser trabalhosa e demorada.

Do ponto de vista técnico o trabalho contribui para as áreas de EI (Extração de Informação) e PLN, segundo os seguintes pontos:

- ✓ O estudo de vocabulários médicos e processamento de linguagem natural em português para Medicina;
- ✓ O desenvolvimento de técnicas e algoritmos para a anotação de textos livres em português;

No que diz respeito a bases de conhecimento, o trabalho apresenta contribuições referentes a estudo de modelos de representação de conhecimento, visto que uma base de conhecimento deve possuir um modelo de representação da informação que permita a pesquisa e a recuperação da informação de forma eficiente.

Do ponto de vista dos profissionais de saúde o trabalho é importante para a obtenção de ferramentas que permitam a busca por conhecimento médico específico de forma mais eficiente, permitindo a pesquisa retrospectiva no conteúdo de uma base de laudos hospitalar.

Do ponto de vista técnico o trabalho contribui com o desenvolvimento de pesquisas, programas e material sobre processamento de linguagem natural em língua portuguesa e para a avaliação do uso de um vocabulário especializado para medicina, também em português e no desenvolvimento de ferramentas voltadas a documentação automática em medicina.

1.4 Sistemas de Informação em Medicina

A medicina ocidental possui algumas dezenas de séculos de história, que chegou até nós através de manuscritos de diversas fontes. Quando pensamos no surgimento da medicina ocidental, o primeiro nome que lembramos é Hipócrates, considerado o pai da medicina. Em seu ofício como médico, filósofo e professor, adotou uma postura mais científica para a prática médica. Entre tantos méritos que reconhecemos em Hipócrates e seus sucessores na história da medicina, evidenciamos neste texto a prática de registrar o conhecimento e as técnicas médicas de forma escrita. Apesar de muitos dos textos antigos nunca terem chegado até a época contemporânea, se não fosse a escrita, a preservação do conhecimento seria uma tarefa muito mais difícil.

Nos dias de hoje, diversas ferramentas viabilizam a preservação do conhecimento. Temos como principais objetos a impressão e a tipografia (tecnologia da prensa móvel desenvolvida por Gutenberg em 1440) e a informatização. Independentemente da ferramenta utilizada, o meio principal de preservação de conhecimento continua sendo a escrita.

O conteúdo médico, informatizado ou não provém de diversas fontes, como fontes bibliográficas, dados de pacientes, protocolos e procedimentos, etc. No entanto, as ferramentas, por si só, usadas para armazenar a informação médica não são suficientes para tal trabalho, também é necessária uma forma de organização do conhecimento. Todo o conjunto de informações necessita de uma organização mínima para que possa ser gerenciado e utilizado adequadamente, pois a informação só tem utilidade se puder ser reconhecida dentro do contexto adequado.

O armazenamento e o gerenciamento do grande volume de informação gerado diariamente em um hospital seria muito difícil ou talvez inviável sem a existência de sistemas

informatizados. Hoje em dia contamos com a utilização de prontuários multimídia de pacientes, armazenados em bases de dados que permitem a manutenção da informação médica por anos, segundo requisitos legais, mas que também organizam esses dados de forma a facilitar a pesquisa e o acesso.

Atualmente a organização das bases de dados nos sistemas de informação médica atende requisitos que viabilizam a manutenção do fluxo de trabalho de uma instituição de saúde, de forma a proporcionar segurança, confidencialidade, consistência e gerenciamento do ciclo de vida dos dados.

Com todos os anos de funcionamento dos sistemas de informação médica, uma quantidade muito grande de informação acaba sendo armazenada em formato eletrônico e apesar de todo desenvolvimento no que diz respeito a aspectos organizacionais permanece ainda uma área pouco explorada a possibilidade de criação de técnicas mais avançadas de pesquisa sobre esses dados.

Através do curso da história da medicina, percebemos que a manutenção e o gerenciamento da informação são primordiais para o desenvolvimento do conhecimento, no entanto no estado que nos encontramos agora existe uma forte valorização apenas dos aspectos funcionais dos sistemas ligados a organização de dados médicos. É claro que essa organização é importantíssima para o atendimento de qualidade ao paciente, porém ainda resta um campo muito amplo de pesquisa sobre sistemas de informação médica, campo este ligado ao potencial da informação armazenada nas bases hospitalares, que permanece inexplorado.

A utilização de aplicações de recuperação e extração de informação e de mineração de dados, revelam-se como boas opções para explorar o potencial de toda a informação em bases de dados médicas, porém neste trabalho há um maior interesse nas técnicas de recuperação e extração de informação.

1.5 Bases de Conhecimento em Saúde

Segundo [PEP 2003] (capítulo 10, pg. 144) grande parte das decisões em saúde são tomadas com base na experiência do profissional e na literatura disponível, que pode incluir livros, manuais, artigos ou mesmo repositórios de dados clínicos. Muitas vezes as fontes de

informações podem ser ultrapassadas ou se apresentarem de forma tendenciosa, dependendo do ponto que se deseje provar.

Sob este enfoque, uma base de dados alimentada constantemente com resultados clínicos e organizada de maneira a permitir o acesso a informação de forma sistematizada pode auxiliar na prática médica, uma vez que serve como fonte de evidências para estudos científicos ou para estudos de casos clínicos.

Podem existir diversos tipos de materiais armazenados em uma base de conhecimento na área de saúde, como artigos, imagens, laudos, resumos de discussões médicas ou mesmo dados extraídos dos prontuários hospitalares. Como já discutido anteriormente, grande parte da informação encontra-se na forma de texto livre, sendo esta uma característica inerente a informação. Toda essa informação, quando agrupada gera um volume de dados muito grande sendo o gerenciamento tradicional e não automatizado muito custoso.

A construção de novas bases de conhecimento pode ser facilitada com a utilização de computadores e sistemas que permitam o armazenamento estruturado e a utilização de ferramentas voltadas a gestão do conhecimento. Entre ferramentas utilizadas em bases de conhecimentos podemos citar o *Data warehouse*, ferramentas de indexação e classificação de conteúdo ou de Recuperação de Informação.

Entre os aspectos envolvidos na organização de uma base de conhecimento temos a necessidade de adoção de padrões de dados para armazenamento e comunicação entre sistemas, o gerenciamento de volumes grandes de dados multimídia, a segurança da informação, visto que os dados de pacientes são sigilosos, a confiabilidade do sistema e as ferramentas de acesso e busca a informação.

O fato das bases de informação conterem dados pouco estruturados, muitas vezes ambíguos e em linguagem natural ou em formatos diversos como som e imagem, gera a necessidade de ferramentas de recuperação de informação e não simplesmente de busca por palavras chaves através da definição de expressões ou de consultas em uma linguagem de busca.

Devido a natureza multimídia dos dados, a busca pode ser feita através de ferramentas de processamento de linguagem natural, no caso de textos livres, através de algoritmos de processamento de imagens no caso de imagens e vídeos ou mesmo através de processamento

sonoro. Em linhas gerais temos de um lado o usuário que busca por informação e fornece através de uma interface uma descrição do que procura e de outro lado temos um repositório contendo a informação em potencial desejada.

Normalmente, em sistemas voltados a recuperação de dados, as buscas são feitas através do fornecimento de algumas palavras chaves e uma ferramenta de busca, que procura por dados em uma base estruturada. As buscas geralmente são booleanas, isto é, o sistema pode encontrar os dados que sejam descritos por uma determinada palavra chave ou simplesmente não encontrar nada. Tais sistemas não se adequam às necessidades das bases de informação, que armazenam dados não estruturados e em formatos variados como descrito anteriormente. Em contrapartida, sistemas de recuperação de informação lidam com a falta de estrutura ou ambigüidade inerente à informação armazenada como texto livre ou imagens.

Por outro lado, sistemas de recuperação de informação armazenada como texto livre ou fala necessitam de uma série de ferramentas de PLN. Quando a informação se encontra armazenada como sons, mais especificamente como fala, são necessárias ferramentas de processamento de sinais que permitam o reconhecimento da fala, restando ainda o problema de estruturação da informação. Quando a informação se trata de texto livre, existe uma série de ferramentas de lingüística computacional para processamento dos textos.

Apesar dos diversos tipos de técnicas que podem ser usadas para a recuperação de tipos diferentes de informação, um requisito constante é a criação de uma estrutura mínima que permita a organização, a indexação e o agrupamento do conteúdo trabalhado.

Segundo o contexto deste trabalho a questão das bases de informação é tratada sob o enfoque da estrutura de armazenamento de textos e da anotação automática com o propósito de facilitar a busca por informação em textos livres, no caso laudos.

A documentação automática de textos livres, isto é, a criação de modelos que são preenchidos automaticamente com informação extraída dos textos, contribui para a criação de uma estrutura de armazenamento que facilita a recuperação da informação. Os modelos de dados extraídos dos textos aproximam a técnica de busca aos métodos usados em recuperação de dados, considerando as devidas restrições relacionadas às técnicas de processamento de linguagem natural.

Toda a estrutura organizacional da informação é importante para o armazenamento e para a utilização por ferramentas computacionais, no entanto, uma base de informação médica requer, além da estrutura de dados, a adoção de um sistema de terminologias que viabilize a criação de uma estrutura semântica de organização da informação. A utilização de um sistema de terminologias além de fornecer um vocabulário padronizado estabelece a relação entre diversos conceitos médicos. Por exemplo, se existe a palavra febre em um documento, através do sistema de terminologias é possível saber que tal conceito constitui um sintoma ou uma queixa do paciente.

Como foi exposto neste item, a criação de uma base de informação médica requer não apenas o esforço de armazenamento de dados, mas também de criação de uma estrutura organizacional, a adoção de terminologias que descrevam a informação, o armazenamento de dados inicialmente desestruturados em diversos formatos e a criação de ferramentas de busca na base. O que fica evidenciado é que uma base de informação não é apenas um repositório de documentos, mas sim um conjunto de ferramentas tanto para armazenamento como para recuperação da informação.

1.5.1 Armazenamento de Texto Livre

Nas bases de conhecimento médico ainda há uma quantidade muito grande de informação armazenada na forma de texto livre, principalmente laudos. Atualmente há alguma perspectiva de mudança neste hábito devido a adoção de dicionários padronizados nas instituições de saúde, mesmo por que ao fornecer um laudo médico, este deve estar em um formato compreensível para qualquer profissional que venha a precisar desta informação. No entanto ainda há uma vasta base legada em texto livre, sendo que a adoção de dicionários e codificações não garante a completa eliminação do texto livre do cotidiano de emissão de laudos médicos.

Se por um lado o texto livre é importante para a expressão da informação, por outro pode gerar uma série de outros problemas, como ambigüidade e má interpretação, dificuldades para busca por informação nas bases de dados e constante intervenção humana no momento da organização da informação.

Na tentativa de melhorar a automatização dos processos de organização da informação e mesmo de diminuir a possibilidade de má interpretação dos textos, a utilização de sistemas

de terminologias padronizados e de técnicas de processamento de linguagem natural se revelam de grande interesse.

Através das terminologias padronizadas e de técnicas de processamento de linguagem natural, busca-se o objetivo de aumentar a eficiência da procura por informações em bases de dados médicas além de oferecer um vocabulário padronizado que minimize problemas de interpretação da informação.

1.6 Estrutura da Tese

O texto desta tese pode ser segmentado em quatro partes. A primeira parte, que compreende os capítulos de introdução e a revisão bibliográfica foram escritos com o objetivo de oferecer ao leitor uma visão geral do problema estudado e fornecer uma introdução teórica básica sobre as disciplinas envolvidas no trabalho, como o PLN, por exemplo.

A segunda parte do texto, referente aos capítulos de metodologia e implementação, tem por objetivo descrever as opções de implementação, as estruturas escolhidas para trabalho e os algoritmos implementados. Através do capítulo de metodologias pretende-se descrever o processo de desenvolvimento de forma a deixar claro como o trabalho foi organizado e como os experimentos foram conduzidos para que possam ser reproduzidos em outras situações. O capítulo de implementação descreve a organização estrutural de programas e algoritmos.

A terceira parte do texto é destinada a descrição dos resultados obtidos, segundo os métodos propostos nos capítulos anteriores. Cada experimento realizado é acompanhado por comentários e conclusões, tentando-se expor os pontos fortes e fracos evidenciados em cada experimento realizado.

A quarta parte do texto, que compreende os capítulos de discussão e conclusão, tem por objetivo expor uma análise geral dos resultados obtidos. Neste ponto é feita uma avaliação do quanto os resultados obtidos estão próximos dos objetivos originais e da influência nos trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

2.1 Processamento de Linguagem Natural

O Processamento de Linguagem Natural é o processamento computacional de linguagens utilizadas por humanos para comunicação através de fala ou escrita. Constitui um tema de pesquisa de alta complexidade e que envolve diversos aspectos computacionais. Para melhor entender tais aspectos, será adaptado neste texto o exemplo dado por [Jurafsky 2000]. No capítulo introdutório de seu livro, Jurafsky e Martin questionam quais seriam as características necessárias para a criação dos sistemas ligados a fala do computador HAL do filme “2001 uma Odisseia no Espaço”.

O computador Hall precisaria ser dotado de um sistema de **reconhecimento de fala** (*speech recognition*), que pudesse capturar um sinal de áudio e o reconhecer como fala, um sistema de **Entendimento de Linguagem Natural** (*Natural Language Understanding*) para interpretar o conteúdo de um diálogo, um sistema de comunicação para **gerar a linguagem natural** (*Natural Language Generation*) e um sistema de **síntese de fala** (*speech synthesis*). Os sistemas de reconhecimento de fala, de síntese de fala e de geração de fala serviriam como interface que possibilitariam ao HAL capturar a informação vinda do mundo exterior na forma de fala e para que ele pudesse expor suas respostas ao mundo também na forma de fala.

Além da interface com o mundo exterior, o HAL também precisaria de sistemas que permitissem a ele buscar a informação necessária para que pudesse chegar a conclusões, segundo um contexto. Desta forma, ele precisaria de um sistema de **Recuperação de Informação** (*Information Retrieval*), para encontrar onde as informações textuais necessárias estão armazenadas, um sistema de **Extração de Informação** (*Information Extraction*) para extrair somente os fatos pertinentes das bases textuais e por fim ele deveria ser capaz de fazer **inferências**, isto é, chegar a conclusões baseadas nos fatos obtidos.

Sistemas de tal complexidade ainda estão longe de serem obtidos, porém as pesquisas referentes às áreas mencionadas se encontram em constante evolução. Tal pesquisa é denominada de **Processamento de Fala e Linguagem** (*Speech and Language Processing*) e pode ser dividida em: Processamento de Linguagem Natural (*Natural Language Processing*), Linguística Computacional (*Computational Linguistics*) e Processamento e Síntese de Fala (*Speech and Language Synthesis*).

A Linguística Computacional é uma área de pesquisa interessada na descrição formal de linguagens sob o enfoque do processamento automático das mesmas, com o objetivo de construir programas que processam palavras e textos em linguagem natural (Adaptado de [Bolshakov 2004]).

O Processamento e Síntese de Fala é o campo de estudo ligado ao reconhecimento de sinais de áudio como sendo sinais de uma linguagem falada e a produção de fala através de sistemas computacionais.

Apesar de não mencionado, o computador Hall deveria ser dotado de inteligência para que pudesse fazer as inferências necessárias para seus diálogos e atos, no entanto a área de inteligência artificial está fora do escopo deste trabalho, não sendo detalhada.

Um segundo exemplo a ser usado refere-se a um trabalho chamado **Eliza** de 1966 publicado por Weizenbaum [Weizenbaum 1966]. Eliza é capaz de manter uma conversa simples com uma pessoa, como em um *chat*. O programa assume o papel de um psicanalista que gera respostas extraindo a informação básica da própria fala do paciente.

Apesar de existirem muitas lendas a respeito de pessoas convencidas de se tratar de uma pessoa real, Eliza usa uma lógica simplista baseada em regras, não havendo algoritmos para Recuperação de Informação ou inferência em sua implementação, no sentido de manter grandes bases de conhecimento, o que leva a uma rápida deterioração da conversa com o programa. Segue a transcrição de uma conversa com o programa Eliza, exemplificando seu funcionamento.

Exemplo 1: Exemplo de conversa com o programa Eliza (Exemplo extraído de [Jackson 2002])

Eliza: how are you?	Eliza: i see
PJ: not so good.	PJ: yup
Eliza: why do you say not so good?	Eliza: a while back you talked about your face
PJ: i suffer from allergies.	PJ: what about your face?
Eliza: that's interesting.	Eliza: what is it you really want to know?
PJ: i come out in spots.	PJ: are you my kind of girl?
Eliza: i'm not sure i know what you are talking about.	Eliza: perhaps in your fantasies I am your kind of girl
PJ: spots on my face.	PJ: in my fantasies you are blonde
Eliza: your face.	Eliza: what makes you think i am blonde?
PJ: that's right.	PJ: I don't know
Eliza: is it really that right?	Eliza: don't you really know?
PJ: you're a program, aren't you?	PJ: no
Eliza: does it please you to believe I am a program, am not I?	Eliza: you are being a bit negative
PJ: it sure does	PJ: no i'm not
	Eliza: how long have you been not?

De forma simplista poderíamos comparar o sistema Eliza com um sistema de Extração de Informação, visto que através de análise sintática das frases e de regras, o sistema extrai palavras chaves do diálogo, atribui-lhes uma pontuação e gera respostas através da substituição de palavras na frase original. Resumidamente, Eliza não entende a conversa, apenas aplica um conjunto de regras codificadas em *scripts* produzindo uma conversa aleatória, mas que leva a ilusão de uma conversa real.

Comparando o Eliza (um trabalho real) com o Hall (o produto da imaginação de um escritor) vemos como estamos distantes de sistemas capazes de captar, entender e gerar fala. Citando Jurafsky e Martin:

“A habilidade das máquinas processarem a linguagem como nós o fazemos simbolizará a chegada de máquinas verdadeiramente inteligentes. Somos levados a acreditar nisso pelo fato de que o uso efetivo da linguagem está entrelaçado com, habilidades cognitivas gerais.” (Adaptado de [Jurafsky 2000])

Este trabalho, menos ambicioso que a produção de um sistema como o Hall, mas mais complexo que o Eliza, pretende usar os recursos computacionais não para entender a linguagem natural e sim para documentar e extrair informação de textos escritos em linguagem natural na área de medicina. Assim sendo, este trabalho tem um interesse especial em PLN na forma de textos digitais na área médica, tema que será melhor desenvolvido no texto subsequente.

Segundo este contexto, defini-se como objetivo do trabalho analisar documentos escritos em linguagem natural de forma a convertê-los em textos que possam ser processados por sistemas computacionais voltados a extração e recuperação de informação.

Resta ainda uma última questão não respondida nesta introdução: “Por que utilizar o PLN?”. A resposta a esta questão é simples. A utilização de sistemas de informação cresce diariamente, com isso muita informação textual acaba armazenada em bancos de dados ou na *Web*, no entanto essas informações apresentam utilidade somente se puderem ser recuperadas de forma fácil e rápida, segundo o contexto ao qual pertencem.

No caso de textos médicos, por causa de fatores culturais os profissionais de saúde mantiveram o hábito de armazenar texto livre nas bases de dados hospitalares por anos, o que tornou a busca de informação nessas bases uma tarefa árdua. Muitas vezes desenvolvedores

de aplicações se vêm obrigados a classificar ou indexar textos manualmente, para que estes possam ser recuperados em sistemas de busca.

Para projetar programas voltados ao processamento de linguagem natural, é necessário o conhecimento de uma série de conceitos sobre as regras de formação de sentenças e palavras em uma linguagem. Os itens subseqüentes devem fornecer o conhecimento básico para o acompanhamento deste trabalho, bem como fornecer algumas definições úteis.

2.1.1 Conceitos Principais em PLN

Para entender o que é o PLN, começa-se definindo o que é uma linguagem. Segundo o dicionário Le Robert¹ a linguagem é “a função de expressão do pensamento e de comunicação entre os homens, seja através da fala ou da escrita”.

Uma segunda definição mais interessante do ponto de vista computacional, foi elaborada por [Chomsky 1956]. Para Chomsky uma linguagem é um “conjunto finito ou infinito de sentenças, cada uma de tamanho finito, todas construídas através de um alfabeto finito de símbolos.”.

No contexto desta tese, a segunda definição possui a informação necessária para começarmos a esboçar o pensamento em um formato mais adequado à computação, onde a linguagem é tratada como um agrupamento de letras de um alfabeto na forma de palavras, que por sua vez constituem sentenças, que agrupadas formam um texto.

Neste trabalho, por um lado há o interesse de capturar parte da expressão do pensamento em textos médicos, de forma a resumir o texto e permitir que uma máquina identifique seu conteúdo, porém para isso é necessário tratar a linguagem conforme a definição de *Chomsky*. Neste ponto surge a necessidade de definir o que é o Processamento de Linguagem Natural (PLN).

Segundo Jackson e Moulinier “o termo **Processamento de Linguagem Natural (PLN)** é normalmente usado para descrever a função de componentes de *software* ou *hardware* em um sistema de computadores que analisa ou sintetiza a linguagem escrita ou falada. O termo *natural* serve para distinguir a linguagem humana de linguagens mais formais

¹ Lê Robert: Dictionnaire de la Langue Française – Edition Poche, Nouvelle édition, 1998

como notações matemáticas, notações de lógicas ou linguagens de computador, como Java, LISP ou C++” [Jackson 2002].

Quando o intuito do processamento da linguagem é entender o conteúdo de um texto ou fala, é mais preciso falar em ELN ou **Entendimento de Linguagem Natural (NLU - *Natural Language Understanding*)**, porém este trabalho tem enfoque apenas em PLN, sendo deixado de lado um maior desenvolvimento do tema de ELN.

Tendo definido o que é a PLN, podemos passar para uma nova definição muito importante quando trabalhamos com linguagem natural. Dois conceitos a serem definidos estão ligados a lingüística. Normalmente podemos quebrar a linguagem em dois conceitos, a **sintática** e a **semântica**. A sintática está ligada as regras de formação estrutural de uma sentença enquanto a semântica está ligada ao significado da sentença.

Segundo [Jackson 2002], em muitas linguagens não naturais, como as linguagens de programação, por exemplo, o significado de uma sentença pode ser obtido através de sua estrutura se ignorando fatores como contexto, no entanto no caso de linguagens naturais a ambigüidade e a subjetividade acabam por gerar situações que dificultam a avaliação do significado.

Tanto o processamento da sintática como o da semântica estão ligados a mais uma série de definições básicas. Seguem tais definições.

Nesta tese um **caractere** é um símbolo gráfico usado por uma linguagem qualquer. Os símbolos gráficos são agrupados em **alfabetos**. Um alfabeto é a coleção dos símbolos existentes em uma linguagem. Aproveitando a definição de *Chomsky* [Chomsky 1956], qualquer seqüência formada pela concatenação dos símbolos de um alfabeto é um **string** (cadeia de caracteres) pertencente à linguagem do alfabeto em questão. Mais uma vez aproveitando uma definição dada por *Chomsky* [Chomsky 1956], tem-se que todos os **strings** que são sentenças de uma determinada linguagem são gerados pela **Gramática** da linguagem.

Pode-se pensar nos **strings** como representação de palavras ou sentenças completas, sendo uma palavra o agrupamento de um conjunto de caracteres alfanuméricos, enquanto uma sentença é um agrupamento de palavras. Tanto a formação de uma palavra como a de uma sentença acontece segundo regras.

Na realidade uma gramática fornece o conjunto de regras de formação das sentenças de uma linguagem. As regras de formação de palavras são chamadas de **Regras Morfológicas** (*Morphological Rules*). As gramáticas e regras morfológicas estão ligadas a forma, portanto são conceitos ligados à sintática.

A **morfologia** é o estudo de como as palavras são formadas a partir de pequenas unidades de significado, os **morfemas**. Podemos dizer que os morfemas são a unidade mínima de significado em uma linguagem (adaptado de [Jurafsky 2000]). Por exemplo, a palavra Gatos é constituída pelo morfema Gato mais o morfema s.

Os morfemas podem ser classificados em (adaptado de [Jurafsky 2000]):

- ✓ **Raiz:** constitui a base da palavra, carregando seu significado principal.
- ✓ **Afixo:** Os afixos são morfemas combinados à raiz, com o objetivo de adicionar um significado ao principal. Podem ser:

- ✓ **Sufixo:** posicionados após a raiz;

<u>Gato</u>	<u>s</u>
radical	sufixo

- ✓ **Prefixo:** posicionados antes da raiz;

<u>tele</u>	<u>medicina</u>
prefixo	radical

- ✓ **Infixo:** Inseridos na raiz, isto é, entre os caracteres que formam a raiz da palavra. Este tipo de afixo não ocorre na língua portuguesa;
- ✓ **Circunfixo:** posicionados tanto antes como depois da raiz, isto é, englobam o radical. Este tipo de afixo não ocorre na língua portuguesa, mas ocorre em línguas como o alemão, por exemplo.

<u>ge</u>	<u>sag</u>	<u>t</u>
prefixo	radical	sufixo

Obs: Em alemão o verbo *sagen* (dizer) tem o particípio passado *gesagt*.

Existem duas maneiras de formar uma palavra a partir de morfemas: a **derivação** e a **inflexão**.

A inflexão é a combinação de uma palavra com um morfema gramatical usualmente resultando em uma segunda palavra da mesma classe gramatical, satisfazendo as mesmas funções sintáticas da palavra original. (adaptado de [Jurafsky 2000])

A derivação é a combinação de um radical de uma palavra com um morfema gramatical usualmente resultando em uma palavra de classe gramatical diferente. (adaptado de [Jurafsky 2000])

Em computação para reconhecermos as partes constituintes de uma palavra é necessário aplicar um algoritmo que faça o **parsing morfológico** (*morphological parsing*) da mesma. O *parsing* é o processo de adquirir uma entrada e gerar uma saída estruturada. O programa que realiza o *parsing* é chamado de *parser*.

Ao processar textos em linguagem natural pode ser muito mais eficiente identificar palavras através do reconhecimento de seu radical (*stem*) do que manter um dicionário de grandes proporções que abrigue cada variação que uma palavra pode assumir.

Por exemplo o *stem* da palavra “gatos” ou da palavra “gatas” é “gat”. O processo de identificação do radical de uma palavra a partir de uma derivação é chamado de *Stemming*. O programa que realiza este processo é denominado *Stemmer*. Um *stemmer* nada mais é do que um *parser* morfológico. Em muitas situações as flexões de uma palavra não são necessárias, importando apenas a categoria gramatical e o significado principal da palavra, por este motivo os *stemmer* acabam sendo muito utilizados.

Sabendo que as palavras são formadas por morfemas. Ainda resta classificar, dentro da morfologia, quais os tipos diferentes de regras que podem ser usadas para unir os morfemas. A morfologia considera essas regras sob três enfoques diferentes, dando origem a três tipos de morfologia das palavras: Morfologia Concatenativa (*Concatenative Morphology*), Morfologia não-concatenativa (*Nonconcatenative Morphology*) e a *Templatic morphology*.

- ✓ Morfologia Concatenativa: onde uma palavra é construída através da concatenação de morfemas
- ✓ Morfologia não-concatenativa: onde os morfemas são combinados de forma mais complexa do que a simples concatenação para formação de uma palavra.
- ✓ *Templatic Morphology*: usando uma definição simplificada, temos que as palavras são formadas através da aplicação dos afixos aos radicais através de modelos predefinidos. Este tipo de morfologia é aplicada a língua Hebraica, por exemplo. Em hebraico a raiz de uma palavra é formada por consoantes fixas, geralmente

três, sendo a inflexão ou derivação feita através da inserção de padrões formados por conjuntos de vogais. Ex: em hebraico a raiz *lmd* significa aprender ou estudar e pode ser combinada com o modelo CaCaC para obter a palavra *lamad*, que significa “ele estudou”. [Jurafsky 2000]

No caso do processamento da língua portuguesa, usamos algoritmos projetados segundo as regras da Morfologia Concatenativa.

Um ultimo conceito a ser definido neste capítulo é o **Léxico** (*lexicon*). Um léxico é um repositório de radicais e afixos, providos de informações como classificação gramatical das palavras. (adaptado de [Jurafsky 2000]).

Tendo definido alguns termos básicos, muito importantes para o entendimento do texto subsequente, passamos agora a discorrer sobre algoritmos e modelos computacionais usados no processamento de linguagens em geral.

2.1.2 Expressões Regulares

Uma **Expressão Regular** (*Regular Expression - RE*) ou ER é uma notação formal usada para descrever cadeias de símbolos (*strings*) de acordo com um conjunto de regras. São usadas nos mais diversos tipos de aplicação, sendo interessante no contexto deste trabalho o uso na caracterização de Linguagens Regulares² e na descrição de Autômatos de Estados Finitos³.

Basicamente uma ER é uma expressão usada para descrever um conjunto de *strings*, através do uso de um conjunto de operadores e constantes que descrevem cadeias de caracteres. Quando definimos uma expressão regular, estamos interessados na forma de uma cadeia de caracteres e não em seu significado, sendo possível validar a forma através de regras formais bem definidas e não através de heurísticas. As expressões regulares servem para que se defina o conjunto de *strings* pertencentes a qualquer linguagem, com algumas restrições que estudaremos mais adiante.

² Linguagens baseadas em gramáticas regulares, que por sua vez podem ser descritas por expressões regulares e representadas através de autômatos finitos.

³ Segundo [Jackson 2002], um Autômato de Estados Finitos é uma máquina idealizada que se move instantaneamente de um estado interno para outro em uma série de passos discretos.

A Tabela 1 mostra as constantes normalmente usadas em expressões regulares dado um alfabeto finito Σ , enquanto a Tabela 2 e a Tabela 3 mostram os operadores.

Tabela 1: Constantes usadas em expressões regulares

Símbolo	Descrição
ϕ	Conjunto vazio
ϵ	String vazio
Caracteres literais	Caracteres pertencentes ao alfabeto Σ .

Tabela 2: Operadores usados em expressões regulares

Símbolo	Nome	Descrição
	Concatenação	Indica a concatenação de caracteres. Se RS denota o conjunto $\{ \alpha\beta \mid \alpha \text{ em } R \text{ e } \beta \text{ em } S \}$.
	União	União dos conjuntos de símbolos. Ex: $a b$ produz o conjunto $\{a, b\}$
*	Kleene Star	Indica o conjunto de strings que pode ser obtido através da concatenação de zero ou mais vezes do string que precede o operador. Ex a^* gera $\{\epsilon, a, aa, aaa, aaaa, \dots\}$

Tabela 3: Operadores redundantes usados em expressões regulares

Símbolo	Descrição
+	Ex: $a+ = aa^*$
?	Ex: $a? = (\epsilon \mid a)$

Usando as constantes e operações acima podemos, por exemplo, gerar ou reconhecer as palavras gato, gata, gatos e gatas através da expressão regular “ $gat(a|o)(s)?$ ”. Os parênteses servem para agrupar caracteres, definindo precedência e escopo.

As expressões regulares costumam ser usadas em aplicações que necessitam fazer o reconhecimento de cadeias de caracteres, como por exemplo, a busca de palavras em um texto ou para validação de valores de campos em um banco de dados através de máscaras. Tais aplicações possuem um conjunto de operações e constantes mais amplo, facilitando a definição das expressões. A notação mais comum para expressões regulares é a *Posix*, um padrão que define a sintaxe e as operações possíveis para a definição das expressões. Em geral, as aplicações *Unix* usam este padrão. Por exemplo, em notação *Posix* a expressão “ $([0-9]\{2\}\backslash-)?[0-9]?[0-9]\{3\}\backslash-?[0-9]\{4\}$ ” reconhece números telefônicos de sete ou oito dígitos, podendo ou não haver o código DDD. A expressão dada possui a seguinte interpretação:

- ✓ $[0-9] \Rightarrow$ qualquer caractere numérico entre zero e nove;
- ✓ $\{2\} \Rightarrow$ quantidade de repetições entre a expressão que precede o operador. Assim temos que “ $[0-9]\{2\}$ ” significa a concatenação de dois caracteres numéricos.

- ✓ ? => indica que a expressão precedente deve aparecer ou uma ou nenhuma vez (interprete o nenhuma vez como o *string* vazio). Assim a sub-expressão “[0-9]{2}-)” indica que o DDD não é obrigatório.
- ✓ A sub-expressão “[0-9]?[0-9]{3}” indica que a porção seguinte ao DDD pode ter três ou quatro dígitos.
- ✓ A sub-expressão “\-?” indica que o traço entre os dois blocos de quatro dígitos e opcional.
- ✓ A sub-expressão [0-9]{4} indica a concatenação entre mais quatro caracteres numéricos.

Se for necessário fazer o reconhecimento de algum dos caracteres especiais usados na definição das expressões, o caractere ‘\’ é usado para indicar que o caractere seguinte é um literal. Por exemplo: se desejarmos reconhecer o ponto de interrogação, que é um caractere especial, devemos usar ‘\?’’. Como a própria barra é um caractere especial, a expressão para seu reconhecimento tem a forma ‘\\’.

Resumidamente, as constantes e operações básicos para definição de expressões regulares no formato *Posix* podem ser encontrados no anexo 11.1. Existem outras notações para a definição de expressões regulares, porém são semelhantes, possuindo os mesmos operadores, mas com sintaxe levemente diferente.

2.2 Ferramentas para Análise Computacional de Textos

Do ponto de vista computacional a análise lingüística de um texto é realizada por etapas. Sabemos que um texto é organizado em parágrafos, que cada parágrafo é composto por orações e cada oração é composta por palavras. As palavras ainda podem ser quebradas para análise morfológica.

Cada fase requer uma ferramenta computacional diferente. Dizemos que o texto analisado é passado por um processador chamado de *parser*. Um *parser* (analisador gramatical, no contexto atual do texto) é tipicamente composto por delimitadores de sentenças (*sentence delimiters*), analisadores léxicos (*tokenizers*), *stemmers*, *chunkers* e *POS Taggers*.

Outra ferramenta necessária para o processamento de linguagem natural é o léxico (*lexicon*), um repositório de palavras da linguagem analisada. Computacionalmente falando, o

léxico é constituído de uma lista de radicais e afixos, mais as regras morfológicas e sintáticas que regem a formação das palavras da língua trabalhada.

2.2.1 Analisador Léxico

A primeira ferramenta a ser definida é o analisador léxico (*Tokenizer*), ferramenta esta que tem a função de separar um *stream* de caracteres em unidades significativas chamadas *tokens*. Um *token* pode ser uma palavra, número ou mesmo um sinal de pontuação. A definição exata de um *token* depende de implementação e dos objetivos de um programa, isto é, um sinal de pontuação pode ou não ser um *token*, dependendo do contexto da aplicação. O analisador léxico é a ferramenta básica para o processamento de textos, pois é o responsável por identificar as unidades a serem processadas dentro dos programas. O Exemplo 2 mostra uma sentença analisada lexicalmente por palavras, considerando a pontuação também como *token*

Exemplo 2: Exemplo de sentença tokenizada por palavras.

O	rato	roeu	a	roupa	do	rei	de	Roma	.
Token 1	Token 2	Token 3	Token 4	Token 5	Token 6	Token 7	Token 8	Token 9	Token 10

Em geral um analisador léxico ao processar um texto qualquer, procura por um conjunto de *tokens* predefinidos em uma seqüência ou posição esperados. Em uma análise inicial e considerando o processamento de textos em geral, os *tokens* são as unidades básicas, como as palavras, as siglas, as abreviações ou os números.

Em uma abordagem simplista, a análise léxica de um texto envolve a procura por um *string* de caracteres contendo espaços em branco (espaços, tabulação ou nova linha) ou certos sinais de pontuação e a quebra do *string* em *tokens* de palavras (adaptado de [NLTK 2004]), podendo ser feita através de expressões regulares.

2.2.2 Delimitadores de Sentenças

A identificação de orações e parágrafos requer um delimitador de sentenças. Em uma abordagem simplista um delimitador poderia trabalhar através da identificação de sinais de pontuação, porém esta abordagem é falha, pois alguns sinais de pontuação podem possuir outros significados. O ponto, por exemplo, pode indicar que a palavra imediatamente anterior esta abreviada.

2.2.3 Rotuladores

Um *tagger*, doravante denominado rotulador, é um programa capaz de associar a cada palavra de um texto, um rótulo. Um *POS Tagger (Part of Speech Tagger)* atribui a cada palavra de um texto um rótulo que adiciona informação a respeito da classe desempenhada pela palavra em um dado ponto de um texto.

As ferramentas de *POS tagger* são construídas sobre analisadores léxicos e são usados para rotular as palavras de uma sentença com a classe mais adequada, como verbo, substantivos, etc. O Exemplo 3 mostra a rotulação de uma sentença:

Exemplo 3: sentença rotulada

Nós	todos	somos	loucos	aqui.
pronome pessoal	pronome indefinido	verbo	adjetivo	advérbio

Alice no País das Maravilhas de Lewis Carroll

Segundo a documentação do NLTK [NLTK 2004], existem quatro tipos básicos de rotuladores na ferramenta. Estes quatro tipos de rotuladores servem como exemplo do funcionamento dos rotuladores em geral.

- ✓ *Default Tagger*: atribui o sempre o mesmo rótulo a um determinado *token* independentemente do texto onde o *token* está inserido. Para este tipo de rotulador, toda palavra é rotulada sendo que ele é capaz de rotular palavras que não possuía conhecimento prévio. A taxa de acertos deste tipo de rotulador não é muito alta, ficando entre 20% a 30%. Geralmente ele é usado como complemento a um rotulador mais robusto.
- ✓ Rotuladores Baseados em Expressões Regulares (*Regular Expression Tagger*): atribui um rótulo ao *token* com base em uma regra definida através de uma expressão regular. Este método pode se basear nos prefixos e sufixos de uma palavra para tentar acertar sua classe gramatical.
- ✓ Rotulador Baseado em Unigramas (*Unigram Tagger*): este tipo de rotulador atribui um rótulo a um determinado *token* baseado em informações estatísticas adquiridas previamente através do treino do algoritmo em um *Corpus* já anotado. Cada *token* recebe o rótulo que é estatisticamente mais adequado para ele no texto dado. Por exemplo, se a palavra “filtro” é usada mais frequentemente como substantivo do que como verbo no *Corpus* de treino, o rotulador tenderá a rotular a palavra “filtro” como substantivo.

- ✓ Rotuladores de Ordem N: Em adição aos Rotulador Baseado em Unigramas, alguns rotuladores trabalham com o contexto onde uma palavra esta inserida. Estes são os Rotuladores de Ordem N (*Nth order Taggers*). Por exemplo, um rotulador baseado em bigramas (*Bigram Tagger*) considera que o contexto de um *token* é dado pela palavra atual mais a anterior e a partir do contexto achado, o rotulador atribui o rótulo mais adequado à palavra. A ordem do rotulador é dada conforme o número de palavras usadas para o estabelecimento do contexto. Segundo este modelo, seríamos induzidos a pensar que quanto maior o valor de N, melhor o desempenho alcançado pelo algoritmo de rotulação, porém um valor muito alto de N pode comprometer severamente o tempo de execução além de consumir uma quantidade de memória considerável para armazenamento dos n-gramas. Normalmente costuma-se trabalhar com bigramas e trigramas, alcançando-se resultados satisfatórios.

Mesmo trabalhando com bigramas ou trigramas, o maior problema no uso de modelos baseados em n-gramas é o tamanho da tabela de n-gramas, normalmente vetores esparsos.

Uma excelente alternativa aos rotuladores baseados em n-gramas é o *Brill Tagger*, que se trata de um algoritmo baseado em aprendizado por transformações (*Transformation Based Learning*). Em uma explicação simplista, o *Brill Tagger* faz uma rotulação inicial do texto e em um segundo passo faz a correção dos rótulos através de regras. Por se tratar de um método de aprendizado supervisionado, é necessário treinar o algoritmo com dados anotados previamente.

O funcionamento do *Brill Tagger* pode ser descrito da seguinte forma: No passo onde é feita a primeira atribuição de rótulos, se a palavra é conhecida deve receber o rótulo mais frequente e se a palavra é desconhecida é rotulada como substantivo. No segundo passo, os rótulos atribuídos inicialmente são substituídos com base na análise de regras do tipo:

- ✓ Mude o rótulo A para o rótulo B [BRILL 1999]:
 - ✓ Se a palavra anterior (posterior) é rotulada como Z.
 - ✓ Se a segunda palavra antes (depois) é rotulada como Z.

Em um teste realizado com o uma porção de 5% do *Brown Corpus* foi relatada uma taxa de 5,1% de erros com a utilização do *Brill Tagger* [BRILL 1999].

2.2.4 Stemmers

Os *stemmers* são ferramentas usadas para fazer a análise morfológica das palavras através da associação de uma palavra com seu radical. Por exemplo, o verbo conjugado “fazendo” é associado ao radical “faz”. Os analisadores morfológicos não usam o contexto de uma palavra como informação, portanto não são capazes de resolver ambigüidades e identificar o radical mais adequado. A palavra “ameaça” pode ser interpretada como o gesto ou como o verbo ‘ameaçar’. Normalmente os *stemmers* removem os afixos das palavras usando regras, encontrando assim o radical das palavras. O Exemplo 4 mostra uma sentença cujas palavras tiverem o radical extraído por um *stemmer*.

Exemplo 4: Exemplo de sentença processada por um *stemmer*

Palavra:	O	rato	roeu	a	roupa	do	rei	de	roma	.
Stem:	O	rat	roe	a	roup	do	rei	de	rom	.

A idéia principal por detrás de um algoritmo de *stemming* é que uma palavra possui diversas formas, todas com um mesmo significado, isto é, podemos flexionar uma palavra através da adição de prefixos ou sufixos, mas a idéia básica representada é compartilhada pelas variações. Na verdade a flexão de uma palavra normalmente adiciona alguma informação. Por exemplo, se trocarmos a vogal “o” do fim da palavra “gato” pela vogal “a”, continuamos a nos referenciar ao animal, porém foi adicionada a informação de gênero.

Em uma definição mais formal, *Stemmer* são analisadores morfológicos que associam variações de um mesmo termo a uma raiz [Jackson 2002]. O resultado do processo de *stemming* aplicado sobre uma palavra é o *stem*. O *stem* é o que resta de uma palavra após a remoção de todos os afixos, obtidos a partir de inflexão. Neste trabalho o termo radical será utilizado para se referenciar ao *stem*.

O processamento usado em um *stemmer* é bastante utilizado e útil em RI, onde a busca pelo radical de uma palavra em uma coleção de textos pode aumentar a quantidade de documentos recuperados por um sistema. Por exemplo se uma pessoa busca por documentos referentes a felinos em um *site* de busca, mais documentos poderiam ser recuperados se a busca fosse estendida para documentos que contenha as palavras felino ou felina. A aplicação do algoritmo de *stem* poderia ampliar os resultados através da busca pelo radical da palavra, no caso “felin” e encontrar os documentos que contenham as palavras felino, felina, felinos e felinas.

Os *stemmers* podem ser baseados em dicionários ou em algoritmos, no entanto esta divisão não é totalmente inflexível, pois alguns algoritmos de *stemmer* podem se valer de minidicionários para tratamento de exceções, enquanto os *stemmers* baseados em dicionários necessitam recorrer a algoritmos para remover pelo menos alguns sufixos de forma a permitir que as entradas sejam localizadas em um dicionário. Normalmente os *stemmers* baseados em dicionários possuem um desempenho melhor com relação a obtenção dos radicais das palavras, porém costumam ser mais lentos que os *stemmers* algorítmicos. (Adaptado de [Porter 2001])

O algoritmo mais conhecido de *stemming* é denominado *Porter Stemmer*, publicado em julho de 1980 no periódico *Program* por Martin Porter. O algoritmo foi escrito para processamento da língua inglesa, existindo diversas implementações disponíveis. Como muitas das implementações continham problemas, Martin Porter resolveu estender seu trabalho desenvolvendo um conjunto de ferramentas para a criação de *stemmers* denominado *Snowball*. Além de implementar um *stemmer* para a língua inglesa, o *Snowball* traz uma coleção de *stemmers* para outras linguagens, como o alemão, o francês, o espanhol ou o português. É importante denotar que o *Snowball* é capaz de processar *strings* em diferentes *encoding* de caractere, possibilitando o trabalho com acentuação.

Neste trabalho, é utilizado o *Snowball* [Porter 2001], que é um *stemmer* puramente algorítmico, ou seja baseado em heurística ou regras, que além de apresentar a vantagem da rapidez, não requer o trabalho de montagem de um dicionário de *stems* na área médica.

Ao optar pelo uso de um *stemmer* algorítmico, devemos ter em mente que haverá uma porcentagem de radicais extraídos que não são os corretos. O evento de uma palavra ser transformada para um radical incorreto pode ser um caso de *mis-stemming*, de *under-stemming* ou de *over-stemming*. O *mis-stemming* ocorre quando é removido de uma palavra algo que parece ser um sufixo, mas não é. O *under-stemming* ocorre quando o sufixo removido deveria ser maior e o *over-stemming* ocorre quando o sufixo removido é maior do que deveria. Os erros de *stemming* podem induzir a associação da palavra processada a um radical cujo significado não é paralelo ao da palavra inicial. A questão do significado é fortemente acoplada ao contexto de um texto, isto é, palavras extraídas de um texto moderno teriam diferentes significados se encontradas em um texto muito antigo por exemplo. Isto ocorre por que a morfologia não muda na mesma velocidade que o vocabulário. (Adaptado de [Porter 2001])

2.2.5 Sintagmas Nominais e *Chunk Parsing*

Os sintagmas, em língua portuguesa, são os elementos constituintes de uma oração. São formados por um núcleo e termos que o acompanham. A caracterização de um sintagma depende da função que ele desempenha dentro de uma frase e segundo [Perini 2005] existem sintagmas nominais, adverbiais ou adjetivos. Nesta tese tem-se um maior interesse nos sintagmas nominais, portanto este será melhor explicado.

Segundo [Perini 2005], um sintagma nominal, ou SN, é um sintagma que pode ser sujeito de uma oração. Usando ainda o exemplo de Perini tem-se que:

Exemplo 5: Sintagmas nominais [Perini 2005]

Este professor	é	um neurótico
SN		SN

Apesar do sintagma “um neurótico” não desempenhar o papel de sujeito no Exemplo 5, ele pode perfeitamente servir como sujeito em uma outra oração qualquer. Nota-se que o núcleo de um sintagma nominal possui um substantivo.

Sua estrutura em nível macroscópico é constituída pela Área Esquerda, que contém os elementos que precedem o núcleo, e pela Área Direita, constituída pelo núcleo mais os elementos que o seguem. A Área Esquerda possui seis posições fixas e quatro posições variáveis. As posições fixas são o determinante, possessivo, reforço, quantificador, pré-núcleo externo (PNE) e pré-núcleo interno (PNI). As posições variáveis ocorrem no intervalo entre as posições fixas, exceto entre os dois pré-núcleos, e exercem a função de numerador (Adaptado de [Perini 2005]). O Exemplo 6 mostra a distribuição dos itens pertencentes a área esquerda.

Exemplo 6: Constituintes da área esquerda de um SN

Determinante PV4 Possessivo PV3 Reforço PV2 Quantificador PV1 Pré-núcleo externo Pré-núcleo interno
PV = Posição variável

Os itens que ocorrem na Área Esquerda podem assumir os valores dados na Tabela 4:

Tabela 4: valores possíveis dos itens da Área Esquerda de um SN (Adaptado de [Perini 2005]).

Função	Valor possível
PV	Outro, dois, ... todos os cardinais exceto um, que possui comportamento diferente
Determinante	O, este, esse, aquele, algum nenhum, um
Possessivo	Todos os pronomes possessivos
Reforço	Mesmo, próprio, certo
Quantificador	Todos os ordinais, poucos, vários, diversos, muitos, único
PNE	Classe aberta. Exemplos: Mero, pretenso, meios, suposto, velhos, novo
PNI	Mau, novo, velho, claro, grande

Obs: 1) Uma classe aberta não possui uma especificação fixa dos componentes que a compõe.
2) Os itens que podem ser PNI também podem ser PNE.

A construção de um sintagma jamais admite que itens que desempenham a mesma função sejam colocados lado a lado em uma oração. As palavras “dois” e “pouco” são incompatíveis semanticamente, não aparecendo no mesmo sintagma. Já as palavras “cada” e “todo” são incompatíveis com a maioria dos outros itens na área esquerda. As palavras “cada”, “todo” e “qualquer” jamais são PNE ou PNI (Adaptado de [Perini 2005]).

A Área Esquerda também pode contar com um pré-determinante (PDET), um termo que não pertence a estrutura do sintagma, mas que o precede. Por exemplo, tem-se:

Exemplo 7: Pré-determinante da área Esquerda.

Perini não define com tanta precisão a Área Direita, com relação a Área Esquerda, mas apesar de não possuir uma definição tão detalhada e restrita, a Área Direita basicamente possui o seguinte formato:

Exemplo 8: Área Direita de um SN (Adaptado de [Perini 2005]).

Tanto o núcleo como o ModE podem ser sintagmas menores, sendo que ModE pode ser separado do restante do sintagma por um sinal de pontuação, como por exemplo na frase “Um ataque cardíaco, fulminante”. Boa parte dos itens que podem aparecer como modificadores, também podem aparecer como PNE. Por exemplo: “Um almoço rápido” ou “Um rápido almoço” (Adaptado de [Perini 2005]).

Apesar da estrutura da Área Direita ser mais simples que a da Área Esquerda, sua análise é mais difícil, pois seus constituintes são classes abertas podendo assumir a forma de sintagmas menores.

Por fim, a definição descrita mostra a estrutura de um Sintagma Nominal Máximo, que na prática não costuma ocorrer. Na realidade o que observa-se é que entre todos os itens que formam um sintagma, a posição de cada um deles é fixa, porém a presença não é obrigatória, isto é, se temos um sintagma formado por um possessivo, um quantificador e o núcleo, o possessivo sempre se posicionará antes do quantificador, porém um sintagma formado apenas

por um quantificador e o núcleo também está correto, pois a presença do possessivo não é obrigatória.

Exemplo 9: Exemplos de sintagmas nominais

Seus Possessivo	diversos quantificador	gatos Núcleo	Diversos quantificador	gatos Núcleo	seus Possessivo	gatos Núcleo
--------------------	---------------------------	-----------------	---------------------------	-----------------	--------------------	-----------------

Tendo definido a estrutura de um sintagma, percebemos que esta não pode ser capturada por algoritmos como os analisadores léxicos ou os rotuladores, devido a sua complexidade. Para que possamos reconhecer pedaços mais complexos de textos, na maioria das vezes precisamos agrupar as estruturas simples dos *tokens* em estruturas mais trabalhadas. Este processo é denominado *Chunk Parsing*.

Normalmente os *Chunk Parsers* utilizam métodos baseados em expressões regulares para identificar os pedaços de texto de interesse, podendo considerar ou simplesmente ignorar certos trechos do texto (Adaptado de [NLTK 2004]).

Um *Chunk Parser* combina estruturas individuais, ou *chunks*, formando uma estrutura de *Chunks*. Esta estrutura é uma árvore de dois níveis e que representa todo o texto, tanto *chunks* como elementos não aglutinados em *chunks* (Adaptado de [NLTK 2004]). A estrutura de textos que passaram pelo processo de *chunking* pode ser representada através de rótulos ou árvores.

Por exemplo, um texto processado para extração de sintagmas nominais apresentaria a estrutura de *chunks* do Exemplo 10:

Exemplo 10: Estrutura de uma frase dividida em *chunks* que representam sintagmas nominais.

Usando a estrutura de *Chunk Parsing* do NLTK, para que possamos extrair os *Chunks* do Exemplo 10, é necessário definir expressões de *chunk* do tipo:

✓ Regra de *Chunk*: " $(\langle \text{NN} \rangle | \langle \text{NNS} \rangle)(\langle \text{JJ} \rangle | \langle \text{JJS} \rangle)^*$ "

✓ Regra de *Chink*: " $\langle \text{IN} \rangle | \langle . \rangle$ "

Onde as Regras de *Chunk* agrupam seqüências que devem ser consideradas como um *chunk*. As Regras de *Chink* são definidas para as expressões que não devem ser consideradas *chunks*, sejam ignoradas.

As regras são definidas em uma sintaxe semelhante a expressões regulares, no entanto os elementos reconhecidos são os rótulos sintáticos usados. No caso deste trabalho, uma adaptação dos rótulos do *Brown Corpus* [Brown 1979].

Na realidade a definição de sintagma nominal dada por [Perini 2005] é complexa e apresenta alguns pontos em aberto. Considerando a estrutura dos textos a serem processados neste trabalho, os sintagmas serão tratados de forma mais simplista, sendo que mais detalhes da extração de *chunks* e sintagmas serão dados mais adiante na descrição das implementações usadas.

2.2.6 NLTK (Natural Language Toolkit)

Desenvolvido em conjunto com o curso de lingüística computacional da Universidade da Pensilvânia, o NLTK [NLTK] é um *toolkit* escrito em linguagem *Python* composto por bibliotecas, programas e um conjunto de dados para demonstração destinados ao processamento de linguagem natural. Possui uma vasta documentação na forma de tutoriais e demonstrações.

Projetado para que estudantes pudessem aprender PLN, tendo sido usado como ferramenta educacional, também pode ser usado por pesquisadores como ferramenta de estudo e criação de protótipos de aplicações de PLN.

O NLTK define a estrutura básica utilizada para a construção de aplicações voltadas ao PLN, oferecendo [NLTK 2004]:

- ✓ Classes básicas para a representação de dados em PLN;
- ✓ Interfaces padrão para o desempenho de tarefas como tokenização, *tagging* e *parsing*;

Implementações padrão de cada tarefa que podem ser combinadas para a solução de problemas.

2.3 Recuperação de Informação (RI)

Segundo Yates [Yates 1999] a RI (Recuperação de Informação / *IR - Information Retrieval*) é: “Parte da Ciência da Computação que estuda a recuperação da informação (não de dados) a partir de uma coleção de documentos escritos. Os documentos recuperados devem satisfazer a necessidade de alguma informação por parte do usuário, que usualmente expressa a busca na forma de linguagem natural.”

Basicamente a RI deve trabalhar com a representação, o armazenamento, a organização e o acesso à informação.

2.3.1 Extração de Informação

A Extração de Informação (*Information Extraction - IE*) é uma sub-área do processamento de linguagem natural voltada a busca de informações úteis dentro de documentos, através da procura por termos de interesse em textos eletrônicos. A EI difere da RI (Recuperação de Informação) pelo fato de que em RI são feitas buscas por documentos e não por informações chave contidas em documentos.

Também é importante diferenciar a EI de uma outra área da PLN, o Entendimento de Textos (*Text Understanding - TU*). Enquanto a EI busca por informações chave que caracterizam o conteúdo do documento, a TU busca o sentido do texto como um todo.

Segundo [Hobbs 1996] as principais diferenças entre EI e TU são

Tabela 5: Comparação entre EI e TU

EI	TU
Apenas uma fração do texto é relevante.	O objetivo é entender o texto como um todo.
A informação é traduzida para uma representação relativamente simples que pode ser mapeada para campos de um banco de dados.	A representação final deve acomodar toda a complexidade da linguagem.
As variações de significado e o objetivo final do autor ao escrever o texto são secundários	É desejável reconhecer as nuances de significado e o objetivo do autor como um todo.

Tomando como exemplo o trecho de caso abaixo, se aplicássemos um algoritmo de extração de informação teríamos o resultado do Exemplo 11.

"Paciente com quadro de movimentos involuntários de início abrupto, explosivos, a maioria de curta duração repetindo-se com intensidade e topografias variáveis (migratórios) acometendo somente o hemicorpo esquerdo há cerca de 20 dias." (Luiz Eduardo Bacheschi)

Exemplo 11: Exemplo de formulário de extração de informação

Quadro:	
Descrição:	Movimentos involuntários de início abrupto, explosivos
Anatomia:	Hemicorpo esquerdo Topografia variável Migratório
Duração:	curta duração
Intensidade:	Variável
Período:	há cerca de 20 dias

Tipicamente uma aplicação de extração de informação funciona em estágios, cada um responsável por um nível diferente do processamento lingüístico. A Figura 1 representa o fluxo de processamento para a extração de informação.

Figura 1: Fluxo típico de processamento de EI [Jackson 2002]

O primeiro estágio é responsável por reconhecer as palavras e delimitar as sentenças, para então o *POS Tagger* rotular os elementos de cada sentença. O Analizador de ER reconhece padrões dependentes do domínio trabalhado para então preencher os formulários de dados que por fim são combinados em um único formulário.

A EI é avaliada de forma semelhante a RI, consideradas as devidas adaptações. O capítulo 2.4 trata com mais detalhes os métodos para avaliação da recuperação de informação.

2.4 Avaliação da Recuperação de Informação

Com relação a avaliação dos sistemas de EI e de RI tradicionalmente são usadas as medidas de recuperação (*recall*) e precisão (*precision*). A recuperação é uma medida usada para avaliar a quantidade de informação relevante recuperada por um sistema com relação à informação relevante existente na base, enquanto que a precisão é uma medida utilizada para avaliar a quantidade de informação realmente correta retornada por um sistema com relação ao total de informação recuperado pelo sistema.

Supondo que seja feita uma busca por documentos relevantes segundo um critério qualquer. Seja R a recuperação, P a precisão, N o número total de itens de uma base de dados,

n_r o número de respostas corretas dadas pelo sistema, n_t o número total de respostas dadas pelo sistema e M o número total itens relevantes na base. Tem-se que:

Equação 1: Recuperação [Jurafsky 2000].

$$R = \frac{n_r}{M}$$

Equação 2: Precisão [Jurafsky 2000].

$$P = \frac{n_r}{n_i}$$

As medidas de recuperação e precisão costumam ser antagonistas, pois se a taxa de recuperação de um sistema aumenta, a precisão tende a cair e vice-versa. Uma métrica que busca conciliar ambas as medidas é chamada de *F-Measure*.

A *F-Measure* procura balancear os valores obtidos através da recuperação e da precisão através de um parâmetro β . Quando β assume o valor 1, a recuperação e a precisão possuem um mesmo peso no cálculo da *F-Measure*. Quando β é maior que 1, a precisão adquire um peso maior e quando β é menor que 1, a recuperação obtém um peso maior na *F-Measure*. A *F-Measure* é usada quando se deseja ter uma métrica única para avaliar um sistema.

Seja F a *F-Measure*, β o parâmetro de calibração, R a recuperação e P a precisão, tem-se:

Equação 3: *F-measure* [Jurafsky 2000].

$$F \text{ Measure} = \frac{((\beta^2 + 1) * P * R)}{(\beta^2 * P + R)}$$

Uma quarta medida, o *Fall-out*, é usada para quantificar a habilidade do sistema em ignorar as informações espúrias (informações não condizentes com a busca) no conjunto de dados trabalhado. Seja n_i o número de resposta incorretas dadas pelo sistema e n_e o número de itens espúrios na coleção de dados, tem-se:

Equação 4: *fall-out* [Jurafsky 2000].

$$Fall \ out = \frac{n_i}{n_e}$$

A medidas definidas neste capítulo serão utilizadas mais adiante para a avaliação de desempenho de algumas partes da aplicação proposta para este trabalho, porém ressalta-se que

os processos de RI ou EI são complexos, não sendo esperado encontrar uma precisão e uma recuperação de 100%. Dependendo de cada aplicação um sistema pode ser calibrado segundo seus requisitos, buscando-se as vezes melhor recuperação ou então melhor precisão. Por exemplo, em um sistema de busca por conhecimento médico, poderíamos calibrar o sistema para obter uma maior precisão e uma recuperação menor (ênfase na qualidade dos documentos recuperados), visto que a precisão é importante em medicina..

As medidas de recuperação, precisão e *F-Measure* são interessantes para avaliar a capacidade de um sistema de recuperar itens relevantes contidos em um conjunto de itens ou em um texto, porém elas não oferecem uma forma de avaliar a situação onde não há itens relevantes em uma base ou coleção de dados, situação esta que resulta em valores zerados de recuperação e precisão, mas que não significa um desempenho ruim do sistema avaliado. A medida de *fall-out*, também falha nesta situação, pois ela fornece a probabilidade de encontrar um item irrelevante entre os itens recuperados, se nada foi recuperado, como distinguir a ausência do item procurado da falha na recuperação?

Para complementar a avaliação através das medidas discutidas, doravante denominadas de medidas tradicionais, é utilizado neste trabalho um método adicional baseado na Curva ROC, discutido no próximo item.

2.4.1 Receiver Operating Characteristics (ROC)

Tendo sua origem em telecomunicações e detecção de sinais, a *Receiver Operating Characteristics* (ROC) também denominada *Receiver Operating Curve*, é bastante utilizada em medicina para avaliar resultados de diagnósticos de forma binária em um conjunto de dados.

A curva ROC é a representação gráfica da sensibilidade e da especificidade de um teste. A curva é construída em termos da fração de verdadeiros positivos pela a fração de falsos positivos, conforme a variação de um *threshold* de decisão, para a obtenção de frações diferentes de decisão em um intervalo de zero a um (adaptado de [Rangayyan 2005]).

Um ponto interessante no uso da curva ROC para avaliação de um teste, é que o resultado da análise não depende simplesmente da qualidade do teste, mas também da definição do que é tido como bom ou ruim.

Como exemplo do processo de avaliação através da área sob a curva ROC, considere a realização de uma pesquisa para recuperar um conjunto de informação em uma base de documentos, onde: A representa o evento de um determinado item ser relevante e N o evento de um determinado item não ser relevante.

Sejam as probabilidades $P(A)$ e $P(N)$ representantes das frações dos itens relevantes e não relevantes na população de teste, respectivamente. Seja T^+ o representante de um resultado relevante e T^- o representante de um resultado irrelevante.

O primeiro passo para a obtenção dos parâmetros para a montagem da curva ROC é a classificação dos dados segundo os seguintes rótulos:

- ✓ VP (Verdadeiro Positivo): quando um item é julgado relevante;
- ✓ VN (Verdadeiro Negativo): Quando um item é julgado irrelevante;
- ✓ FP (Falso Positivo): Quando um item é julgado relevante, mas na verdade é irrelevante;
- ✓ FN (Falso Negativo): Quando um item é julgado irrelevante, mas na realidade é relevante;

Classificados os itens envolvidos no teste, podemos então calcular a Sensibilidade S^+ (*Sensitivity*) ou a FVP (Fração de Verdadeiros Positivos / *True Positive Fraction - TPF*), que representa a capacidade de um teste em identificar itens relevantes e é dada por $P(T^+ | A)$. Podemos calcular a sensibilidade através da fórmula dada :

Equação 5: Sensibilidade de um teste (FVP ou S^+) [Rangayan 2005].

$$FVP = \frac{(N^{\circ} \text{ de VP})}{(N^{\circ} \text{ itens relevantes})}$$

A capacidade de um teste em indicar a ausência de um item relevante é dada pela especificidade S^- (*Specificity*) ou Fração de Verdadeiros Negativos (FVN / *True Negative Fraction - TNF*). A especificidade é dada por $P(T^- | N)$. Podemos calcular a especificidade através da Equação 6.

Equação 6: Especificidade de um teste (FVN ou S^-) [Rangayyan 2005].

$$FVN = \frac{(N^{\circ} \text{ VN})}{(N^{\circ} \text{ de itens irrelevantes})}$$

A probabilidade da perda de um item relevante é denominada de FFN (Fração de verdadeiros Negativos / *False Negative Fraction*) e é dada por $P(T^- | A)$. A probabilidade de um item irrelevante ser considerado relevante é denominada FFP (Fração de Falso Positivo / *False Positive Fraction*) e é dada por $P(T^+ | N)$. Temos que:

Equação 7: Probabilidade de um item relevante não ser encontrado [Rangayyan 2005].

$$FFN = 1 - FVP$$

Equação 8: Probabilidade de um item irrelevante ser considerado relevante [Rangayyan 2005].

$$FFP = 1 - FVN$$

Em complemento temos a medida da acurácia (*accuracy*), que é a probabilidade de um item irrelevante ser considerado relevante e é dada por:

Equação 9: Probabilidade de um item irrelevante ser considerado relevante [Rangayyan 2005].

$$Acurácia = FVP * P(A) + FVN * P(N)$$

Para indicar a eficiência de um teste, podemos ainda usar as medidas de PPV (*Positive Predictive Value*), que representa a porcentagem de itens rotulados como relevantes em um teste e que são realmente relevantes. O valor de NPV (*Negative Predictive Value*) a porcentagem de casos rotulados como negativos e que são verdadeiramente negativos. Os valores de PPV e NPV são dados por:

Equação 10: PPV [Rangayyan 2005].

$$PPV = 100 * VP$$

Equação 11: NPV [Rangayyan 2005].

$$NPV = 100 * \left(\frac{VN}{VN + FN} \right)$$

Por fim, a probabilidade de um teste rotulado como positivo ser realmente relevante é dada pela probabilidade condicional $P(A | T^+)$ e pode ser encontrada através da regra de Bayes:

Equação 12: $P(A | T^+)$ [Rangayyan 2005].

$$P(A | T^+) = \frac{P(A) * FVP}{P(A) * FVP + P(N) * (1 - FVN)}$$

2.5 Alguns Trabalhos em Processamento de Linguagem Natural e Extração de Informação

Para situar o contexto deste trabalho, segue a descrição de alguns trabalhos e eventos importantes na área de extração de informação e processamento de textos médicos.

2.5.1 MUC (*Message Understanding Conferences*)

No fim da década de 80 e durante a década de 90 a DARPA (*Defense Advanced Research Project Agency*) realizou uma série de sete *workshops* chamados de MUC (*Message Understanding Conferences*), onde os participantes recebiam uma tarefa preestabelecida e deveriam projetar e construir sistemas para a extração de informação segundo a tarefa dada, sendo avaliados por desempenho. A última conferência (MUC-7) ocorreu em 1998.

2.5.2 Fastus (*Finite State Automaton Text Understanding System*)

O nome FASTUS é um acrônimo para *Finite State Automaton Text Understanding System*, sendo seu objetivo extrair informação de um texto escrito em linguagem natural (inglês ou japonês) de forma a gerar entradas para um banco de dados ou outra aplicação qualquer.

Trata-se de um do projeto que participou da maioria das conferências do MUC, evoluindo até a obtenção de bons resultados em extração de informação. Nos testes do MUC-4, que tratava a extração de informação sobre terrorismo na América Latina, em uma coleção de 100 textos obteve 44% de recuperação e 55% de precisão. No MUC-6 obteve 74% de recuperação e 76% de precisão na tarefa de preenchimento de um formulário, mas obteve 92% de recuperação e 96% de precisão na tarefa de reconhecimento de entidades nomeadas (*named entity recognition*). O reconhecimento de entidades nomeadas é a tarefa de identificação de nomes próprios, como nomes de empresas, pessoas, lugares, etc.

2.5.3 MedLEE (*Medical Language Extraction and Encoding System*)

O MedLEE (*Medical Language Extraction and Encoding System*) é um sistema usado para analisar e codificar relatórios clínicos. O sistema é composto por diferentes módulos responsáveis por processar e transformar textos em inglês segundo um aspecto particular da linguagem até obter uma saída estruturada. [Mendonça 2000]

O sistema gera saídas no formato XML e usa um léxico desenvolvido manualmente baseado em relatórios clínicos, apesar de testes com o vocabulário UMLS terem sido realizados.

2.6 Vocabulários Médicos e Ontologias

O trabalho apresentado tem uma forte dependência da existência de sistemas de terminologias e ontologias em medicina. Esta seção tem por objetivo introduzir alguns conceitos sobre ontologias e dar uma visão geral de alguns dos vocabulários e sistemas existentes atualmente. Existem muitos outros vocabulários e sistemas de terminologia, no entanto foi dada atenção aos mais relevantes, do ponto de vista deste trabalho. Mais adiante será descrito detalhadamente a estrutura dos dois sistemas utilizados neste trabalho, o UMLS e o RadLex.

2.6.1 Ontologias

Em nosso dia a dia temos a necessidade constante de comunicação em qualquer ambiente, seja ele profissional ou não. Especialmente em ambientes profissionais a comunicação clara e objetiva é importante para o bom andamento dos processos envolvidos em cada tarefa executada por uma equipe.

A comunicação requer que as pessoas compartilhem um mesmo conjunto de terminologias e que essas terminologias sejam claras e precisas o suficiente de forma a evitar ambigüidades e a má interpretação, havendo a necessidade não apenas de um vocabulário comum, mas também por um contexto onde tal vocabulário esteja inserido. Essa necessidade nos guia ao estudo das Ontologias.

Segundo [Noy 2001] uma ontologia é uma especificação formal explícita dos termos pertencentes a um domínio e as relações entre eles, isto é, trata-se de um conjunto de abstrações usadas para comunicação envolvendo a definição do que desejamos representar, quais as propriedades envolvidas em cada entidade representada e qual o relacionamento entre as entidades representadas. As ontologias além de descreverem o conhecimento de forma estruturada também são constituídas por uma base de conhecimento.

Definida uma ontologia, esta pode ser reutilizada e adaptada segundo as necessidades da aplicação ou processo envolvido. Devemos ter em mente que uma ontologia deve ser

reavaliada constantemente, uma vez que o mundo não é constante e a ontologia não passa de uma abstração da parte do mundo de interesse em uma determinada situação.

Do ponto de vista de aplicações voltadas ao processamento de linguagem natural, as ontologias são importantes na definição dos objetos de interesse para um determinado escopo, bem como para a estruturação dos léxicos com suas regras, categorias lingüísticas e do papel de cada objeto.

No que diz respeito a medicina, as ontologias são muito usadas para a definição de vocabulários padronizados para alimentação de sistemas como prontuários médicos, para a formatação de laudos ou para a indexação de artigos científicos, por exemplo. Um requisito importante em medicina é a clareza da informação, fator que torna importante a utilização das ontologias.

Neste trabalho usamos dois sistemas de ontologias médicos, o UMLS (*Unified Medical Language System*) [UMLS] e o RadLex (*RadLex Lexicon*) [RadLex]. Ambos os sistemas de ontologias tem o propósito de fornecer meios para que sistemas computacionais possam capturar, indexar e recuperar informação médica, porém o segundo trata apenas da área de radiologia.

O UMLS trata-se de uma ontologia ampla, que abrange diversos vocabulários médicos em diversas áreas da medicina. Seu tamanho a torna uma ontologia genérica, no entanto útil para aplicações de indexação de artigos.

O RadLex é uma ontologia voltada a radiologia, projetada para acomodar as particularidades da área em questão, sendo muito mais útil para aplicações focadas nesta área da medicina.

A modelagem de ontologias está ligada a área de Representação de Conhecimento (*Knowledge Representation* ou KR), que estuda os métodos necessários para descrever o conhecimento de forma a possibilitar o processamento computacional (Adaptado de [Farquhar 1997]). Em linhas gerais a Representação de Conhecimento (RC) deve fornecer métodos e ferramentas formais para a organização do conhecimento segundo domínios de interesse.

Segundo [Randall 1993] a representação de conhecimento pode ser feita através de lógica ou de forma comportamental através da aproximação de uma metodologia humana de

solução de problemas, onde estes são quebrados em objetivos, planejamento e algumas estruturas mentais complexas.

Ao criar uma ontologia, definimos os termos e os relacionamentos entre eles segundo um determinado domínio, em uma estrutura hierárquica. Os conceitos são descritos através de classes dispostas em uma hierarquia de classes e subclasses, onde cada classe é descrita por atributos, os *slots*, e um conjunto de restrições impostas ao valor dos atributos. Além da estrutura hierárquica de classes, uma ontologia agrega também uma base de conhecimento, representada pelos *slots* preenchidos.

Ao definir uma ontologia devemos ter em mente um conjunto de possíveis perguntas e respostas buscadas. Em analogia as metodologias de projeto orientado a objetos, a definição do conjunto de perguntas e respostas serve como um caso de uso para a criação da ontologia.

Segundo a metodologia usada na ferramenta *Protégé*, ferramenta esta utilizada neste trabalho, é necessária a descrição de alguns conceitos que servem para a definição de uma ontologia. A Tabela 6 descreve esses conceitos:

Tabela 6: Conceitos utilizados na definição de ontologias.

Classe	Uma classe descreve um conjunto de elementos semelhantes entre si.
Instância	Uma instância é um membro de uma classe, isto é, um elemento materializado que é identificado como possuindo as características necessárias para pertencer a um dado conjunto de elementos (classe).
Subclasse	Uma subclasse é uma especialização de uma classe, descrevendo conceitos mais específicos que a super-classe. Se uma classe é um conjunto de elementos, uma subclasse é um subconjunto desses elementos.
Meta-classe	Uma meta-classe é classe cujas instâncias são classes.
Slots (roles ou properties)	Os <i>slots</i> na definição de uma classe descrevem os atributos e relacionamentos de uma instância da classe. Os <i>slots</i> são herdados das super-classes pelas subclasses.
Meta-slot	Especificação de um <i>slot</i> e suas propriedades.
Sub-slot	Assim como em classes o conceito de sub-slots permite a especialização de um <i>slot</i> .
Facet ou restrições de propriedade	Descreve ou delimita o conjunto de possíveis valores de um <i>slot</i> (Ex: cardinalidade, tipo de valor ou domínio de valores). As <i>facets</i> de um <i>slot</i> podem ser sobrescritas nas subclasses de uma classe.
Inverse slot	Possibilitam a criação de relacionamentos recíprocos entre <i>slots</i> , isto é, possibilitam a representação de uma informação em dois sentidos. Representam uma informação de forma redundante acrescentando verificações adicionais e habilitando a aquisição de uma forma bidirecional.
Frame	Um <i>frame</i> representa um conceito, podendo conter <i>frames</i> instanciados (<i>Instance Frame</i>) que representam instâncias em particular de um conceito.

O ultimo item da Tabela 6 fala sobre o conceito de *Frame*. A Teoria dos *Frames* tem origem em um artigo de Minsky [Minsky 1974], onde ele descreve um *frame* como a representação de uma situação estereotipada.

Para as pessoas familiarizadas com a Programação Orientada a Objetos (POO), a estrutura de representação baseada em *Frames* é bastante familiar, apesar de existirem

algumas diferenças sutis, mesmo assim o conhecimento em modelagem orientada a objetos pode ser de grande valia para o aprendizado de modelagem de ontologias baseada em *Frames*.

Em orientação a objetos, as classes descrevem o estado e o comportamento de um conjunto de objetos bem como o relacionamento entre as classes. Em modelagem de ontologias uma classe apenas descreve um conceito com suas propriedades, restrições e relacionamentos entre classes e instâncias. Normalmente vemos ontologias modelando hierarquias de conceitos.

Classes e objetos em POO constituem estruturas de dados dentro de programas. Em ontologias as classes são o modelo de um conceito real e não uma estrutura de dados diretamente, entretanto as classes de uma ontologia podem ser mapeadas para uma estrutura de dados.

Com relação ao processo de criação de uma ontologia, não se pode afirmar que exista um único processo ou mesmo um melhor processo de desenvolvimento. A modelagem de uma ontologia é um processo criativo e pessoal, isto é, se duas pessoas diferentes desenvolverem ontologias para um mesmo domínio, estas não serão necessariamente iguais. O resultado final depende muito do conhecimento, da habilidade e da visão que o projetista tem sobre o tema modelado. A melhor forma de determinar a qualidade de uma ontologia acaba por ser a utilização da mesma (adaptado de [Noy 2001]).

Apesar de não existirem fórmulas únicas para a criação de ontologias, existem diversas ferramentas voltadas a facilitar seu projeto. Entre as ferramentas citamos o Protégé [Protégé] ou o Ontolingua [Farquhar 1996], entre outros. Neste trabalho optou-se por trabalhar com a ferramenta Protégé.

2.6.2 Protégé

O Protégé é uma ferramenta integrada usada por desenvolvedores de sistemas e especialistas para desenvolver Sistemas Baseados em Conhecimento (*Knowledge Based Systems*) [Protégé].

A ferramenta permite a definição de forma gráfica dos esquemas de classes com seus respectivos *slots* e relacionamentos bem como armazenar as instâncias das classes definidas.

Segundo a documentação da ferramenta, basicamente ela permite [Protégé]:

- ✓ A modelagem de uma ontologia de classes sobre um assunto em particular;
- ✓ A criação de uma ferramenta de aquisição de conhecimento;
- ✓ O cadastro de instâncias específicas de dados e a criação de uma base de conhecimento.

A modelagem do conhecimento através do Protégé é feita em termos de classes, subclasses, meta-classes, *slots*, *meta-slots*, *facets*, instâncias e relacionamentos. Associadas ao esquema de uma ontologia existem no Protégé telas e elementos gráficos que permitem a edição do conteúdo da base de conhecimento associada ao modelo da ontologia e a criação de instâncias. Desta forma a ferramenta permite de forma integrada modelar, preencher e fazer buscas na base de conhecimentos de uma ontologia.

2.7 Sistemas de Terminologias

2.7.1 SNOMED CT (SNOMED *Clinical Terms*)

O SNOMED CT é um sistema de terminologias médicas multilíngue que deve servir como referência para profissionais de saúde. Inclui a definição e a descrição de termos clínicos, além de apresentar uma rede de relacionamentos entre termos. Sua estrutura permite o mapeamento para outros sistemas de terminologias. (Adaptado de [SNOMED CT])

O SNOMED CT combina o conteúdo do SNOMED RT (SNOMED *Reference Terminology*) com o sistema utilizado no Reino Unido denominado *Clinical Terms Version 3*. É composto por cerca de 366.170 conceitos organizados de forma hierárquica. (Adaptado de [SNOMED CT])

2.7.2 MeSh (Medical Subject Headings)

O MESH é um vocabulário controlado criado pela NLM (*National Library Of Medicine*) usado para indexar documentos referentes à área de Biomedicina e saúde. Assim como os demais vocabulários e sistemas de terminologias em saúde, o MESH categoriza uma lista de termos, dispondo-os de forma hierárquica. (Mais detalhes em [MESH])

2.7.3 UMLS (Unified Medical Language System)

Por décadas os sistemas de informação voltados para a área médica armazenam todo tipo de dados, como dados de pacientes, resultados e laudos, casos de estudos ou mesmo fontes literárias, no entanto a organização de tal conteúdo depende da existência de terminologias e nomenclaturas que descrevam esses dados com toda sua complexidade semântica.

Mesmo antes da utilização dos computadores na medicina, já era comum o uso de terminologias para descrição de informações médicas. A CID (Classificação Internacional de Doenças), teve sua primeira publicação no ano de 1900[MUCHMORE 2001].

A utilização de sistemas de informação na área de saúde é altamente dependente da existência de tais terminologias, que são vitais ao processo de organização e recuperação de informação. Atualmente existem diversos esforços para a criação de terminologias, muitos destes esforços chegam a ser redundantes, mas citamos como exemplos de padrões de terminologias o SNOMED CT, o MESH e o RadLex.

A existência de diversos sistemas de terminologias motivou a NLM (*National Library of Medicine*) a iniciar a produção do UMLS (*Unified Medical Language Systems*), um padrão que agrupa diversos sistemas de terminologias em um único sistema destinado ao uso em formato eletrônico.

Mais especificamente, o UMLS é uma coletânea de conceitos médicos mais uma coleção de ferramentas, construído com o objetivo de facilitar o desenvolvimento de sistemas que fazem uso de vocabulários de Biomedicina e saúde. O UMLS foi concebido se pensando em sistemas que armazenam, processam, recuperam e integram informações no formato eletrônico. (Adaptado de [UMLS 2004AB])

É formado por uma compilação de diversos sistemas de terminologias em diversas línguas, mantendo-se os relacionamentos, classificações e códigos das várias fontes, porém adicionando uma organização e classificação própria independente das fontes. Segundo esta estrutura, o UMLS não representa uma visão única de autoria da NLM, mas sim um agrupamento das visões contidas nas diversas terminologias armazenadas em sua estrutura de dados.

Como o UMLS agrega conteúdo de diversas fontes, seu uso é restrito a concessão de uso de cada uma delas, porém esta restrição é flexível visto que o usuário pode obter as licenças de cada vocabulário de forma independente. Existem dentro do UMLS vocabulários de uso livre, vocabulários cujo uso em pesquisa é livre e vocabulários de licença paga. Para uma melhor compreensão da licença de uso do UMLS consultar o apêndice A (*License Agreement For Use Of The Umls® Products*) da documentação do UMLS.

Produzido e distribuído pela NLM (*National Library of Medicine*), o pacote completo do UMLS pode ser obtido no *site* da própria NLM, mediante registro gratuito prévio. Os indivíduos registrados devem submeter anualmente um formulário de relatório de uso do UMLS.

O UMLS está organizado em fontes e ferramentas, havendo três grandes módulos: O *Metathesaurus*, o *Semantic Network*, e o *SPECIALIST Lexicon*. Os itens subseqüentes descrevem mais detalhadamente cada módulo do UMLS.

2.7.3.1 UMLS Metathesaurus

O *Metathesaurus* é uma coletânea multilíngue de diferentes terminologias, contendo informações a respeito de conceitos, relacionamentos e nomes em Biomedicina e saúde. Abriga em sua base de dados diversos vocabulários, classificações, codificações e listas de termos usados nos mais diversos serviços em saúde e indexação bibliográfica. Cada vocabulário fonte pode apresentar diversas versões diferentes, todas representadas dentro do *Metathesaurus*.

As diversas terminologias e o próprio *Metathesaurus* podem ser usadas nos mais diversos propósitos, como prontuário eletrônico do paciente, classificação e indexação de documentos eletrônicos, relatórios estatísticos, sistemas de tradução automática, etc.

Organizado através de conceitos, ou significados, o *Metathesaurus* estabelece conexões entre os diversos nomes e visões que um mesmo conceito pode assumir bem como os relacionamentos entre conceitos diferentes (Adaptado de [UMLS 2004AB]). Um conceito não é formado necessariamente por uma única palavra, mas sim por uma expressão que agregue um significado na área de saúde.

Todos os conceitos armazenados preservam a classificação, os relacionamentos e a codificação dos vocabulários fonte. Quando um conceito apresenta mais de uma

representação, todas as representações são armazenadas e mapeadas para o mesmo código de conceito do UMLS, mesmo que as diferentes representações tenham origem em fontes diferentes.

Cada conceito armazenado no *Metathesaurus* é mapeado para pelo menos um tipo semântico descrito na *Semantic Network*, o que proporciona aos conceitos uma categorização consistente, ainda que em um nível geral (Adaptado de [UMLS 2004AB]).

Originalmente o *Metathesaurus* foi concebido para trabalhar com vocabulários da língua inglesa, o que não impede a incorporação de traduções para outras línguas em sua base de dados. No entanto a quantidade de conceitos em inglês é maior que as demais linguagens, sendo que algumas partes da estrutura de dados do *Metathesaurus* só existem para a língua inglesa, como por exemplo, o *Índice Normalizado de Strings*, que será descrito mais adiante.

2.7.3.1.1 Organização dos Conceitos

Como já dito anteriormente, o *Metathesaurus* é organizado por conceitos, sendo que cada conceito representa um significado. Cada conceito recebe um código único denominado CUI (*Concept Unique Identifier*), um nome e características como linguagem, vocabulário fonte e tipo de nome.

Além do nome principal que o conceito recebe, pode haver outros nomes, pois cada significado pode possuir diversas formas diferentes de representação. Por exemplo: O conceito C0004238 usado para *Atrial Fibrillation* também pode ser mapeado para as seguintes representações, todas com o mesmo significado:

Atrial Fibrillations

Auricular Fibrillation

Auricular Fibrillations

A representação *Atrial Fibrillation* é dita preferencial.

O identificador CUI é permanente, sendo que um CUI é removido apenas quando é encontrado um sinônimo não conhecido, isto é, quando dois CUIs diferentes mapeiam o mesmo conceito. Os CUIs removidos não são reutilizados, sendo possível os mapear para o identificador que foi mantido no *Metathesaurus*.

Cada nome de conceito ou *String*, independentemente da linguagem ou do vocabulário fonte, recebe um identificador único e permanente denominado SUI (*String Unique Identifier*). Qualquer variação de codificação dos caracteres (*character set*), maiúscula ou minúscula, pontuação, gênero (masculino ou feminino) ou mesmo linguagem, requer a atribuição de um novo SUI. Isto significa que o mesmo conceito, em diferentes linguagens, recebe identificadores de *string* diferentes para cada linguagem.

Apenas para a língua inglesa existe um terceiro atributo, o LUI (*Lexical Unique Identifier*). É usado para classificar as variações léxicas de um mesmo *string*. No *Metathesaurus* o conjunto de variações léxicas de um mesmo *String* é denominado termo (*term*). O valor do LUI é único para cada termo identificado.

A partir das versões lançadas no ano de 2004, o bloco básico do UMLS passou a ser o AUI, *Atom Unique Identifier*. Os *Atoms* são os nomes dos conceitos ou *Strings* contidos em cada vocabulário fonte. Se um mesmo *string* aparecer em vocabulários diferentes, cada ocorrência recebe um AUI diferente. Se um mesmo *string* aparecer no mesmo vocabulário como um nome e um nome alternativo de um conceito, por exemplo, cada ocorrência recebe um AUI único. Assim como o caso do CUI, LUI e SUI, o AUI atribuído a um *Atom* é único.

Os identificadores definidos no *Metathesaurus* relacionam-se da seguinte forma: cada CUI categoriza diversos LUI, porém cada LUI está ligado a um único CUI. Cada LUI categoriza diversos SUI, porém cada SUI está ligado a apenas um LUI. Cada SUI categoriza diversos AUI, porém cada AUI está ligado a apenas um SUI. A Figura 2 representa o relacionamento entre os identificadores usados no *Metathesaurus*, enquanto a Tabela 7 exemplifica os relacionamentos descrito na Figura 2.

Tabela 7: Representação dos relacionamentos entre identificadores no *Metathesaurus* ([UMLS 2004AB])

Concept (CUI)	Terms (LUIs)	Strings (SUIs)	Atoms (AUIs)
C0004238 Atrial Fibrillation (preferred) Atrial Fibrillations Auricular Fibrillation Auricular Fibrillations	L0004238 Atrial Fibrillation (preferred) Atrial Fibrillations	S0016668 Atrial Fibrillation (preferred)	A0027665 Atrial Fibrillation (from MSH) A0027667 Atrial Fibrillation (from PSY)
		S0016669 Atrial Fibrillations	A0027668 Atrial Fibrillations (from MSH)
	L0004327 (synonym) Auricular Fibrillation Auricular Fibrillations	S0016899 Auricular Fibrillation (preferred)	A0027930 Auricular Fibrillation (from PSY)
		S0016900 (plural variant) Auricular Fibrillations	A0027932 Auricular Fibrillations (from MSH)

Figura 2: Diagrama de classes representando os identificadores do *Metathesaurus*.

2.7.3.2 Índices e Índices Normalizados

Existem três índices no *Metathesaurus*: o Índice de Palavras (*Word Index*), o Índice de Palavras Normalizadas (*Normalized Word Index*) e o Índice de Strings Normalizados (*Normalized String Index*).

O **Índice de Palavras** conecta cada palavra individual existente no *Metathesaurus* a todos os *strings*, termos e conceitos onde a palavra aparece. Cada linguagem possui um Índice de Palavras, sendo cada índice armazenado em uma tabela de banco diferente (Adaptado de [UMLS 2004AB]).

O **Índice de Palavras Normalizadas** conecta cada palavra normalizada em inglês a todos os *strings*, termos e conceitos onde a palavra aparece. Este índice existe apenas para a língua inglesa.

A normalização de palavras é o processo de transformá-las em letras minúsculas, reverter a flexão das palavras através do uso do SPECIALIST lexicon, caso a palavra apareça no SPECIALIST lexicon, ou gerar através de algoritmos caso contrário (Adaptado de [UMLS 2004AB]).

O **Índice de Strings Normalizados** conecta a forma normalizada de um *string* do *Metathesaurus* a cada *Strings*, termo e conceitos relacionado. Este índice existe apenas para a língua inglesa.

A normalização de um *string* envolve sua quebra nas palavras constituintes, passadas para formato minúsculo e não flexionado, sendo as palavras colocadas em ordem alfabética. As formas não flexionadas são obtidas através do SPECIALIST lexicon ou geradas através de algoritmos (Adaptado de [UMLS 2004AB]).

2.7.3.3 *Semantic Network*

A Rede Semântica (*Semantic Network*) do UMLS consiste em um conjunto de categorias dispostas segundo seu assunto ou Tipos Semânticos (*Semantic Types*) [UMLS 2004AB]. A organização de Rede Semântica do UMLS proporciona um conjunto de relacionamentos entre os Tipos Semânticos e a categorização dos conceitos existentes no *Metathesaurus*.

Está organizada de forma hierárquica (em forma de árvore) onde os Tipos Semânticos são os nós de uma rede ligada através dos relacionamentos. Ao todo existem 154 Tipos Semânticos e 54 relacionamentos na Rede Semântica do UMLS. Os Tipos Semânticos do UMLS são genéricos o suficiente para permitir a categorização de diversas terminologias em diversos domínios de conhecimento na área médica.

Todo conceito existente no *Metathesaurus* recebe pelo menos um tipo semântico, mesmo que sendo em um nível maior de granulosidade, o que faz com que o nível de granulosidade não seja constante na rede. Por exemplo, o conceito “Macaco” recebe o tipo semântico “Mamífero” pois não existe na rede a definição do conceito “Primata” (adaptado de [UMLS 2004AB]).

O principal tipo de relacionamento na Rede Semântica é do tipo “is a”. Este tipo de relacionamento define a hierarquia semântica. A porção da rede ligada através do relacionamento do tipo “is a” esta claramente organizada na forma de classes e subclasses. Apesar desta organização, a herança de relacionamentos entre classes e subclasses nem sempre é aplicada, sendo a herança de alguns relacionamentos bloqueada explicitamente para todos os filhos de uma classe.

2.7.3.4 *Organização dos Arquivos/Tabelas*

Para maiores informações sobre a organização de arquivos e tabelas do conjunto de dados do UMLS consultar o anexo 11.3.

2.7.3.5 *SPECIALIST Lexicon*

O *Specialist Lexicon* foi desenvolvido para disponibilizar as informações léxicas necessárias aos sistemas de Processamento de Linguagem Natural em língua inglesa, cobrindo palavras comuns do vocabulário inglês assim como termos de Biomedicina encontrados na coleção de texto do NLM e no *Metathesaurus*.

O léxico consiste em um conjunto de entradas para cada variação ortográfica de um *part of speech*, sendo que um item léxico pode ser composto por mais de uma palavra. Expansões de acrônimos e abreviações são permitidos como termos compostos.

Os termos e palavras contidos no léxico foram extraídos de varias fontes, como *UMLS Test Collection of MEDLINE abstracts*, *UMLS Metathesaurus* ou o *Dorland's Illustrated Medical Dictionary*. Este conjunto de cerca de vinte mil palavras forma o corpo básico de palavras do léxico. De forma complementar foram adicionadas cerca de dez mil palavras mais freqüentes do vocabulário inglês extraídas do *The American Heritage Word Frequency Book* e cerca de duas mil palavras do *Longman's Dictionary of Contemporary English*. Como a maioria dos termos obtidos são substantivos, há um esforço para identificar adjetivos e verbos a partir do *Computer Usable Oxford Advanced Learner's Dictionary* e do *Dorland's Illustrated Medical Dictionary* em adição ao uso de heurísticas. (adaptado de [UMLS LEXICON 2000]).

Para mais informações sobre a estrutura do *Specialist Lexicon* consultar [UMLS LEXICON 2000].

2.7.3.6 O UMLS em português

A estrutura do *Metathesaurus* é basicamente construída para abrigar um vocabulário da língua inglesa, o que resulta em um maior número de *strings* e conceitos codificados nesta língua e na falta de algumas porções da base de dados, como o Índice de Palavras Normalizadas e o Índice de *Strings* Normalizados.

A falta de ambos os índices pode levar a uma queda de desempenho nas consultas ao banco, pois se torna necessário pré-processar os *strings* para que em tempo de execução as comparações entre *strings* sejam mais eficientes.

Uma solução seria normalizar os *strings* de interesse, no nosso caso os *strings* em português, para criar uma nova estrutura de dados contendo esses *strings* em forma mais adequada para processamento, contudo esse seria um índice de *strings* normalizados não oficial ao vocabulário do UMLS e que deveria ser processado a cada instalação do vocabulário. Também deve ser considerado que a normalização dos *strings* em inglês utiliza o SPECIALIST lexicon, que não existe em Português, restando apenas a opção por algoritmos não adotados oficialmente pelo UMLS.

Na realidade qualquer adição seja de novos conceitos ou de processos sobre os dados já existentes na base não são oficiais, o que implica na utilização de material que não é padronizado pelo sistema do UMLS, o que pode se tornar bastante desvantajoso, principalmente se uma aplicação envolva a integração entre sistemas.

Comparativamente com o vocabulário em língua inglesa, o vocabulário em Português é bastante reduzido, mas mesmo em vocabulários extensos, por maior e mais complexo que sejam, é muito difícil cobrir todas as formas que um ser humano pode utilizar para expressar conceitos, mesmo que somente na área de medicina.

O sistema de terminologias do UMLS, apesar de agrupar diversos vocabulários continua sendo uma abstração da realidade, portanto incompleto. Percebemos este fato quando tentamos encontrar no vocabulário conceitos específicos em algumas áreas da medicina, isto é, não é viável a intenção de criar uma ferramenta genérica que sirva a qualquer tipo de aplicação.

2.7.3.7 Estatísticas

Os números fornecidos nesta seção foram obtidos na instalação do UMLS utilizada neste trabalho, cuja versão é a 2005AA e apenas com os vocabulários de uso livre e para pesquisa (licença categoria 2). Os vocabulários utilizados foram:

Tabela 8: Vocabulários utilizados

Idioma	Sigla	Versão	Descrição
Port.	ICPCPOR	1993	ICPC, Portuguese Translation, 1993
Port.	WHOPOR	1997	WHOART, Portuguese Translation, 1997
Port.	MDRPOR	7.1	Medical Dictionary for Regulatory Activities Terminology (MedDRA), Portuguese Edition, 7.1
Port.	MSHPOR	2005	Descritores em Ciências da Saúde (Portuguese translation of the Medical Subject Headings), 2005
Ing.	AIR	1993	AI/RHEUM, 1993
Ing.	CST	1995	COSTART, 1995
Ing.	DXP	1994	DXplain, 1994
Ing.	LCH	1990	Library of Congress Subject Headings, 1990
Ing.	MCM	1992	McMaster University Epidemiology Terms, 1992
Ing.	WHO	1997	WHO Adverse Reaction Terminology, 1997
Ing.	ICPC	1993	International Classification of Primary Care, 1993
Ing.	QMR	1996	Quick Medical Reference (QMR), 1996
Ing.	AOD	2000	Alcohol and Other Drug Thesaurus, 2000
Ing.	HL7	1998-2002	Health Level Seven Vocabulary, 1998-2002
Ing.	RAM	1999	QMR clinically related terms from Randolph A. Miller, 1999
Ing.	MTHMST	2001	<i>Metathesaurus</i> Version of Minimal Standard Terminology Digestive Endoscopy, 2001
Ing.	COSTAR	1989-1995	COSTAR, 1989-1995
Ing.	VANDF	03	Veterans Health Administration National Drug File, 03
Ing.	UWDA	1.7.3	University of Washington Digital Anatomist, 1.7.3
Ing.	CCS	2003	Clinical Classifications Software, 2003
Ing.	SPN	2003	Standard Product Nomenclature, 2003
Ing.	GO	2004/03/02	Gene Ontology, 2004_03_02
Ing.	MTHPDQ	2004	<i>Metathesaurus</i> Forms of Physician Data Query Terms 2004
Ing.	PDQ	2004	Physician Data Query, 2004
Ing.	CSP	2004	CRISP Thesaurus, 2004
Ing.	MTHFDA	2004/01	<i>Metathesaurus</i> FDA National Drug Code Directory, 2004_01
Ing.	NCI	2004	NCI Thesaurus, 2004
Ing.	NDFRT	2004/01	National Drug File - Reference Terminology, 2004_01
Ing.	HUGO	2004/04	HUGO Gene Nomenclature, 2004_04
Ing.	ICD9CM	2005	ICD-9-CM, 2005
Ing.	MTHICD9	2005	<i>Metathesaurus</i> additional entry terms for ICD-9-CM, 2005, 2005
Ing.	MEDLINEPL	2004/08/14	MedlinePlus Health Topics 2004_08_14, 20040814
Ing.	NCBI	2004/09/30	NCBI Taxonomy, 2004_09_30
Ing.	MTHCH	2005	<i>Metathesaurus</i> CPT Hierarchical Terms, 2005
Ing.	MDR	7.1	Medical Dictionary for Regulatory Activities Terminology (MedDRA), 7.1
Ing.	MDRAE	7.1	Medical Dictionary for Regulatory Activities Terminology (MedDRA), American English Equivalents, 7.1
Ing.	MDREA	7.1	Medical Dictionary for Regulatory Activities Terminology (MedDRA), American English Equivalents with expanded abbreviations, 7.1
Ing.	MDREX	7.1	Medical Dictionary for Regulatory Activities Terminology (MedDRA), with expanded abbreviations, 7.1
Ing.	MSH	2004-05/10/12	Medical Subject Headings, MSH2005_2004_10_12
Ing.	LNC	2.13	LOINC 2.13
Ing.	HCPCS	2005	Healthcare Common Procedure Coding System, 2005
Ing.	MTHHH	2005	<i>Metathesaurus</i> HCPCS Hierarchical Terms, 2005
Ing.	RXNORM	2005AA	RXNORM Project, META2005AA Cumulative Update 2004_11_17, 2005AA
Ing.	MTH		UMLS <i>Metathesaurus</i>
Ing.	SRC		<i>Metathesaurus</i> Source Terminology Names

Tabela 9: Características da base de conceitos

Abreviação	Medida	Valor
C	Total de conceitos diferentes	845.824
CI	Total de conceitos diferentes em inglês	845.824
CP	Total de conceitos diferentes em português	67.409
S	Total de strings diferentes	1.715.839
SI	Total de strings diferentes em inglês	1.609.057
SP	Total de strings diferentes em português	106.782
LS	Comprimento médio dos strings	3,96
LSI	Comprimento médio dos strings em inglês	4,44
LSP	Comprimento médio dos strings em português	3,45
NSM	Número de strings em português com comprimento maior que a média	40.968
NSN	Número de strings em português com comprimento menor ou igual que a média	65.814
NSM4	Número de strings em português com comprimento maior que quatro	24.909
NSN4	Número de strings em português com comprimento menor ou igual a quatro	81.873
P	Total de palavras diferentes	418.144
PI	Total de palavras diferentes em inglês	376.596
PP	Total de palavras diferentes em português	41.548

Obs: medidas obtidas para os vocabulários do arquivo mrconso.rtf importado para uma base de dados relacional, considerando os vocabulários mencionados anteriormente.

2.7.4 RadLex

Desenvolvido pela *Radiological Society of North America* (RSNA) o Radlex é um léxico voltado para a organização e recuperação de informação na área de radiologia. Seu desenvolvimento foi motivado pela deficiência dos sistemas de terminologias e vocabulários padrões em medicina quando utilizados como léxico em radiologia.

A estrutura descrita nesta seção é baseada na versão inicial do Radlex, uma versão que encontrava-se em desenvolvimento. No final de 2006 foi liberada uma nova versão, com o acréscimo de novos itens e com algumas modificações estruturais. Como não havia tempo hábil para incorporar a nova versão ao trabalho, esta tese prosseguiu o trabalho com a versão inicial.

A versão inicial possui um rascunho de um léxico para a área de radiologia de tórax em inglês. Em seu desenvolvimento pretende-se unificar e ampliar termos extraídos de outros

léxicos como o ACR, SNOMED, UMLS, *Fleischner Society Glossaries* e o DICOM. Os termos do Radlex devem ser disponibilizados pela *Internet* de forma gratuita.

Os termos existentes no RadLex são distribuídos entre categorias, descritas na Tabela 10. As categorias são divididas em três grupos, relacionados ao grau de experiência para o desenvolvimento de cada um. Por fim, cada termo possui um conjunto de atributos que descrevem sua estrutura interna e as ligações com outros sistemas de terminologias.

Tabela 10: Categorias de termos do RadLex [RadLex 2003]

	Categoria	Descrição
1	Patient Identifiers	Informações de identificação do paciente e dados demográficos
2	Clinical History	Histórico médico do paciente.
3	Image Acquisition, Processing, and Display	Informações sobre a aquisição, processamento, visualização e interpretação de imagens.
3.1	Examination Type	Informação sobre tipo de exame.
3.2	Technique	Informações sobre equipamentos, parâmetros de aquisição, medicamentos e contrastes, posicionamento do paciente e aquisição digital ou analógica.
3.3	Image Processing and Preparation	Informações sobre a preparação de uma imagem, ex: uso algoritmos, window/level, etc.
3.4	Observer Context	Informações sobre quem visualizou as imagens, tipo de mídia, uso de software de detecção, formas de suporte a decisão.
4	Location on the Image	Informações sobre o posicionamento do observador ao examinar uma imagem.
5	Image Quality	Informações sobre a qualidade de uma imagem.
6	Anatomic Location	Informações sobre partes anatômicas do corpo.
7	Findings	Informações sobre atributos observados na imagem.
7.1	Visual Features	Características gerais que não dependem de alguma referência como partes anatômicas, processo patológico ou estrutura. Ex: opacidade.
7.2	Morphologic and Physiologic Processes	Atributos que descrevem a morfologia ou um processo fisiológico, tais como massa, nódulo, aumentado, etc.
7.3	Diagnoses and Etiologies	Diagnóstico encontrado.
8	Relationships	Relacionamentos entre os termos de categorias diferentes.
8.1	Causal Relationships	Ocorre normalmente entre achados.
8.2	Logical Relationships	Relacionamentos lógicos mais elaborados sobre a presença ou falta de combinações de achados.
8.3	Spatial and Visual Relationships	Ocorre tipicamente entre os achados e partes anatômicas.
9	Uncertainty	Representa o grau de certeza aplicado a um termo ou achado em particular.
10	Conclusions	Conclusões globais sobre um exame.
11	Recommendations	Lista de ações relacionadas aos resultados de um exame.
12	Teaching Attributes	Indica o nível de habilidade necessária para a interpretação correta das imagens interpretadas.
12.1	Perceptual Difficulty	Nível de dificuldade de percepção.
12.2	Analytical Difficulty	Nível de dificuldade analítico.
12.3	Representativeness	Indica quando um caso é um caso típico (exemplo)

Tabela 11: Grupos de termos [RadLex 2003]

Grupo	Termos do grupo
Subspecialty-Specific Imaging Terms	Anatomic Location
	Findings
	Recommendations
	Conclusions
Imaging-Specific Terms	Image acquisition, processing, and display
	Location on the image
	Image quality
	Uncertainty
	Relationships
	Teaching attributes
General Clinical Terms	Patient identifiers
	Clinical history

Tabela 12: Atributos dos termos [RadLex 2003]

Atributo	Descrição
RadLex Unique Identifier	Identificado único de um conceito no RadLex (alfanumérico).
RadLex Name	Nome do termo.
Source	Fonte de onde o termo foi extraído.
Definition	Definição ou explicação textual do termo.
Sample Images	Imagens que contenham informação representativa sobre o termo.
Related Terms	Lista de termos em outros léxicos que contenham definições similares.
Related Lexicon Unique Identifier	Identificador no léxico referenciado.
Comments	Informação textual extra.
Version	Atributo inteiro que é incrementado a cada atualização do léxico.
Last Update	última data de alteração do termo mais a identificação de quem fez a atualização.

2.8 Alguns Trabalhos em Língua Portuguesa na área Médica

2.8.1 DECs (Descritores de Ciências e Saúde)

É um vocabulário estruturado trilingue (português, inglês e espanhol) usado para indexar artigos científicos, livros, *proceedings*, relatórios técnicos e outros tipos de materiais para busca e recuperação de literatura científica. [DECS]

É organizado em estrutura de árvore, possuindo cerca de vinte e seis mil descritores, porém está em permanente processo de mudança para incorporação de novos descritores. Participa do desenvolvimento do UMLS.

2.8.2 LILACS - Literatura Latino-americana e do Caribe em Ciências da Saúde

O LILACS é uma metodologia que faz parte da BVS (Biblioteca Virtual de Saúde). É constituída por normas, manuais, guias e aplicativos, tendo como objetivo a coleta, a seleção, a descrição e a indexação de documentos para uma base de dados. Deve servir como uma metodologia comum para o tratamento descentralizado da literatura da área de saúde produzida na América Latina e Caribe. O conjunto de bases de dados que utilizam os padrões

estabelecidos na Metodologia LILACS é chamado de Sistema LILACS. (Adaptado de [LILACS])

2.9 Trabalhos Relacionados

2.9.1 Avaliação da Tradução Automatizada de Relatórios de Radiografias de Tórax

Este projeto, em desenvolvimento no departamento de UP&D do InCor, tem como objetivo avaliar a eficiência do uso de um tradutor automático do português para o inglês. São utilizados laudos de radiografia de tórax para posterior extração de informação. Basicamente o projeto consiste em aplicar um tradutor de textos comercial em um conjunto de relatórios médicos para então submeter os relatórios ao MedLEE e obter um descritor dos relatórios [Castilla 2005].

2.9.2 Base de Imagens para Avaliação de Algoritmos de Processamento de Imagens

Este projeto, em desenvolvimento no departamento de P&D do Instituto do Coração da Faculdade de Medicina da USP, tem como objetivo o desenvolvimento de uma base de dados de imagens de uso livre para avaliação de algoritmos e *softwares* de processamento de imagens. A base foi projetada para trabalhar de forma distribuída, com espaço para armazenamento de documentações e *softwares* de processamento de imagem, além de oferecer ferramentas para anotação de imagens.

Futuramente os laudos anotados podem fornecer informação adicional às imagens que venham a ser armazenadas na Base de Imagens para Avaliação de Algoritmos de Processamento de Imagens. Por exemplo, a informação extrai'da dos laudos pode ser usada para indexar a base de dados de imanges.

3 METODOLOGIA

3.1 Metodologia de Projeto e Desenvolvimento

Para o desenvolvimento da pesquisa proposta, o contexto da aplicação de PLN foi reduzido para o vocabulário de laudos de Raios-X de Tórax pela disponibilidade dos textos para a realização do trabalho e para simplificar o desenvolvimento do léxico e das aplicações de reconhecimento de linguagem natural. Um vocabulário mais amplo exigiria uma quantidade grande de textos genéricos da área médica para treinamento dos algoritmos de PLN além de que parte da organização do vocabulário requer a execução de trabalho manual, o que levaria a um maior consumo de tempo de desenvolvimento. O vocabulário reduzido ao escopo de Raios-X de Tórax cumpre o propósito de prova de conceito.

A proposta do trabalho envolve diversas tarefas que encadeadas viabilizam a anotação dos laudos. Entre as tarefas temos:

- ✓ Pesquisa da teoria e algoritmos ligados à área de PLN e EI;
- ✓ Avaliação de sistemas de terminologia médicos e organização de um sistema de terminologias para que sirvam de vocabulário básico no trabalho. Foram escolhidos para o trabalho o UMLS e o Radlex;
- ✓ Escolha, implementação e avaliação de algoritmos básicos de PLN;
- ✓ Construção de um *Corpus* de laudos anotado para treinamento de algoritmos e de um *Corpus* de teste para a área estudada;
- ✓ Criação do modelo de representação do conhecimento na área de Raio-X de Tórax e mapeamento deste para um modelo de dados para armazenamento em um gerenciador de banco de dados;
- ✓ Programação das ferramentas de extração de informação dos laudos médicos;
- ✓ Anotação automática dos laudos;
- ✓ Avaliação de resultados;

O passo inicial do trabalho foi a realização de um levantamento bibliográfico sobre a área de processamento de linguagem natural e dos projetos e ferramentas já existentes na área. Neste levantamento merece destaque o projeto MedLee, da *Columbia University*, cujo tema é

justamente a extração de informação em textos médicos em língua inglesa, o conjunto de ferramentas NLTK (*Natural Language Toolkit*), que se trata de um kit de ferramentas em linguagem *Python* voltadas ao ensino de PLN, o Protégé, uma ferramenta utilizada no projeto de ontologias e o UMLS e o Radlex, que são os sistemas de terminologias médicas escolhido para o trabalho.

Escolhidos os sistemas de terminologias, estes foram organizados em um banco de dados relacional, no caso o PostgreSQL 8. Foi escolhido o PostgreSQL por se tratar de um gerenciador robusto, de ampla aceitação, de código fonte aberto e por apresentar diversas características como *triggers*, *stored procedures*, integridade referencial e por ser uma banco relacional, mas que oferece alguns recursos semelhantes a orientação a objetos.

A ferramenta Protégé não foi usada para o armazenamento de dados, pois foi concebida para o projeto de ontologias com as respectivas bases de conhecimento. Usa uma base de dados em formato de documentos XML armazenados na estrutura de diretórios do aplicativo. Considerando o tamanho da base do UMLS, que possui cerca de 1.700.000 *strings*, a forma de armazenamento do Protégé não é conveniente.

A base de dados de terminologias médicas em PostgreSQL recebeu dados importados tanto do UMLS como do Radlex. Para a importação dos dados foi necessária a criação de algumas estruturas adicionais como índices, o armazenamento de algumas consultas pré-processadas para diminuir o tempo de busca em alguns casos e a normalização de alguns dados. No caso do Radlex, este foi submetido a um aplicativo de tradução automática e a saída traduzida foi revisada manualmente por um profissional da área médica. Os conceitos traduzidos foram introduzidos na base de dados, sendo mantidas os relacionamentos e a hierarquia existentes na versão inglesa do vocabulário.

Com o auxílio da pesquisa inicial, optou-se por trabalhar com o NLTK, por apresentar a implementação de uma série de ferramentas já prontas destinadas ao processamento de linguagem natural, o que facilita o desenvolvimento deste trabalho. Apesar de ser um conjunto de ferramentas destinado ao ensino, possui todas as características necessárias para o desenvolvimento de um projeto de pesquisa e para provar os conceitos discutidos neste trabalho. Esta escolha foi influenciada pelo fato de que o processo envolvido na extração de informação possui passos e ferramentas gerais que são utilizadas na maioria dos casos, como

por exemplo: *parsers* e rotuladores (*POS-Taggers*). O NLTK já oferece estas ferramentas padrão, exigindo apenas a adaptação à língua portuguesa e aos objetivos deste trabalho.

O passo seguinte, refere-se a montagem do conjunto de *Corpus* e a avaliação de ferramentas. Foi necessário organizar um conjunto de dados de treinamento e outro de testes para o algoritmo *Brill Tagger* [Brill 1992], implementado no NLTK. Detalhes sobre a montagem do *Corpus* serão descritos mais adiante na seção adequada.

Para que os laudos possam ser devidamente anotados, é necessário definir uma estrutura que descreva o tipo de informação buscada nos textos. Tal estrutura utiliza um modelo de representação de conhecimento modelado com a ferramenta Protégé 2000 [Protégé], um aplicativo voltado à criação de ontologias. Trata-se de um modelo simples, visto que deve ser preenchido automaticamente, porém deve conter o mínimo de informação necessária para o posterior uso dos textos anotados por algoritmos de recuperação de informação.

Com o modelo de dados definidos, as ferramentas do NLTK puderam ser agrupadas e adaptadas para a obtenção de um aplicativo de extração de informação a ser aplicado no *Corpus* de teste do projeto.

A aplicação de EI contam com um grupo de ferramentas, que devem funcionar de forma conjunta para que o objetivo de extrair informações de textos de laudos seja alcançado. O conjunto de ferramentas utilizado conta com ferramentas implementadas especificamente para esta tese e com algumas ferramentas oferecidas pelo NLTK ou disponíveis sob licença gratuita.

Inicialmente se mostrou necessária a criação de um algoritmo de busca por conceitos na base de dados de terminologias médicas. O algoritmo serve para que possamos comparar duas *strings* que representam um conceito médico e dizer o quanto essas *strings* são semelhantes. O algoritmo é necessário em duas situações no contexto desta pesquisa: A avaliação da adequação do vocabulário em português do UMLS ao vocabulário usado nos laudos de Raio-X de Tórax e para a busca de *strings* extraídos dos textos no léxico médico usado neste trabalho.

O algoritmo de busca serviu para verificar a existência de diversos conceitos contidos nos laudos na base do UMLS, processo esse que será descrito posteriormente com mais detalhes.

Entre as ferramentas pertencentes ao NLTK utilizadas temos o *Brill Tagger*, devidamente treinado com o *Corpus* de treinamento, o rotulador baseado em unigramas (*Unigram Tagger*) e o módulo de *Chunk Parser*. O rotulador baseado em unigramas foi utilizado como rotulador auxiliar, uma característica exigida pela implementação do *Brill Tagger*. O *Chunk Parser* é utilizado para separar pequenos trechos de interesse no texto, sendo utilizada a implementação disponível no NLTK.

A idéia inicial era extrair os sintagmas nominais através do *Chunk Parser*, porém este se mostrou eficiente apenas para extrair expressões mais simples, sendo que optou-se por fazer a extração de sintagmas em duas fases. A primeira fase implementada através do *Chunk Parser* do NLTK e a segunda que faz um reprocessamento dos *chunk* iniciais e cria estruturas mais complexas baseadas em trechos texto. A segunda fase foi implementada especificamente para este projeto e é constituída de um algoritmo baseado em regras, que será descrito no capítulo de implementação. Esta fase de processamento denomina-se “Segmentação do Texto”.

Também foi implementada uma ferramenta que verifica os trechos de texto identificados pela fase de segmentação, tentando encontrar o conceito mais adequado na base do Radlex e do UMLS para cada segmento. Optou-se por utilizar as duas bases, apenas como complemento. A abordagem aqui é que se um conceito não é encontrado no Radlex, ele é buscado no UMLS, a título de complementação da informação.

A última ferramenta que deveria ter sido implementada é a que identifica os conjuntos de conceitos segundo o modelo de extração de informação. Esta ferramenta não foi implementada devido ao tempo, mas deve ser finalizada em trabalhos futuros.

Tendo implementado a estrutura de extração de informação, o ultimo passo do processo foi a avaliação dos resultados obtidos. Esta avaliação é baseada em formas clássicas de avaliação existentes em PLN, com a utilização de medidas como recuperação, precisão e o *f-measure*. Alguns dos passos são avaliados em seu término, para que o processo como um todo não seja prejudicado.

A avaliação do desempenho do *Brill Tagger* é feita, através de correções manuais e um aplicativo de comparação entre o texto de referência, corrigido manualmente, e o texto rotulado original.

O desempenho do algoritmo de *Chunk Parsing*, também é avaliado de forma manual. Uma amostra da saída final do algoritmo de *chunk* foi revisada manualmente contabilizando-se os trechos segmentados corretamente e os trechos segmentados erroneamente. A partir desta contagem foi verificada a taxa de erro da amostra e então através do teste de hipótese, verificou-se a estimativa da taxa de erros da população, com a devida significância.

O desempenho da rotulação é de mais difícil avaliação, pois depende fortemente da qualidade do léxico utilizado e dos passos anteriores. A avaliação também é manual e por amostragem, pois não há um *Corpus* anotado para comparação. Uma possibilidade seria a comparação com outros aplicativos de extração de informação, porém seria necessário ter o acesso a tais aplicativos, que deveriam ter uma implementação em português e ainda precisaria ser de alguma forma normalizado com relação a aplicação discutida nesta tese.

3.2 Organização do Trabalho

3.2.1 Premissas para Organização do Trabalho

Este trabalho tenta seguir as recomendações de [Friedman 1997] para a avaliação de algoritmos de PLN. Estas recomendações são importantes não só no estabelecimento da metodologia de avaliação, mas na criação de toda a estrutura do trabalho, pois trata de aspectos ligados a escolha e estrutura dos dados ou então versa sobre como os dados devem ser manipulados de forma a evitar resultados viciados, isto é, o desenvolvimento não pode ser guiado pelo conhecimento prévio das características particulares do conjunto de dados usado.

No decorrer do trabalho será discutido o respeito ou a fuga dos itens da recomendação, com sua respectiva justificativa e comentários sobre sua implementação.

A Tabela 13 mostra os tópicos abordados por Friedman para a avaliação de ferramentas de PLN em geral e voltadas a medicina:

Tabela 13: Recomendações para o desenvolvimento de aplicações de PLN [Friedman 1997].

Minimizar a interferência <ol style="list-style-type: none"> 1) O desenvolvedor não pode ver o conjunto de documentos de teste. 2) Se especialistas do domínio trabalhado forem escolhidos para estabelecer o padrão de referência, estes não podem participar do projeto e desenvolvimento do sistema. 3) O desenvolvedor não deve realizar a avaliação. 4) O sistema de PLN deve ser fechado antes do início da fase de testes. 5) Se a capacidade de generalização do sistema está sendo testada, o desenvolvedor não deve conhecer os detalhes do estudo. 7) Idealmente, a pessoa que projeta a avaliação não deve ser a mesma que desenvolve o sistema.
Estabelecimento de um padrão de referência <ol style="list-style-type: none"> 8) Se especialistas são usados para determinar um padrão de referência, deve haver um número suficiente deles para assegurar a variabilidade do padrão de referência. 9) O conjunto de testes deve ser grande o suficiente para que seja possível distinguir níveis de desempenho. 10) Se forem usados especialistas no estabelecimento do padrão de referência, o tipo de especialista deve ser apropriado (ex: radiologista X residente).
Descrição dos métodos de avaliação <ol style="list-style-type: none"> 11) O método usado para a obtenção do padrão de referência deve ser descrito claramente, especialmente se houver o envolvimento de especialistas. 12) A forma de escolha dos documentos de teste deve ser descrita. 13) Os métodos usados para calcular o desempenho devem ser descritos claramente, ainda mais no caso da utilização de métodos não padronizados.
Apresentação de resultados <ol style="list-style-type: none"> 14) Os resultados de desempenho devem ser relacionados com o conjunto inteiro de testes. 15) No caso da utilização de especialistas deve haver a troca de pareceres entre estes. 16) Os intervalos de confiança devem ser fornecidos para toda medida.
Discussão das conclusões <ol style="list-style-type: none"> 27) As limitações do estudo devem ser discutidas. 18) Os resultados devem ser discutidos segundo os requisitos da aplicação. 19) Evitar a generalização de resultados. 20) Uma análise das falhas do sistema deve ser fornecida juntamente com a discussão das dificuldades em se obter as devidas correções.

3.2.2 Metodologia de Avaliação do UMLS

Considerando o conjunto que compõe o UMLS, neste trabalho inicialmente havia um maior interesse no *Metathesaurus* e na rede semântica, por organizarem o conjunto de conceitos que servem como referência para o algoritmo de extração de informação na área médica. Desta forma, o primeiro passo tomado para a utilização do *Metathesaurus* foi a avaliação de sua adequação à coleção de laudos escolhida para trabalho, no caso a coleção de textos de Raio-X de Tórax.

O processo de avaliação foi concebido com o intuito de constatar a existência dos conceitos utilizados nos laudos de Raio-X de Tórax no vocabulário do UMLS. A adequação do vocabulário do UMLS com o vocabulário utilizado nos laudos é importante para o

desempenho dos algoritmos de extração de informação, visto que só é possível gerar laudos anotados, se for possível reconhecer os conceitos trabalhados nos textos.

O método de avaliação possui poucos passos, mas é semi-automático, levando ao consumo de tempo no que diz respeito à parte manual da avaliação. Os passos envolvidos são:

1) Desenvolvimento de algoritmo de comparação entre frases.

O algoritmo de comparação é inspirado no trabalho de avaliação de tradução automática denominado Bleu [Papineni 2002]. O algoritmo compara duas frases e devolve como resposta um valor que indica o quanto elas se assemelham;

2) Composição de 4 grupos de teste:

O primeiro grupo é formado por 100 conceitos extraídos da própria base do UMLS através de sorteio, este grupo tem o objetivo de avaliar se o algoritmo usado é capaz de recuperar todos os conceitos, uma vez que todos existem na base de dados. O segundo grupo é composto pelos mesmos conceitos, porém estes foram removidos da base de dados para verificar o comportamento do algoritmo quando os conceitos buscados não existem na base de dados. O terceiro grupo é formado por 50 conceitos sorteados a partir de um dicionário de conceitos médicos existente nas páginas da Faculdade de Medicina da Universidade de São Paulo. Este teste tem como objetivo verificar o comportamento do algoritmo ao buscar por conceitos que não necessariamente foram representados de forma idêntica aos conceitos existentes na base de dados do UMLS. O quarto teste usa 20 conceitos extraídos manualmente de resumos de artigos existentes na base de BVS (Biblioteca Virtual em Saúde) [BVS].

3) Escolha de um conjunto de conceitos extraídos dos laudos de Raio-X de Tórax para busca na base do UMLS.

A busca deve devolver como resultado um conjunto de conceitos que mais se assemelhem a cada frase procurada acompanhada de uma pontuação que indica o quanto a frase buscada é semelhante à encontrada.

4) Calibração da busca através da seleção de um valor de corte para a pontuação de semelhança entre as frases. Como o algoritmo foi desenvolvido especificamente

para este trabalho, não havendo referências de execuções anteriores. O valor de corte foi obtido empiricamente, isto é, após alguns testes iniciais e a observação dos resultados, foi estabelecida uma faixa de valores iniciais de corte para utilização na avaliação da busca, sabendo-se que quanto maior o valor de corte, mais semelhantes são os *strings*.

5) Avaliação dos resultados obtidos com os passos 3 e 4.

Nesta etapa é verificada a existência ou não de conceitos na base que representem os conceitos procurados, podendo-se então indicar se o vocabulário dos laudos está contido no vocabulário do UMLS.

Todos os resultados foram inseridos em planilhas de dados, onde foram executados os cálculos de medidas padrão de RI, como recuperação e precisão além de ter sido levantada a curva ROC referente ao experimento.

Os resultados obtidos devem indicar a viabilidade da utilização do *Metathesaurus* como vocabulário base para a extração de informação dos laudos de Raio-X de Tórax, além de que após os testes e a avaliação foi possível identificar mais adequadamente o valor de corte mínimo a ser utilizado posteriormente pelo algoritmo de comparação em outras fases de projeto.

Cogitou-se a utilização do UMLS apenas por que o RadLex é um vocabulário ainda em desenvolvimento existindo apenas em língua inglesa. Apesar de traduzido especialmente para este projeto, a versão disponível até o fim de 2006 não era completa. Apesar de incompleto, o Radlex possui uma perspectiva maior de atender aos requisitos de uma aplicação de PLN voltada a radiologia.

3.2.3 Metodologia de Organização do Corpus

As etapas do processo de extração de informação exigem que os textos analisados recebam anotações sintáticas. Estas informações são importantes na determinação do papel de cada elemento contido em uma sentença, permitindo assim que os conjuntos de palavras sejam agrupados e que sejam encontrados os relacionamentos entre cada conjunto identificado. Resumidamente, é feita uma análise sintática de cada frase do texto para então se determinar quais os conjuntos de palavras que formam possíveis conceitos médicos a serem

buscados no sistema de ontologias usado. Identificados os conceitos médicos, torna-se possível anotar o texto com códigos que formam um resumo do conteúdo do texto.

A anotação sintática do texto é feita através de rotuladores que devem ser treinados em um *Corpus* previamente anotado manualmente. O treinamento dos rotuladores é necessário por se tratarem de algoritmos probabilísticos e baseados em n-gramas ou então em aprendizado supervisionado.

Assim sendo, faz-se necessária a criação de um *Corpus* de laudos. Tal *Corpus* é dividido em dois conjuntos, um conjunto de treinamento e outro de testes.

O *Corpus* de treinamento foi inicialmente anotado manualmente com o conjunto de rótulos do *Brown Corpus* [Brown 1979], reservadas algumas pequenas adaptações, descritas mais adiante.

Obtido o *Corpus* anotado este foi utilizado para treinamento da implementação do *Brill Tagger* existente no NLTK. Posteriormente, o rotulador é utilizado em uma das fases do processo de extração de informação.

Note que os algoritmos de rotulação mais conhecidos, como o *Brill Tagger*, por exemplo, são sensíveis ao *Corpus* de treinamento, o que significa que para processar laudos de Raio-X de Tórax, é necessário treinar o algoritmo com laudos de Raio-X de Tórax.

Após treinar o algoritmo de rotulação, o passo seguinte foi verificar a qualidade das saídas geradas pelo algoritmo através do método de avaliação descrito a seguir.

3.2.3.1 Avaliação

Após treinar o algoritmo do *Brill Tagger*, o próprio *Corpus* de treinamento foi submetido ao rotulador. Através de um *script* escrito em *python*, a nova saída foi comparada sentença a sentença com o *Corpus* de treinamento. Com este procedimento obteve-se uma primeira medida do desempenho do rotulador. Se o algoritmo fosse ideal, a saída rotulada pelo *Brill Tagger* deveria ser idêntica ao *Corpus* de treinamento, o que não ocorreu.

O próximo passo foi submeter o *Corpus* de teste ao rotulador e fazer a correção manual dos resultados para completar avaliação. Corrigida a saída rotulada do *Corpus* de teste, esta pode ser comparada, através do mesmo *script* em *Python* mencionado na etapa

anterior, com a saída original para levantamento da taxa de erros. Os resultados serão discutidos no capítulo adequado.

Este procedimento foi utilizado para determinar se a implementação em questão é satisfatória e para se estimar se o processo de EI pode vir a ser afetado em caso de um baixo desempenho do rotulador..

3.3 Construção de Ontologias

Com base nos experimentos realizados com o UMLS, descritos no capítulo de resultados, ficou claro que no momento este sistema de terminologias não é o mais adequado para a indexação e reconhecimento de conceitos na área de Raio-X de Tórax. O RadLex seria a melhor escolha para o trabalho na área de radiologia, no entanto este sistemas de terminologias não existe em português e uma versão mais completa foi liberada apenas no fim de 2006. A versão inicial do Radlex, possui cerca de 1.300 termos, enquanto a mais nova ultrapassou os 6.000 termos. A versão 2005AA do UMLS, utilizada neste trabalho, possui cerca de 60.000 termos em português, porém em áreas diversas do conhecimento médico.

Inicialmente a idéia era ampliar o sistema de terminologias do Radlex, para complementar seu conteúdo ainda deficiente, porém com a expectativa da liberação da nova versão optou-se por não levar adiante a idéia da expansão, pois seria todo um trabalho no mínimo redundante com referência a equipe do Radlex.

Segundo [Noy 2001], as ontologias devem se reutilizáveis, sendo uma boa prática reciclar o conhecimento de uma ontologia já existente. Uma vez que já existem diversos sistemas de terminologias na área médica, partiu-se do principio de apenas utilizar de forma conjunta os vocabulários das terminologias já mencionadas para uma maior adequação ao vocabulário de Raio-X de Tórax em português.

Ainda restava o problema de que o Radlex existe na versão em língua inglesa. Para montar o vocabulário em língua portuguesa, o termos originais foram submetidos a um sistema de tradução automática [Systran]. A saída traduzida automaticamente foi revisada manualmente para obtenção do vocabulário em português. Os conceitos em português foram introduzidos na base de dados do Radlex, mantendo-se as referências para a codificação e texto originais.

Para melhor visualização das ontologia do Radlex, e para facilitar a criação dos mapas entre ontologia médica e ontologia dos laudos, a hierarquia do Radlex foi exportada para a ferramenta Protégé, usada neste trabalho para modelagem das ontologias.

A estrutura hierárquica mapeada para o Protégé, resultou em um trabalho adicional, publicado no Congresso RSNA em 2006 [RSNA 2006].

3.4 Definição da Estrutura de Representação de Conhecimento dos Laudos

Em uma abordagem inicial, foi criada uma primeira estrutura de representação do conhecimento contido nos laudos na forma de um grafo que poderia ser convertido facilmente para um documento XML. Esta estrutura inicial foi reorganizada através da utilização da ferramenta Protégé. A nova estrutura na forma de uma ontologia permite uma melhor representação da estrutura do documento, permitindo a definição de relacionamentos entre as diferentes partes do documento, a definição de restrições para cada conceito representado e o mapeamento mais direto para a ontologia que representa o vocabulário utilizado como referência.

3.5 Metodologia de Avaliação

O processo de construção de um sistema como proposto neste trabalho envolve várias etapas de trabalho e a criação ou adaptação de diversas ferramentas de PLN.

Basicamente são usados dois métodos distintos para avaliação. O primeiro método envolve a tomada de medidas tradicionalmente usadas na avaliação de RI e EI, como a precisão e a taxa de recuperação. Como essas medidas não são adequadas para avaliar a inexistência de itens buscados, optou-se por complementar a avaliação através das métricas oferecidas pela curva ROC. Lembrando que para este trabalho é interessante não apenas identificar conceitos em um sistema de terminologias, mas também saber que um *string* não encontrado na base de conceitos provavelmente não é um conceito ou não está definido na base.

Cada passo do processo deve possuir uma avaliação individual, possibilitando a descrição dos pontos fortes e falhos em cada etapa da construção das ferramentas utilizadas

no sistema final de extração de informação. A avaliação individual deve influenciar nas opções de etapas adjacentes e fornecer material para a avaliação final da eficiência do sistema

4 ANÁLISE E PROJETO

4.1 Contexto do Trabalho

O primeiro passo para a criação de qualquer sistema é o estudo do problema a ser resolvido para a posterior elaboração da análise de requisitos do sistema. Desta forma, esta seção do texto busca explicar o contexto que motivou o desenvolvimento deste trabalho.

Apesar de os sistemas de informação na área médica já possuírem muitos anos de existência e consolidação, ainda é uma prática muito comum armazenar texto livre nas bases de dados médicos. Apesar do armazenamento de texto livre ser desvantajoso do ponto de vista computacional, ele não deixa de ser um requisito dos profissionais de saúde que precisam da liberdade para expressar seus achados e conclusões acerca de um paciente ou assunto.

Apesar da manutenção dos textos livres nas bases de dados médicas, uma padronização se faz necessária, principalmente com relação ao vocabulário utilizado nos textos. A padronização do vocabulário utilizado está ligada a clareza dos textos e como efeito pode facilitar o processamento computacional destes. A padronização de vocabulários é amplamente requisitada e aceita, percebemos o fato devido a existência de diversos sistemas de terminologias médicas, como o MESH, o SNOMED-CT, o UMLS ou o RadLex. No caso do texto livre, a utilização de vocabulários padrão depende apenas da disciplina dos profissionais envolvidos no processo de alimentação da base de dados hospitalar com os textos médicos.

Uma vez armazenados os laudos, a informação contida nos mesmos, não pode ser mais recuperada de forma eficiente no contexto onde é necessária, justamente por não estar em um formato mais adequado para o processamento computacional.

Tomando como exemplo o conjunto de dados referentes aos laudos de Raio-X de Tórax, poderia ser interessante a um estudante ou pesquisador consultar a base em busca de um determinado conjunto de achados descritos nos laudos com as respectivas conclusões, para uma pesquisa clínica, para um estudo de caso ou mesmo para a busca de um caso que sirva como referência. Na forma como os dados são armazenados atualmente esta busca é praticamente impossível por não haver pelo menos uma indexação adequada dos laudos.

Organizar, indexar ou anotar os laudos manualmente é trabalhoso e demorado. Ainda seria necessário formatar todo o conjunto já armazenado na base de dados do hospital além de organizar um esquema para formatação dos laudos novos que alimentam diariamente tal base. A criação de um método automático para a anotação de laudos com códigos padronizados seria de grande utilidade para a organização de uma base de conhecimentos médica. Essa base poderia ser usada por profissionais de saúde que buscam por informação em meio a grande quantidade de dados clínicos.

4.2 Análise inicial do problema

Os requisitos do sistema implementado neste trabalho podem ser resumidos segundo os seguintes itens básicos:

- ✓ Anotar laudos de Raio-X de Tórax com rótulos que identifiquem o conteúdo do laudos segundo os achados, conclusões e diagnósticos;
- ✓ Utilizar um sistema de terminologias baseado em codificação de conceitos;
- ✓ Realizar o processo automaticamente, isto é, sem intervenção humana;
- ✓ Armazenar os laudos anotados em uma base de dados.

O escopo do trabalho foi diminuído para a manipulação de textos de Raio-X de Tórax, para simplificar o vocabulário utilizado e pela disponibilidade dos laudos para a pesquisa.

Os itens mencionados estabelecem qual o foco do trabalho, no entanto se o objetivo principal é anotar laudos, devemos ter em mente qual o tipo de informação que eventualmente será buscada nos laudos, isto é, para que possamos classificar ou rotular os laudos é necessário determinar quais os conceitos de interesse para uma aplicação em um domínio de trabalho determinado. Em um tutorial sobre construção de ontologias, Noy [Noy 2001] comenta sobre a criação de um levantamento das Questões do Domínio (*Competency Questions*) que servem para determinar quais tipos de perguntas uma base de conhecimento deve ser capaz de responder.

Para melhor entender qual tipo de questão pode surgir se baseando em um laudo, o Exemplo 12, mostra o texto de um laudo usado neste trabalho.

Exemplo 12: Texto original de um laudo

Osteofitose dorsal. Seios e cúpulas diafragmáticas livres. Vascularização pulmonar normal. Hilos de configuração, topografia e dimensões normais. Aorta alongada. Arco médio normal. Parênquima pulmonar de transparência normal. Mediastino sem alterações.

Percebemos que basicamente um laudo contém regiões anatômicas com as devidas observações sobre características visualizadas e um possível diagnóstico. As características observadas são tratadas aqui como achados. Uma análise inicial, nos leva as seguintes questões, considerando a criação de uma base de conhecimento baseada em laudos de Raio-X de Tórax:

Exemplo 13: Lista de Questões do Domínio para o problema de rotulação laudos de Raio-X de Tórax.

1. Quais os laudos referentes a uma dada doença?
2. Quais os laudos que contenham um dado achado?
3. Quais os achados relacionados a um dado diagnóstico?
4. Descrito um tipo de achado, quais os diagnósticos associados?
5. Quais os diagnósticos relacionados a uma dada região anatômica?
6. Quais as regiões anatômicas afetadas em um determinado diagnóstico?
7. Listar as possíveis descrições relacionadas a uma região anatômica.
8. Quais os grupos de doenças associados a um dado achado?
9. Quais os grupos de doenças associados a uma dada região anatômica?

Os rótulos criados para a anotação dos laudos devem expressar conceitos contidos no corpo dos laudos de forma que permitam a identificação das respostas para o conjunto de perguntas do Exemplo 13. É conveniente que tais rótulos possam ser associados a uma ontologia que permita não apenas identificar conceitos, mas também os relacionamentos entre estes.

Também é necessária a escolha de um tipo de notação para que seja feita a rotulação dos textos, sendo assim, para a anotação dos laudos julgou-se conveniente a utilização do formato XML. Para tanto é necessária a definição da estrutura de documento XML para anotação dos laudos, sendo que os rótulos utilizados devem permitir a identificação de conceitos através da utilização de uma codificação para o mesmo.

Conhecido um conceito, deve ser possível identificar a super-classe ao qual ele pertence segundo uma hierarquia predefinida. Em um nível alto da hierarquia, teríamos como super-classes conceitos como achados, conclusões e diagnósticos. Nos níveis mais baixos, encontramos a descrição do laudo segundo o que o profissional observou na imagem.

Figura 3: Representação em árvore da estrutura de uma laudo em formato XML.

Podemos observar na Figura 3 que um laudo é constituído por achados e conclusões. Uma conclusão se resume a um conjunto de diagnósticos, já ao achados são descritos por uma ou mais anatomias, características do achado (características), características da imagem (visualização) e objetos. Por exemplo, a frase “parênquima pulmonar de transparência normal.” assumiria a seguinte estrutura no documento XML de descrição do laudo:

Exemplo 14: Exemplo de trecho anotado semanticamente.

```

<ACHADO ORIGINAL="parênquima pulmonar de transparência normal.">
  <ELEMENTO TIPO="anatomia"> parênquima pulmonar </ ELEMENTO >
  <ELEMENTO TIPO="visualizacao"> transparência </ ELEMENTO >
  <ELEMENTO TIPO="característica"> normal </ ELEMENTO >
</ACHADO>

```

O elemento do tipo “anatomia”, indica uma parte ou região do corpo. O elemento “visualização” indica uma característica da imagem, como textura ou propriedade visual. O elemento “característica” contém a descrição geral do achado. O elemento “objeto” descreve objetos ou elementos externos ao corpo humano, como por exemplo “suturas metálicas”. Por fim, o elemento “geral” é utilizado apenas para denotar que um trecho de texto foi reconhecido como um provável conceito, porém não foi possível definir qual seu papel com referência ao conjunto de elementos.

Além de indicar a função semântica que cada trecho de texto obedece na frase, também é interessante codificar cada elemento identificado com um código de conceito, seja este código UMLS ou RadLex. A identificação dos códigos é feita através de atributos, descritos mais adiante através da definição do formato de arquivo de saída.

A identificação nos elementos do código de conceito satisfaz o requisito da utilização de uma codificação padronizada de conceitos médicos, no entanto, a identificação dos

conceitos no texto com a atribuição do código através do rótulo, depende do vocabulário utilizado e do nível de detalhes dos textos processados.

O processo de rotulação do texto deve ser feito automaticamente, isto é, o texto deve ser rotulado sem intervenção manual, utilizando para isso uma base de dados que contenha o vocabulário utilizado e uma série de ferramentas de PLN.

Neste ponto deve ficar claro que não se pretende fazer nenhum tipo de inferência sobre achados, anatomias e diagnósticos. Por exemplo, um diagnóstico só pode ser identificado se explicitamente descrito no texto do laudo. Posteriormente, uma busca na base de conhecimento deve retornar apenas os documentos que contenham um determinado termo de busca (ex. um achado) evidenciando os conceitos de interesse dentro do mesmo texto.

Apesar de não haver inferências sobre o conteúdo rotulado dos laudos, através da rede de relacionamentos da ontologia utilizada pode ser possível melhorar os resultados de uma busca na base de conhecimentos. Por exemplo, se é feita uma busca por “Doença de Chagas”, essa busca poderia ser ampliada através da utilização dos sinônimos do conceito procurado, definidos na ontologia utilizada.

Outra restrição do trabalho é que seu escopo prevê a definição da estrutura do documento rotulado a ser armazenado na base de conhecimento, o processamento automático dos textos para rotulação e a definição da ontologia utilizada para os rótulos utilizados. A estruturação da busca em si e da recuperação de informações na base de conhecimentos é um tópico a ser abordado em trabalhos futuros.

4.2.1 Estrutura dos Laudos

Os laudos são obtidos a partir do banco de dados do prontuário eletrônico de pacientes existente no InCor, sendo eliminada a identificação do paciente da estrutura do laudo. Os laudos encontram-se em formato de texto puro, composto por linhas de 80 colunas (caracteres), sendo necessário fazer uma limpeza prévia no texto para eliminar excesso de espaços em branco, sendo eles salvos em arquivos de texto puro.

Normalmente apresentam duas seções, uma de descrição da técnica e outra de descrição do laudo em si. O Exemplo 15 mostra o texto de um laudo como ele é obtido do banco de dados do hospital.

Exemplo 15: Exemplo de laudo**TECNICA:**

Exame realizado em equipamento de alta potência, com técnica de alta kilovoltagem, que fornece escala de contraste longa (baixo contraste).

LAUDO:

Osteofitose dorsal. Acentuação da cifose fisiológica. Seios e cúpulas diafragmáticas livres. Vascularização pulmonar normal. Hilos de configuração, topografia e dimensões normais. Imagem cardíaca aumentada. Aorta alongada. Arco médio normal. Parênquima pulmonar de transparência normal. Mediastino sem alterações.

Notamos que o texto do laudo é simples e objetivo, no que diz respeito à construção literária. Em uma análise do conjunto de laudos, percebemos também que o vocabulário utilizado é mais ou menos constante, isto é, frases como “Parênquima pulmonar de transparência normal.” aparecem em diversos laudos com grafia semelhante, devido a utilização de um dicionário próprio da instituição.

As construções encontradas na seção de laudos raramente possuem verbos, sendo mais semelhantes a uma lista de itens encontrados. Normalmente não há uma seção com uma conclusão ou diagnóstico final, o que requer uma inteligência adicional de programação além da PLN, caso se deseje identificar o diagnóstico em um laudo. Como a proposta deste trabalho é de apenas processamento de linguagem natural, a rotulação de conclusões e diagnósticos nos laudos não é garantida, podendo ser feita apenas quando explicitado o diagnóstico no texto através de alguma palavra chave ou então através da identificação de um conceito, que no sistema de ontologias utilizado é definido apenas como diagnóstico.

Para facilitar a manipulação dos laudos estes foram armazenados em um banco de dados relacional. A estrutura de armazenamento dos laudos pode ser vista no diagrama Entidade/Relacionamento do apêndice 11.7.

4.2.2 Sistemas de Terminologias

Para a rotulação dos textos médicos é necessário utilizar um sistema de terminologias, de preferência padronizado. Os vocabulários padronizados utilizados neste trabalho como referência, no caso o RadLex e o UMLS, já foram descritos com mais detalhes no capítulo 2.

Basicamente o sistema de terminologias utilizado possui duas utilidades: servem como léxico para o processamento dos textos e como referência para a organização da ontologia referente a estrutura dos laudos.

Os sistemas mencionados devem ser avaliados para averiguar sua adequação ao vocabulário utilizado nos laudos de Raio-X de Tórax. Baseando-se no trabalho de Langlotz e Caldwell [Langlotz 2002], a maioria dos léxicos médicos não são completos no que diz respeito a conceitos em radiologia. Langlotz e Caldwell sugerem a criação de um léxico direcionado a técnicas que envolvam imagens, o que vem a ser tornar realidade através do RadLex.

Entre os vocabulários avaliados por Langlotz e Caldwell, encontra-se o UMLS. A avaliação feita por Langlotz e Caldwell foi realizada através de um processo manual e para a porção inglesa dos vocabulários analisados.

Intuitivamente sabemos que o vocabulário do UMLS em português, que tem um tamanho aproximadamente de 8% do vocabulário em inglês, não deve conter os conceitos utilizados em Raio-X de Tórax, no entanto como parte do trabalho é proposta a criação de uma avaliação automática do UMLS, que pode ser facilmente adaptada para outros vocabulários. Detalhes do método de avaliação serão fornecidos mais adiante no texto.

4.3 Arquitetura Geral

Para rotular um texto utilizando ferramentas de processamento de linguagem natural é necessário um conjunto de ferramentas que juntas viabilizam a extração de informação. A Figura 4 mostra os módulos envolvidos no processo de rotulação de laudos segundo a proposta deste trabalho.

Figura 4: Módulos do sistema proposto

As ferramentas de PLN envolvem algoritmos como o *Brill Tagger*, o *Snowball Stemmer* e as ferramentas disponíveis no NLTK todos devidamente adaptados e treinados para a língua portuguesa quando necessário. Além dos algoritmos básicos, as ferramentas de

PLN englobam os algoritmos desenvolvidos especificamente para este trabalho para a anotação dos laudos.

O *Corpus* de teste e o de treinamento são constituídos por um conjunto de laudos de Raio-X de Tórax. O conjunto foi dividido em dois grupos, o grupo de treinamento e o grupo de testes. O grupo de laudos de treinamento é usado basicamente para treinar o *Brill Tagger*, enquanto que o conjunto de testes serviu tanto para avaliar o desempenho do *Brill Tagger* quanto para avaliar a aplicação de EI como um todo.

As ontologias constituem a base de conceitos da área médica e de informação semântica usada pela aplicação de EI aplicada a laudos de Raio-X de Tórax. Este trabalho considera o uso de duas ontologias, o RadLex e o UMLS.

Ambos os sistemas de ontologias acabaram por serem mesclados superficialmente na montagem documento de saída da aplicação de EI, para obtenção de um vocabulário mais completo. A adaptação das ontologias utilizadas envolve a tradução do RadLex para o português, a criação de estruturas de auxílio, como os índices de palavras e a normalização de sentenças.

O léxico é utilizado para determinar o vocabulário a ser reconhecido pelo sistema e é baseado nos conceitos definidos no RadLex e UMLS. A fronteira entre o léxico e os sistemas de terminologias é tênue, pois ao mesmo tempo que um grupo de palavras pode servir como léxico para a aplicação, também pode servir como um índice de palavras para a ontologia. Na realidade o que é chamado de léxico neste trabalho é o índice de palavras associadas as respectivas raízes (*stem*), não havendo ainda informação sintática sobre as palavras. A informação sintática pode ser adicionada futuramente, através da rotulação feita com o *Brill Tagger* dos termos da ontologia mais o trabalho manual de correção.

Definida a ontologia, o modelo de anotação dos laudos deve ser mapeado para a estrutura da hierarquia de conceitos, de forma a definir a interseção entre subconjuntos de conceitos e os rótulos usados para a anotação dos laudos. Por exemplo, um rótulo do sistema de anotação de laudos ligado a anatomia, só pode ser usado para marcar conceitos referentes a partes anatômicas, conforme o exemplo da Figura 5.

Figura 5: Exemplo de mapeamento entre rótulos dos laudos e uma ontologia

Tendo uma visão geral das partes do projeto, podemos agora definir o processo de anotação de um laudo. A Figura 6 descreve este processo.

Figura 6: Fluxo do processamento de laudos

O primeiro passo do processamento é feito com ferramentas padrão utilizadas em PLN e serve para preparar o texto para o algoritmo que deve reconhecer os sintagmas nominais contidos em um laudo.

Os passos dois e três estão sendo executados dentro do mesmo processo. O passo dois consiste apenas em buscar os rótulos que marcam a pontuação de final de sentença. O terceiro passo é composto em parte por uma ferramenta padrão contida no NLTK e em parte por um algoritmo próprio, explicado mais adiante. Neste ponto, a informação sintática obtida nos passos anteriores é usada para segmentar o texto em sintagmas ou uma estrutura próxima de um sintagma, que represente um candidato a conceito médico. A saída deste passo está em formato XML. Os trechos de interesse extraídos do texto, serão a partir de agora chamados de segmentos.

A saída do terceiro passo alimenta um algoritmo que busca cada candidato na base de dados da ontologia trabalhada, gerando como saída um arquivo XML onde cada conceito reconhecido é marcado com o respectivo código definido no sistema de ontologias. Cada conceito identificado também é marcado com uma pontuação que indica o quanto o *string* achado se assemelha ao buscado. A implementação da identificação de conceitos é feita através de algoritmos próprios escritos em *Python* e do léxico e Ontologias adotados para este trabalho.

A ultima fase de processamento envolve a identificação dos rótulos para a notação final do laudo. Esta fase do processamento deve receber como entrada um arquivo de laudo com os conceitos médicos já identificados e gerar como saída o laudo anotado em XML segundo uma estrutura predefinida de anotação. Esta fase é programada em *Python* e deve utilizar a Ontologia adotada para identificar os devidos rótulos do documento XML a ser gerado.

4.4 Estrutura de Arquivos de Processamento

4.4.1 Estrutura de Dados para a Saída de Laudos

Conforme já descrito, os laudos devem ser anotados em linguagem XML com rótulos que expressem os conceitos encontrados nos textos. A estrutura de documento XML proposta é descrita no Exemplo 16.

Exemplo 16: Modelo do documento XML para representação do laudo.

```
<?xml version='1.0' encoding='UTF-8'?>
<!-- ..... -->
<!-- Definição do formato de um laudo anotado ..... -->

<!-- ..... -->
<!-- Tag raiz
-->
<ELEMENT LAUDO (GRUPO)*>
<ELEMENT GRUPO (ACHADO|CONCLUSAO)* >

<!-- ..... -->
<!-- Definição dos elementos da tag de conclusão
-->
<ELEMENT CONCLUSAO (DIAGNOSTICO)*>
<ELEMENT DIAGNOSTICO (#PCDATA)>

<!-- ..... -->
<!-- Tags gerais
-->
<ELEMENT ACHADO (ELEMENTO)* >
<ELEMENT ELEMENTO (#PCDATA) >

<!-- ..... -->
<!-- Atributos:
```

```

<!-- GRUPO:
      Frase original que foi processada
-->
<!ATTLIST GRUPO ORIGINAL CDATA #REQUIRED>

<!--
      ACHADO:
      enfermidade: Usado quando foi identificada uma doença
      processo:    Usada quando foi identificado um processo, por exemplo,
                   uma inflamação
      funcao:      usado para definir uma função do organismo, por exemplo
                   respiração.
-->
<!ATTLIST ACHADO TIPO      ( enfermidade | processo | funcao | indeterminado ) #REQUIRED
                  ORIGINAL CDATA #IMPLIED>

<!--
      ELEMENTO:
      TIPO: Tipo de elemento
      COD:  Código do conceito
      FNT:  Vocabulário fonte
      VER:  Versão do vocabulário fonte
      PROC: texto procurado
      ENC:  texto encontrado
      CERT: Grau de certeza sobre a atribuição do rótulo
-->
<!ATTLIST ELEMENTO TIPO (anatomia|visualizacao|objeto|caracteristica|geral) #REQUIRED
                  COD   CDATA #IMPLIED
                  FNT   CDATA #IMPLIED
                  VER   CDATA #IMPLIED
                  PROC  CDATA #IMPLIED
                  ENC   CDATA #IMPLIED
                  CERT  CDATA #IMPLIED >

```

A estrutura hierárquica da Figura 3 foi criada com base na vistoria de parte dos laudos adquiridos para o trabalho e tendo como referência a estrutura hierárquica do RadLex. Esta estrutura serve como um primeiro esboço de documento que deve ser gerado pelo sistema. A partir da Figura 3 foi projetada a estrutura do documento referente ao laudo através da ferramenta *Protégé*. A descrição através do *Protégé* além de conter a estrutura hierárquica dos rótulos possui a definição dos atributos de cada um deles. A partir desta descrição, foi criado o DTD que o documento XML do laudo deve seguir.

4.4.2 Estrutura do Arquivo de Segmentos

Os segmentos de texto de um laudo, obtidos através do passo 3 do processamento (conforme a Figura 6), são armazenados em um arquivo intermediário em formato XML, segundo o Exemplo 17:

Exemplo 17: Arquivo DTD para os documentos intermediário de sintagmas

```

<?xml version="1.0" encoding="UTF-16LE"?>
<!ELEMENT SEG      (#PCDATA|SEG|COMP) *>
<!ELEMENT COMP     (#PCDATA) >
<!ELEMENT SENT     (SEG|COMP)+>
<!ATTLIST SENT     ORIG CDATA #REQUIRED>
<!ELEMENT LAUDO    (SENT)+>

```

Basicamente temos quatro tipos de rótulos. O rótulo LAUDO, que é a raiz do documento, o rótulo SENT, que delimita cada sentença existente no laudo, o rótulo SEG, que

denota um segmento de texto e o rótulo COMP, onde entram os elementos considerados como complemento dentro da sentença. O complemento não tem a mesma importância semântica de um segmento, contudo ainda carrega alguma informação, não sendo descartado.

O rótulo SENT possui apenas um atributo usado para não se perder a informação da sentença original.

4.4.3 Estrutura do Arquivo de Conceitos

O passo 4, segundo a Figura 6, é responsável por buscar cada segmento de texto na base de dados da ontologia utilizada. O Exemplo 18, mostra a definição do DTD usado pelo documento XML gerado nesta fase de processamento.

Exemplo 18: Arquivo DTD para os documentos intermediário com a informação dos conceitos

```
<?xml version="1.0" encoding="UTF-16LE"?>
<!ELEMENT TERM (TERM) *>
<!ATTLIST TERM
    ID      CDATA #REQUIRED
    SRC     CDATA #REQUIRED
    VER     CDATA #REQUIRED
    SEARCH  CDATA #REQUIRED
    FOUND   CDATA #REQUIRED
    SCORE   CDATA #REQUIRED>
<!ELEMENT SENT (TERM) *>
<!ATTLIST SENT ORIG CDATA #REQUIRED>
<!ELEMENT LAUDO (SENT) +>
```

O rótulo raiz do documento é o LAUDO, que deve abrigar um conjunto de sentenças, rotuladas como SENT. Neste ponto as sentenças continuam sendo a estrutura básica do texto, devendo possuir um único atributo que contém o valor da sentença original procurada.

Cada sentença possui diversos termos, rotulados como TERM, que são os conceitos identificados na base de dados da ontologia. Cada termo é caracterizado por um ID, que é o código no sistema de ontologias, uma fonte (SRC), que é o nome do sistema de ontologias, a versão da ontologia (VER), o *string* procurado originalmente na ontologia (SEARCH), o *string* encontrado na ontologia (FOUND) e a pontuação de semelhança atribuída pela comparação entre o *string* procurado e o encontrado (SCORE). O *string* encontrado é mantido no arquivo de conceitos para facilitar a avaliação posterior do resultado desta fase de processamento, pois apesar dos *strings* serem semelhantes, os valores semânticos são diferentes em algumas situações.

4.5 Ontologia e Léxico Utilizados

Como léxico básico deve ser usado neste trabalho uma tradução para o português do material disponibilizado pelo RadLex. O processo de criação do léxico em português envolve a tradução automática dos termos definidos em inglês no RadLex, através de uma ferramenta comercial de tradução automática, denominada Systran [Systran]. O Systran foi alimentado manualmente com um dicionário específico para medicina elaborado por um médico radiologista. A tradução dos termos foi revisada manualmente para eliminação de erros de tradução. Para maiores detalhes sobre a criação do dicionário médico e avaliação da tradução automática de conceitos médicos ver [Castilla 2005].

O novo léxico em português é utilizado em conjunto com o UMLS de forma a ampliar o máximo possível o vocabulário utilizado, no entanto, a utilização do UMLS deve considerar algumas restrições devido ao fato de a estrutura de anotação do laudo ser baseada no RadLex.

Figura 7: Hierarquia do RadLex visualizada com a ferramenta *Protégé*.

Os dados referentes ao RadLex foram obtidos através do *site* da RSNA em formato XML. A partir do arquivo XML foi gerado um conjunto de arquivos em formato CSV. Os arquivos CSV foram carregados para uma base de dados PostgreSQL, gerando 4 tabelas: uma tabela de termos, uma tabela de sinônimos, uma tabela de termos correlatos entre diferentes sistemas de terminologias e uma tabela que representa a hierarquia de conceitos do RadLex.

A hierarquia de conceitos do RadLex também foi exportada para a ferramenta *Protégé*, através de um *script* escrito em linguagem *Python*. O *script* consulta a base de dados do *RadLex* no PostgreSQL e monta o arquivo de definição de classes para o *Protégé*.

O *Protégé* serve como uma ferramenta de projeto, onde é possível analisar a hierarquia de conceitos do RadLex para facilitar a criação dos vínculos entre a Ontologia do RadLex e a Ontologia usada na rotulação dos laudos. Os vínculos são criados através da definição de relacionamentos entre as classes do RadLex e dos rótulos usados para anotação dos laudos. A Figura 7 mostra os níveis iniciais da hierarquia do RadLex na ferramenta *Protégé*.

No *Protégé* cada conceito foi definido como uma classe, respeitando a devida hierarquia de conceitos. Cada classe de conceito possui quatro *slots*, um para o código de conceito, uma para o código da hierarquia, um para o *string* que representa o nome do conceito e uma para indicar a língua da instância da classe de conceito.

Este modelo de classes pode não parecer adequado em uma primeira análise, pois todas as classes possuem a mesma estrutura e os mesmos *slots*, mas em seu tutorial sobre construção de ontologias Noy e McGuinness [Noy 2001] comenta a codificação em classes individuais no caso de ontologias baseadas em grandes hierarquias de termos de um dado domínio. Na realidade a estrutura hierárquica se apresenta como sendo o conteúdo mais importante neste caso.

A criação de uma classe para cada conceito facilita não só a exploração da hierarquia, mas também o mapeamento dos conceitos para os rótulos do sistema de anotação, pois mapeada uma super-classe para um rótulo, todas as subclasses são mapeadas para o mesmo rótulo.

No caso da ampliação da estrutura de dados para anotação de laudos, a definição das classes da ontologia, por possuir uma maior granularidade, permite definir mapeamentos mais detalhados.

Como já dito, foi adicionado um *slot* para identificar a língua utilizada pela cadeia de caracteres que armazena o nome do conceito. Com a estrutura descrita de classes, a intenção aqui é criar instâncias diferentes de cada classe de conceito para cada língua codificada na ontologia.

Como a versão inicial do RadLex não é completa, optou-se por complementar o vocabulário com o uso do UMLS, até que seja obtida uma versão mais completa. Pretende-se manter a estrutura básica do RadLex, mas ampliar o vocabulário com termos do UMLS,

apenas nos casos onde um termo não foi reconhecido com o RadLex. Espera-se encontrar com o UMLS um maior número de definição de doenças, por exemplo. A desvantagem do UML é a dificuldade de se extrair uma hierarquia concisa, visto que este agrega diversos vocabulários distintos.

Os conceitos do UMLS também estão armazenados em uma base de dados PostgreSQL, sendo que no caso do UMLS os dados do *Metathesaurus* já são obtidos no formato CSV. Para agilizar as buscas pelos *strings* em português na base de dados, foi criada uma tabela extra contendo apenas os *strings* em português, uma vez que a tabela original de conceitos, a *mrconso*, contém cerca de dois milhões de linhas, considerando os vocabulários instalados nas línguas portuguesa e inglesa.

4.5.1 Estrutura Relacional de Armazenamento do RadLex

Como já mencionado, os dados referentes ao sistema de terminologias do RadLex foram importados para um banco de dados relacional. A Figura 8 representa a estrutura de dados relacional criada para armazenar os termos e a hierarquia do RadLex.

Figura 8: Estrutura relacional de armazenamento do RadLex

4.5.2 Sumário de Conceitos do Radlex

A Tabela 14 mostra um resumo das características do vocabulário do RadLex. Os dados foram obtidos por meio de consulta SQL no banco de dados descrito pelo modelo do item 4.5.1.

Tabela 14: Sumário de conceitos do RadLex

	Variável	Valor
1	Quantidade total de conceitos	1326
2	Relacionamentos de parentesco	1302
3	Quantidade total de <i>strings</i> idênticos (mas com código de conceito diferente)	17
4	Quantidade total de conceitos mapeados para outros vocabulários	333
4.1	Quantidade de termos mapeados com o UMLS	216
4.2	Quantidade de termos mapeados com o vocabulário da ACR	93
4.3	Quantidade de termos mapeados com o próprio RadLex	24
5	Quantidade de sinônimos	24

4.5.3 Índice de Sentenças Normalizadas e Índice de Palavras Normalizadas

Em Português o vocabulário do UMLS não possui os índices de sentenças e palavras normalizadas, no entanto esses índices são importantes para os algoritmos de processamento de linguagem natural, pois para comparar dois *strings* que representam um conceito, por exemplo, é necessário que ambos utilizem os mesmos critérios de formatação.

Um índice normalizado além de eliminar variações entre letras minúsculas e maiúsculas, deve considerar apenas os radicais de cada palavra contida em uma sentença e deve descartar as *stop words*. Resumidamente cada conceito UMLS normalizado deve preservar o significado principal, descartando as flexões e as palavras que não adicionam sentido à sentença.

Por exemplo, a expressão (1) “aorta alongada” tem o mesmo significado que (2) “alongamento da aorta”. Se extraíssemos os radicais das palavras em cada sentença e removêssemos as *stop words* teríamos as seguintes sentenças normalizadas: (1) “aort along” e (2) “along aort”. Se usássemos como critério de comparação entre sentenças a contagem de palavras semelhantes, concluiríamos que ambos as sentenças possuem o mesmo conteúdo semântico, mas se comparássemos as sentenças originais literalmente, caractere a caractere, concluiríamos que se tratam sentenças com significados diferentes. A partir do exemplo concluímos que o índice normalizado de conceitos é necessário para melhorar o desempenho dos algoritmos de comparação entre sentenças que representam conceitos.

O primeiro passo foi criar duas tabelas, uma para os conceitos do UMLS, onde todos os *strings* foram armazenados em letra minúscula e uma para as todas as palavras existentes nos *strings*, também em letra minúscula. As tabelas foram derivadas dos arquivos fonte MRXW_POR.RRF, para a lista de palavras e MRCONSO.RRF, filtrada apenas para *strings* em português, para a lista de *strings* de conceitos.

Optou-se por não armazenar as palavras e conceitos normalizados, para que seja possível trocar o algoritmo de normalização conforme a necessidade da aplicação e para a realização de testes. Desta forma a normalização foi implementada como parte de uma biblioteca de utilitários para processamento de *strings*. Se as palavras e conceitos fossem armazenados de forma já normalizada, haveria um ganho de desempenho, porém seria necessário recriar o índice a cada sequência de testes ou troca de algoritmo. Em uma aplicação real seria recomendável armazenar os *strings* já normalizadas.

O algoritmo para normalização utiliza as seguintes etapas:

- ✓ Transformar o *string* para letra minúscula;
- ✓ Remover sinais de pontuação;
- ✓ Remover caracteres acentuados e o 'ç';
- ✓ Tokenizar o *string*;
- ✓ Remover as *stop words*;
- ✓ Submeter cada palavra do *string* ao *Snowball Stemmer*;

A premissa básica foi remover dos *strings* os sinais que não influenciam fortemente no significado de um conceito, assim se optou por usar letras minúsculas, eliminar a pontuação e as *stop words*. Foi utilizada como referência a lista de *stop words* em português contida no pacote do NLTK. A Figura 9 representa o processo de normalização de um *string*.

Figura 9: Passos da normalização de um *String*

Os caracteres acentuados também foram eliminados, pois normalmente encontramos bases de dados cuja aplicação que as alimenta não é internacionalizada. Essas bases acabam por armazenar os textos em formato ASCII desconsiderando os acentos e o 'ç' em português.

O *Snowball Stemmer* [Snowball] é um *stemmer* implementado em diversas línguas, inclusive em português, disponível para *download* na *Internet*. Implementado originalmente

em linguagem C, mas utilizado em uma aplicação *Python* neste trabalho através da utilização de um *Wrapper* chamado *PyStemmer* [PyStemmer] também disponível na *Internet*.

Outro benefício da criação de um utilitário de normalização de *strings*, é a possibilidade de utilizá-lo posteriormente em outros pontos da aplicação de extração de informação, como por exemplo para processar os conceitos do RadLex ou dos próprios laudos.

4.6 Mapeamento Entre Rótulos dos Laudos e Ontologia Médica

Definidos o modelo de documento anotado e a ontologia a ser utilizada para anotação dos textos, foi necessário criar um mapeamento entre classes de termos da ontologia e os rótulos a serem utilizados para anotação dos laudos.

O mapeamento foi criado segundo a estrutura da versão inicial do RadLex, mas deverá ser refeito futuramente para acomodar a estrutura da nova versão do RadLex.

Optou-se por trabalhar com uma estrutura de documento do laudo simples, para facilitar o mapeamento e para criar um documento XML legível para o usuário. Outro aspecto importante da estrutura simplificada de rótulos e mapeamento, é que a programação de algoritmos baseados em regras, para implementação do passo 5 do processo de notação do laudo (conforme Figura 6).

A Tabela 15 resume o mapeamento entre a ontologia do RadLex e a estrutura proposta para anotação dos laudos.

Tabela 15: Mapeamento entre estrutura de laudos e Radlex

Laudo	Radlex	Mapeamento	Exceção	Estrutura Sintática
Achado / Função	-	-	-	Sintagmas com núcleo substantivo
Achado / Processo	RL800 - Morphologic and Physiologic Processes	Conceito e descendentes		Sintagmas com núcleo substantivo
Achado / Enfermidade	RL956 - Diagnoses and Etiologies	Conceito e descendentes	Sub-árvore RL931	Sintagmas com núcleo substantivo
Elemento / Anatomia	RL29 - Anatomic Location	Conceito e descendentes		Sintagmas com núcleo substantivo
Elemento / Objeto	RL931 - Foreign Body	Conceito e descendentes		Sintagmas com núcleo substantivo
Elemento / Característica	RL598 - Finding Related Features	Conceito e descendentes		Normalmente adjetivos
Elemento / Visualização	RL765 - Modality Related Features RL797 - Artifacts	Conceito e descendentes		Sintagmas com núcleo substantivo Sintagmas com núcleo substantivo
Conclusão / Diagnóstico	RL956 - Diagnoses and Etiologies	Conceito e descendentes	Sub-árvore RL931	Sintagmas com núcleo substantivo

Considerando a estrutura dos laudos, o rótulo Função não foi mapeado para classe alguma do Radlex, pois na versão inicial do vocabulário não foram encontradas classes que satisfizessem a semântica do rótulo. Para a ontologia definida para os laudos, uma função poderia ser conceitos do tipo “batimentos cardíacos” ou “respiração”, por exemplo. Este tipo de conceito normalmente não é descrito em laudos de Raio-X de Tórax, no entanto foi mantido o rótulo, pois na eventual troca de escopo na área médica, o rótulo pode ser necessário.

Uma dificuldade encontrada no mapeamento, foi a diferenciação entre o que é uma “característica” e o que é uma “visualização”. O valor semântico dos rótulos é muito próximo, porém o primeiro trata de característica observadas em um achado ou anatomia, por exemplo, se temos o conceito “aorta alongada”, alongada é uma característica da região anatômica aorta. Não importam a modalidade, as características do aparelho usado na aquisição ou qualidade da imagem, trata-se de uma alteração anatômica, que sempre será observada. Com relação a visualização, tem-se em mente características visuais como transparência ou textura, que poderiam variar segundo a modalidade de exame ou qualidade da imagem. Sintaticamente observa-se que “características” normalmente são adjetivos e “Visualização” são substantivos.

Considerando a estrutura do RadLex, segundo a versão inicial, ambos os tipos de conceitos encontram-se dispersos sob uma mesma estrutura de sub-árvore, o que dificulta o mapeamento entre RadLex e Laudos. Optou-se por adotar o mapeamento da Tabela 15, que acaba por flexibilizar o conceito de “Visualização”, que acabou incorporando mais classes semânticas.

A fronteira entre achados e diagnósticos também é tênue, pois se identificado um achado, como pneumonia por exemplo, ele também se enquadraria com diagnóstico. Apesar dos laudos de Raio-X de Tórax não apresentarem uma seção de diagnósticos o rótulo foi mantido para futuras adaptações da aplicação referentes à mudança na estrutura dos laudos ou adaptação para outras modalidades que explicitem o diagnóstico.

4.7 Algoritmos

4.7.1 Busca de *Strings*

Os conceitos médicos devem estar organizados segundo uma ontologia que representa o conhecimento em uma estrutura hierárquica de conceitos e relacionamentos. A ontologia então é mapeada em uma estrutura de dados para armazenamento em um banco. A partir do armazenamento, os conceitos passam a existir como registros em um banco relacional que nada mais são que uma coleção de *strings*, códigos e relacionamentos entre registros. Os dados armazenados só são úteis se houver uma forma eficiente de recupera-los para então os reprocessar de forma a gerar a informação procurada.

No caso deste trabalho, a base de dados da ontologia deve servir de referência para o reconhecimento de *strings* como representantes de um determinado conceito médico. Ou seja, dado um *string* de entrada o problema se resume a encontrar o *string* da base de dados que mais se assemelhe ao *string* de entrada. Através do *string* encontrada é possível identificar o conceito que a entrada representa.

A simples comparação entre *strings* caractere a caractere, não é eficiente, pois em linguagem natural existem diversas formas gráficas para representar uma mesma unidade de significado. Por exemplo, o conceito ‘acalasia do esôfago’ e ‘acalasia esofágica’ são *strings* graficamente diferentes mas que possuem o mesmo significado, isto é, representam o mesmo conceito médico.

A busca por *strings* representantes de um conceito médico no banco de dados tem por finalidade a identificação de trechos de um laudo como sendo a representação de um conceito médico descrito no sistema de ontologias utilizado. Resumidamente o problema encontrado neste ponto do trabalho foi a partir de um *string* de entrada, encontrar no banco de dados o *string* que mais se assemelhe a ele, identificando-se assim qual o conceito médico representado pelo *string* de entrada.

O algoritmo utilizado percorre o banco de dados comparando *string* a *string* e associando uma pontuação que indica a semelhança entre os dois *strings* comparados. Pares cuja pontuação esteja acima de um certo valor de corte são considerados como semelhantes.

A Figura 10 ilustra o processo de comparação e busca por conceitos no banco de dados que contém os conceitos da terminologia utilizada.

Figura 10: Comparação e busca por conceitos

O primeiro passo a ser tomado é normalizar a sentença de entrada, segundo o processo já descrito.

Como o banco de dados de conceitos é extenso, optou-se por filtrar os dados dos bancos através do índice de palavras. O filtro consiste em separar cada palavra existente no *string* normalizado e fazer a busca por todos os conceitos que contenham cada palavra do *string* de entrada. O filtro prévio dos conceitos diminui a quantidade de *strings* a serem comparados posteriormente. Por exemplo, em uma base como a do UMLS, utilizada neste trabalho, existem 104.320 *strings* diferentes em português. Se contássemos as repetições, este número chega a 133.839 *strings*. Como a busca deve passar por toda a base até encontrar o *string* de conceito que mais se assemelhe ao *string* de entrada, a busca seria mais lenta. O filtro serve para descartar os conceitos cujo *string* obterá pontuação nula na comparação, por não possuírem interseção em seus conjuntos de palavras. Após filtrar os conceitos, cada um deve ser normalizado para comparação.

Tendo o *string* de entrada normalizado e uma lista de conceitos normalizada, é feita a comparação *string* a *string* de forma similar a proposta pelo algoritmo *Bleu* [Papineni 2002], usado para avaliação de tradução automática. A comparação é feita através da quantidade de palavras idênticas mais o comprimento das frases, isto é, cada par de *strings* recebe uma

pontuação calculada a partir no número de palavras idênticas mais uma pontuação referente a diferença de comprimento entre os *strings*. A ordem das palavras não é considerada para a comparação.

Equação 13: Cálculo da pontuação de semelhança para comparação de *strings*.

$$pontuação = 0,2 * (1 - (\frac{|len(ref) - len(ent)|}{len(ref)})) + 0,8 * (\frac{len(ref \cap ent)}{len(ref)})$$

A Equação 13 mostra como é feito o cálculo da pontuação de semelhança entre o *string* de referência (ref) e o *string* de entrada (ent). A função *len* retorna o comprimento de um *string*. Durante a busca dos conceitos, o *string* de referência é o *string* buscado na base de conceitos médicos. O *string* de entrada é o *string* que representa o conceito médico a ser comparado. Conforme o que observamos na Equação 13, a pontuação de semelhança é composta por vinte por cento referente à diferença de comprimento e oitenta por cento referente a interseção de palavras entre os *strings* comparados. Lembramos que aqui a comparação é feita através dos *strings* normalizados, ou seja, sem sinais de pontuação, sem *stop words* e com apenas os *stems* de cada palavra.

A saída do algoritmo é um código de conceito acompanhado de uma pontuação que varia de zero a um. Quando a pontuação aproxima-se do zero, significa que os dois *strings* comparados são bastante diferentes, mas quando a pontuação aproxima-se de um, os *strings* comparadas são semelhantes.

É aconselhável escolher uma pontuação de corte para a lista de saída do algoritmo para eliminar os conceitos de pontuação muito baixa, ou seja, irrelevantes.

Apenas adiantando uma das deficiências deste algoritmo, ele não considera a existência de palavras sinônimas, isto significa que o conceito ‘Cardiomegalia’ não seria encontrado se procurássemos por ‘Coração aumentado’. Apesar dos dois *strings* representarem o mesmo conceito, teriam uma pontuação de semelhança nula.

Esta metodologia de busca independe da estrutura do sistema de terminologias e poderia ser aplicada na busca por sentenças em qualquer vocabulário com um mínimo de estrutura organizacional.

4.7.2 Segmentação de Sentenças (Extração de Sintagmas)

A segmentação das sentenças foi usada para identificar trechos de texto candidatos a conceitos nas ontologias usadas. Considerando, neste trabalho, que uma sentença é um conjunto de palavras agrupadas segundo regras sintáticas, contendo ao menos um substantivo, partiu-se do princípio que uma sentença possui trechos principais que expressam o valor semântico do texto e componentes acessórios, que satisfazem as exigências sintáticas da linguagem escrita. O Exemplo 19 mostra a estrutura do que é considerado como sentença neste trabalho.

Exemplo 19: Sentenças extraídas dos laudos de Raio-X de Tórax.

```
Osteofitose/NN dorsal/JJ./.  
Marcapasso/NN com/IN um/CD eletrodo/NN intracavitário/JJ./.  
Parênquima/NN pulmonar/JJ de/IN transparência/NN normal/JJ./.  
Hilos/NNS de/IN configuração/NN, topografia/NN e/CC dimensões/NNS normais/JJS./.
```

A definição aqui utilizada é baseada na estrutura de texto encontrada nos laudos, onde praticamente não há verbos. Observou-se que cada sentença é constituída por um termo principal, sempre da classe substantiva (rotulado como NN/NNS) e termos que o completam ou qualificam, normalmente da classe adjetiva (rotulados como JJ/JJS) ou numerais (cardinais, rotulados como CD ou ordinais, rotulados como OD). Preposições, artigos, conjunções são considerados termos acessórios, sendo que apenas em alguns casos específicos adicionam valor semântico a expressão. Citamos como exemplos destes casos os termos marcados com o símbolo “*”, que denotam a negação, ou os termos marcados como advérbios, que possuem a função de intensificar ou diminuir o valor semântico da expressão que acompanham.

Adjetivos encontrados sozinhos em uma sentença, como no caso de expressões do tipo “((infiltrado/JJ?/?))”, são considerados como parte da sentença maior, apesar da delimitação imposta pelos sinais de pontuação.

No caso de termos marcados com o rótulo CC, apesar de não agregarem valor semântico, possuem um papel sintático importante que nos permite identificar quando há uma ligação entre termos próximos, por exemplo se temos um conjunto de substantivos separados por vírgula e o termo “e/CC”, sabemos que o adjetivo subsequente se refere a todos os

substantivos da lista, portanto no momento da segmentação do texto, deve ser aplicado a todos os substantivos.

Segundo a estrutura de texto descrita, neste trabalho segmentar um texto, significa quebra-lo em segmentos menores que constituem uma unidade de significado.

Seguindo as premissas descritas, a abordagem para segmentação de textos seguiu os passos descritos abaixo:

- ✓ Aplicação do *Brill Tagger*, treinado com um *Corpus* de frases extraídas dos laudos de Raio-X de Tórax, ao texto de entrada;
- ✓ Divisão do texto sentenças delimitadas por pontuação ou quebra de linha;
- ✓ Divisão de cada sentença em segmentos mínimos.
- ✓ Agrupamento dos segmentos mínimos em segmentos que carreguem uma unidade semântica, como por exemplo, “Pulmão direito superior”;

O primeiro passo consistiu em aplicar a implementação do *Brill Tagger* existente no NLTK ao texto de entrada do processo de segmentação. A rotulação feita pelo *Brill Tagger* serve para identificar as classes sintáticas de cada palavra ou sinal do texto e os sinais de pontuação.

Através das marcas feitas pelo *Brill Tagger*, é possível separar as sentenças de texto baseando-se no sinais de pontuação ou quebras de linha. Sinais como ponto final, interrogação ou exclamação, indicam o término de uma sentença. No caso considerou-se apenas os pontos como final de sentença.

O terceiro passo foi executado através da utilização do *Chunk Parser* existente no NLTK [NLTK 2004]. O processo de *Chunk parsing*, é utilizado para identificar segmentos de texto importantes ou irrelevantes. Um exemplo de utilização de *Chunk parsing* é a identificação de sintagmas nominais. O objetivo aqui era identificar sintagmas nominais.

A montagem do *chunk parser* para extração de sintagmas nominais baseou-se nos trabalhos de Perini [Perini 2005]. Porém, como a definição de sintagma nominal é complexa e apresenta pontos ainda em estudo, optou-se por utilizar uma simplificação das regras propostas por Perini para montagem de regras para o *Chunk Parser*, permitindo a identificação de segmentos mínimos no texto. Considerou-se também o fato de que a

montagem de regras complexas como as definidas por Perini não é viável para a estrutura baseada em regras do *Chunk Parser* do NLTK. Como o objetivo deste passo é identificar os trechos principais de texto, o *Chunk Parser* serviu ao propósito. A saída desta fase de processamento é uma lista de sentenças, segmentada em uma árvore de pequenos sintagmas e os demais termos da sentença.

Nesta fase foram aplicadas as seguintes regras ao algoritmo de *chunk parsing* do NLTK, na ordem dada:

Tabela 16: Regras usadas no *Chunk Parser* do NLTK

	Regras para o <i>Chunk Parser</i>	Estrutura do sintagma
1	MergeRule('<CD OD PP\$><NN NNS>', '<JJ JJS>', 'SN2'))	PV + Núcleo + ModI
2	ChunkRule('<NN NNS><JJ JJS>?', 'SN3'))	Núcleo + ModI
3	ChunkRule('<RB.*><JJ JJS>+<RB>?', 'QL1'))	ModI + ModE
4	ChunkRule('<JJ JJS>+<RB>', 'QL2'))	ModE ou ModI
5	ChunkRule('<RB>', 'ADV'))	-
6	ChinkRule('<.,>+', 'CK1'))	-
7	ChinkRule('<VBD VBG CC IN DTI DT AT>+', 'CK2'))	-

A sintaxe de definição das regras é semelhante a definição de expressões regulares, porém os rótulos são vistos como unidades. Mais detalhes sobre a definição das regras de *chunking* consultar a documentação do NLTK [NLTK 2004].

O Exemplo 20 mostra algumas sentenças de entrada e a respectiva saída serializada da fase de *chunk parsing*. As expressões marcadas como “SINT” são as expressões principais e as marcadas como “DESC” são considerados termos acessórios, na maioria das vezes não considerados como aditivos de valor semântico.

Exemplo 20: Aplicação do *Chunk Parser*

abaulamento/NN do/IN arco/NN médio/JJ ./.	SINT: abaulamento/NN DESC: do/IN SINT: arco/NN médio/JJ DESC: ./.
acentuação/NN da/IN cifose/NN dorsal/JJ fisiológica/JJ ./.	SINT: acentuação/NN DESC: da/IN SINT: cifose/NN dorsal/JJ SINT: fisiológica/JJ DESC: ./.
aorta/NN não/RB-* avaliavel/JJ ./.	SINT: aorta/NN SINT: não/RB-* avaliavel/JJ DESC: ./.
vascularização/NN pulmonar/JJ,/, hilar/JJ e/CC periférica/JJ aumentada/JJ./.	SINT: vascularização/NN pulmonar/JJ DESC: ,/, SINT: hilar/JJ DESC: e/CC SINT: periférica/JJ SINT: aumentada/JJ DESC: ./.

O ultimo passo da segmentação é o reagrupamento dos segmentos mínimos de forma a obter as unidades de significado para identificação de termos do Radlex. Nesta fase são identificados os conjuntos de segmentos mínimos que devem ser agrupados ou mantidos separados, aplicando-se regras de maior complexidade que a fase apenas de *chunk parsing*. O Exemplo 21 mostra sentenças de entrada com suas respectivas saídas segmentadas.

Exemplo 21: Sentenças segmentadas

Entrada:	abaulamento/NN do/IN arco/NN médio/JJ ./.
Saída:	<pre> <SENT ORIG="abaulamento do arco médio ."> <SEG>abaulamento/NN</SEG> <COMP>do/IN</COMP> <SEG>arco/NN médio/JJ</SEG> <COMP>./.</COMP> </SENT> </pre>
Entrada:	Espessamento/NN pleural/JJ ápico/JJ lateral/JJ à/IN direita/NN ./.
Saída:	<pre> <SENT ORIG="espessamento pleural ápico lateral à direita ."> <SEG>espessamento/NN pleural/JJ ápico/JJ lateral/JJ</SEG> <COMP>à/IN</COMP> <SEG>direita/NN</SEG> <COMP>./.</COMP> </SENT> </pre>
Entrada:	aorta/NN não/RB-* avaliavel/JJ ./.
Saída:	<pre> <SENT ORIG="aorta não avaliável ."> <SEG> <SEG>aorta/NN</SEG> não/RB-* avaliável/JJ </SEG> <COMP>./.</COMP> </SENT> </pre>
Entrada:	vascularização/NN pulmonar/JJ hilar/JJ e/CC periférica/JJ aumentada/JJ./.
Saída:	<pre> <SENT ORIG="vascularização pulmonar , hilar e periférica aumentada ."> <SEG> <SEG>vascularização/NN pulmonar/JJ</SEG> <COMP>,/</COMP> <SEG>vascularização/NN hilar/JJ</SEG> <COMP>e/CC</COMP> <SEG>vascularização/NN periférica/JJ</SEG> </SEG> <COMP>aumentada/JJ</COMP> <COMP>./.</COMP> </SENT> </pre>

Cada segmento é marcado pelo par de rótulos <SEG>, podendo haver seqüências aninhadas de segmentos. Cada sentença é marcada com o par de rótulos <SENT>

Neste passo temos regras de união simples de segmentos mínimos ou de composição de segmentos mínimos. A união consiste apenas no reagrupamento dos termos mínimos, porém mantendo as marcas que delimitam os termos mínimos. A composição consiste na criação de uma seqüência nova de termos mínimos, baseada em um critério de distribuição e reagrupamento de palavras. A composição acontece na ocorrência de um conectivo como “/,” ou “e/CC”. Podemos observar um exemplo de composição no ultimo item mostrado no Exemplo 21.

O reagrupamento dos segmentos mínimos é feito por um algoritmo baseado em regras. O algoritmo busca por um padrão de segmentos mínimos e os reagrupa para obter um novo segmento. O Exemplo 22 mostra os casos de reagrupamento:

Exemplo 22: padrões de reagrupamento de *chunks*

<p>(1) SE: chunk 1 = SINT: (NN NNS) (JJ JJS) S1 = (NN NNS) S2 = (JJ JJS) chunk 2 = DESC: ,/, chunk 3 = SINT: (JJ JJS) S3 = (JJ JJS) chunk 4 = DESC: ,/, chunk 5 = SINT: (JJ JJS) S5 = (JJ JJS) *** chunk N - 1 = DESC: ,/, "e/CC" chunk N = SINT: (JJ JJS) Sn = (JJ JJS) SAÍDA => <SEG><SEG>chunk 1</SEG><SEG>S1 chunk 3</SEG> ... <SEG>S1 chunk N</SEG></SEG></p>
<p>(2) SE: chunk 1 = SINT: RB chunk 2 = SINT: (NN NNS) (JJ JJS)* SAÍDA => <SEG><SEG>chunk 1</SEG><SEG>chunk 2</SEG></SEG></p>
<p>(3) SE: chunk 1 = SINT: (NN NNS) (JJ JJS)* chunk N+1 = SINT: (JJ JJS)+ (RB-*) SAÍDA => <SEG><SEG>chunk 1</SEG> chunk 2</SEG></p>
<p>(4) SE: chunk 1 = SINT: (NN NNS) (JJ JJS)* chunk 2 = SINT: RB[-*] (JJ JJS)+ SAÍDA => <SEG><SEG>chunk 1</SEG> chunk 2 </SEG></p>
<p>(5) SE: Chunk 1 = SINT: (JJ JJS)? (NN NNS) (JJ JJS)* Chunk 2 = SINT: (JJ JJS)+ Chunk 3 = SINT: (JJ JJS)+ Chunk 4 = SINT: (JJ JJS)+ *** Chunk N = SINT: (JJ JJS)+ SAÍDA => <SEG><SEG>Chunk 1</SEG><SEG>Chunk 2</SEG><SEG>Chunk 3<SEG>...<SEG>Chunk N</SEG></SEG></p>
<p>(6) SE: Chunk 1: SINT: RB Chunk 2: SINT: IN Chunk 3: SINT: (CD OD) (NN NNS) SAÍDA => {{Chunk 1}{Chunk 2}{Chunk 3}}</p>
<p>(7) SE: Chunk 1: SINT: RB Chunk 2: SINT: (CD OD) (NN NNS) SAÍDA => <SEG><SEG>Chunk 1</SEG><SEG>Chunk 2</SEG></SEG></p>

O reagrupamento trabalha testando inicialmente se um *chunk* é do tipo SINT ou DESC. O reconhecimento deve obedecer a seqüência de SINTs e DESCs descrita em cada caso. Em seguida cada *chunk* é testado segundo a seqüência de palavras que contém, onde o que importa é o rótulo sintático e não a palavra em si. Por exemplo, se um *chunk* é descrito pelo padrão “SINT: NN|NNS (JJ|JJS)*”, significa que o *chunk* deve ser do tipo SINT e que deve possuir uma seqüência de onde a primeira palavra é um substantivo (NN ou NNS) seguido por zero ou mais adjetivos (JJ ou JJS). Procurou-se aqui usar os operadores de expressões regulares para definir os casos.

Quando uma seqüência de *chunks* se enquadra em um dos casos, é gerada uma saída onde os *chunks* são reagrupados da forma descrita no Exemplo 22: padrões de reagrupamento de *chunks*. É importante atentar para o detalhe de que o teste de cada caso deve ser feito na

ordem dada. Normalmente temos que testar primeiro as expressões maiores ou as expressões que tem primazia em relação as demais.

Quando um grupo de *chunks* é enquadrado em um dos casos, o grupo formatado segundo a descrição do caso é anexado à sentença de saída, porém quando um *chunk* não se enquadra a sequência alguma, este é simplesmente anexado ao *string* da sentença de saída e o processamento se move para o próximo *chunk*.

Ao fim do processamento, obtemos um arquivo de saída que segue o padrão de documento do Exemplo 17. Este arquivo é formado por sentenças com seus respectivos segmentos.

4.7.3 Identificação de Conceitos

Após identificar os segmentos que formam cada sentença de um laudo, estes segmentos tornam-se candidatos a conceitos a serem identificados no sistema de ontologia utilizado. Para tanto é necessário ler segmento a segmento do texto de um laudo buscando cada um dos segmentos candidatos no banco de dados que armazena a ontologia. Esta fase do processo corresponde ao passo 4 da Figura 6.

O processamento ocorre inicialmente lendo cada sentença do arquivo intermediário gerado pelo algoritmo de segmentação de textos. A leitura do arquivo de entrada é feita através da biblioteca *Minidom* existente na distribuição do *Python*.

Carregada uma sentença, cada segmento existente na sentença é submetido ao algoritmo de busca por conceitos, descrito anteriormente, sempre procurando-se identificar o maior *string* possível. Caso o segmento não seja encontrado na base de dados de terminologias, é realizada uma operação de corte no *string* inicial, onde as palavras que não são substantivos são eliminadas uma a uma e o *string* resubmetido à busca, até que seja encontrado um conceito no banco de dados ou que o *string* tenha sido totalmente podado mas não se tenha encontrado conceito algum.

A idéia do corte é que a palavra principal em um conceito é sempre um substantivo, sendo que as demais são acessórios que complementam o significado de um conceito. Por exemplo, o conceito “cifose dorsal fisiológica” tem como centro do significado o substantivo

“cifose”, que denota o achado principal. Se o *string* fosse cortado até obtermos somente a palavra “dorsal”, teríamos noção de uma anatomia, porém referente a que achado?

Quando um *string* é submetido à busca, inicialmente é feita a busca por conceitos do RadLex. Se nada foi encontrado no Radlex, o mesmo processo é aplicado novamente porém a busca agora é feita no UMLS. Optou-se em buscar os segmentos não identificados no UMLS, pois o vocabulário da versão inicial do RadLex é incompleto e toda informação que se consiga adicionar ao laudo de saída pode vir a ser útil em um momento futuro.

O arquivo de saída gerado segue o modelo descrito no Exemplo 18, onde se observa que cada sentença agrupa uma coleção de conceitos identificados.

O ultimo passo do processamento segundo a Figura 6, seria a anotação do laudo em si, porém esta etapa não foi completada ficando para trabalhos futuros.

5 IMPLEMENTAÇÃO

A implementação do trabalho proposto envolve diversas etapas, algumas independentes. Quando agrupadas as ferramentas e os métodos desenvolvidos em cada etapa, devemos ter como resultado final o processamento dos laudos para rotulação.

As seções do texto a seguir descrevem as tarefas executadas e as pendências para que possamos alcançar o objetivo final do trabalho.

5.1 Estruturação de um Sistema de Ontologia Médica

Os sistemas de Ontologias usados, Radlex e UMLS foram obtidos através da *Internet* e importados para um banco de dados relacional, o PostgreSQL 8.0, segundo os métodos já descritos.

O RadLex foi importado também para a aplicação *Protégé*, servido de base para a modelagem da ontologia referente à anotação de laudos e para estudo da estrutura hierárquica de um sistema de ontologias. A ontologia do Radlex, por fim, foi traduzida automaticamente e revisada manualmente para montagem do vocabulário em português usado no desenvolvimento das ferramentas de *PLN* deste trabalho.

Como o Radlex utilizado não está completo, a idéia inicial seria de complementar o vocabulário com termos do UMLS ou da organização de conceitos extraídos a partir dos próprios laudos, no entanto, com a expectativa da liberação da nova versão do RadLex, optou-se por apenas complementar o trabalho com o uso do UMLS.

5.1.1 Avaliação de Ontologias

O método de avaliação de ontologias descrito no capítulo referente à metodologia foi aplicado aos UMLS, sendo que seus resultados serão descritos no capítulo de discussão de resultados. A aplicação do método ao RadLex, deverá ser feita futuramente para avaliação da versão mais recente, assim que esta for traduzida para o português, segundo o mesmo método utilizado para a tradução da versão inicial da ontologia.

5.2 Corpus de Treinamento e Teste

5.2.1 Extração dos Laudos da Base Hospitalar e Armazenamento no Banco de Dados do Projeto

Os laudos foram extraídos da base do InCor através de consultas SQL simples, filtrando-se apenas por intervalos de data e eliminando-se a identificação do paciente. Extraídos os laudos, estes são passados por um *script* que faz a limpeza, através de expressões regulares, dos excessos de espaços em branco e remove a formatação em colunas de 80 caracteres. Cada laudo foi salvo em um arquivo de texto comum.

Após a extração da base hospitalar, os textos foram pré-processados e organizados por linha de laudo, isto é, cada frase é armazenada separadamente das demais, mas com uma identificação do laudo e da sequência de apresentação no texto. Para extrair as frases foi usado um algoritmo simples baseado em expressões regulares que identifica o ponto final em cada frase do texto.

Foram recebidas duas remessas de dados, a primeira contém cerca de 1300 laudos enquanto que a segunda remessa contém cerca de 1500 laudos. Ambas as remessas foram organizadas de forma semelhante, mas são mantidas em esquema diferentes no banco de dados, apenas por critério de organização, visto que a primeira remessa não contém referência de data.

5.2.2 O Corpus de Treinamento

O *Corpus* de treinamento foi montado antes do recebimento da segunda remessa de laudos, portanto contém dados apenas da primeira remessa. Entre os laudos existentes na base da primeira remessa, o critério de seleção para a composição do *Corpus* de treinamento foi através da coluna de id do laudo na base. A separação foi feita com base no valor médio encontrado entre os identificadores dos laudos, tendo sido selecionados para o *Corpus* de treinamento todos os laudos cujo identificador fosse menor que o valor médio. Os demais foram deixados para o *Corpus* de teste. Com este critério os laudos da primeira remessa de dados foram divididos aproximadamente na metade entre os dois *Corpus*.

A estrutura de armazenamento de dados permitiu a identificação de muitas frases idênticas entre os laudos, sendo assim optou-se por fazer a rotulação sintática dos laudos, por

frases ao invés de rotular os laudos, medida que visou minimizar o esforço manual de anotação.

As sentenças extraídas dos laudos foram anotadas manualmente com informação sintática. O trabalho de anotação foi realizado por um profissional da área de letras com experiência na criação e anotação de *Corpus*, inclusive na área de Biomedicina. As sentenças anotadas foram então armazenadas no banco de dados de laudos, sendo mantida a referência para a sentença original. Os textos podem então ser recompostos tanto sem rótulos sintáticos como rotulados. O Exemplo 23 mostra uma sentença anotada sintaticamente.

Exemplo 23: Senteça extraída de um laudo e rotulada sintaticamente.

acentuação/NN da/TN cifose/NN dorsal/JJ fisiológica/JJ./.

Foram usadas os rótulos do *Brown Corpus* [Brown 1979], considerando as devidas adaptações. Como os rótulos utilizados no *Brown Corpus* foram criados para a língua inglesa, nem todos são apropriados para o português, havendo a necessidade de uma pequena adaptação. O rótulo JJS, originalmente usado para Adjetivos Superlativo Semântico, foi usado para adjetivos no plural, visto que não há rótulos para esta classe sintática no *Brown Corpus*. Optou-se por manter os rótulos do *Brown Corpus* por questão de padronização de aplicações, visto que este conjunto de rótulos é amplamente utilizado, tornando-se mais fácil a eventual troca da implementação do rotulador ou de outra aplicação que utilize a saída do rotulador. O conjunto de rótulos utilizados é descrito no Anexo 11.4.

A Tabela 17 mostra algumas características do *Corpus* de treinamento:

Tabela 17: Descrição do *Corpus* anotado

	Valor
Total de laudos	589
Total de frases	4999
Total de frases diferentes	508
Quantidade média de frases por laudo	8,48
Quantidade máxima de frases por laudo	12
Quantidade mínima de frases por laudo	1
Quantidade de <i>tokens</i> no <i>Corpus</i> (considerando as 508 frases diferentes)	3786

5.2.3 O *Corpus* de Teste

Inicialmente pretendia-se usar a segunda metade dos laudos da primeira remessa de dados, para a montagem do *Corpus* de teste, porém com a obtenção da segunda remessa de dados, o *Corpus* de teste foi montando usando os laudos deste novo conjunto de dados.

A Tabela 18 resume as características do *Corpus* de teste utilizado.

Tabela 18: Descrição do *Corpus* de teste

	Valor
Total de laudos	1.490
Total de frases	20.927
Total de frases diferentes	864
Quantidade média de frases por laudo	12
Quantidade máxima de frases por laudo	18
Quantidade mínima de frases por laudo	1
Quantidade de <i>tokens</i> no <i>Corpus</i> (considerando as 864 frases diferentes)	6984

Como o vocabulário usado nos laudos segue normas do InCor, há uma taxa de repetição entre frases encontradas no vocabulário do *Corpus* de treinamento e da segunda remessa de dados, que constitui o *Corpus* de teste. Fazendo a comparação simples, *string* a *string*, foi encontrado um total de 153 frases iguais entres os dois *Corpus*, no entanto esse número é maior, pois apenas a existência de um espaço em branco a mais já torna duas frases diferentes para uma consulta feita em SQL na base onde estão armazenados os laudos.

A existência de uma interseção entre os dois *Corpus* deve aumentar o desempenho do rotulador, que uma vez treinado com o *Corpus* de treinamento, tenderá a errar menos as frases pertencentes à área de interseção entre *Corpus*.

A interseção entre os *Corpus* é aceitável se pensarmos que na vida real, mesmo em textos referentes a outros domínios de conhecimento, sempre há expressões que se repetem em textos diferentes. Se a divisão entre os *Corpus* é feita por conjunto de textos, não há como evitar que existam itens idênticos em mais de um conjunto de textos.

Vale ressaltar que a divisão dos *Corpus* é baseada em conjuntos de laudos, e a quebra em sentenças é apenas um artifício para minimizar o trabalho manual executado, tanto no momento de anotação quanto na avaliação dos resultados.

5.3 POS-Tagger

Foi utilizado no trabalho a implementação do *Brill Tagger* distribuído com a ferramenta NLTK. Ao contrário do que se previa inicialmente, não foi necessário fazer adaptações do algoritmo para o funcionamento com textos em português. A implementação do rotulador no NLTK já utiliza os rótulos do *Brown Corpus*, sendo que tanto treinamento como o processo de rotulação sucederam sem problemas.

5.3.1 Treinamento

O processo de treinamento é simples. Inicialmente o *Corpus* de treinamento foi usado para treinar um rotulador mais simples, no caso um rotulador baseado em unigramas já existente no NLTK. O rotulador baseado em unigramas serviu de rotulador auxiliar para o *Brill Tagger*, visto que este trabalha em duas fases, onde temos uma rotulação inicial menos precisa que é refinada através do algoritmo de Aprendizado Baseado em Transformações do *Brill Tagger*.

Ambos os rotuladores treinados foram salvos, para serem carregados posteriormente no processamento de um texto. Foi necessário serializar os rotuladores treinados, pois não foi encontrada uma forma de salvar a tabela de frequência de palavras por rótulos do rotulador baseado em unigramas nem as regras geradas pelo *Brill Tagger*. Na realidade as regras poderiam ser salvas, mas para carregá-las seria necessário fazer um *parser* do arquivo de regras gerado.

O rotulador auxiliar pode ser de qualquer tipo, baseado em expressões regulares, unigramas ou trigramas. Optou-se por utilizar um rotulador baseado em unigramas pois o *Corpus* de treinamento não é grande e este produziu um resultado inicial considerado suficiente para alcançar um desempenho satisfatório com o *Brill Tagger*.

A avaliação do rotulador será discutida no capítulo referente aos resultados obtidos.

5.4 Algoritmos

5.4.1 Algoritmo de Comparação e Busca de Sentenças

O algoritmo de comparação e busca de sentenças no banco de dados das ontologias usadas foi implementado em linguagem *Python*, tendo sido usada a API *PyPgSQL* para conexão com o PostgreSQL e a biblioteca *PyStemmer* para extração dos radicais das palavras que constituem os *string* buscados.

A biblioteca *PyStemmer* foi alterada para correção de alguns casos onde os radicais estavam sendo extraídos erroneamente. Basicamente o aplicativo extrai radicais diferentes para o plural das palavras terminadas em “l” com referência ao respectivo singular. O mesmo

problema acontece para as palavras terminadas em “ção” e “ções” cujos radicais não combinam com os respectivos radicais das palavras no singular.

A abordagem usada para corrigir os problemas do *stemmer* foi de derivar a classe responsável por extrair os radicais e criar casos de exceção. Se a palavra processada se enquadra em um dos casos de exceção, é gerada uma saída sem que se passe pelo *stemmer* do *Snowball*, caso contrário a palavra é passada para o *Snowball* que extrai o devido radical.

A implementação do algoritmo funciona tanto para a base do UMLS quanto para a base do RadLex, sendo que para o RadLex foi usada a nova implementação do *Stemmer* e para o UMLS a implementação original, de forma a nos permitir a visualização entre os resultados de ambas as implementações.

5.4.2 Segmentação de Sentenças

A segmentação de sentenças foi implementada toda em *Python* com o auxílio do NLTK. A primeira fase da segmentação foi realizada com o uso do *Chunk Parser* do NLTK, enquanto que a segunda fase foi implementada especialmente para este trabalho.

A saída gerada está de acordo com a especificação de documento XML discutida no capítulo de Projeto, segundo o modelo oferecido no Exemplo 12.

5.4.3 Identificação de Conceitos

A identificação de conceitos também foi toda implementada em *Python*, sendo que utilizou-se da biblioteca *Minidom* para manipulação do arquivo XML de entrada. A saída gerada está de acordo com a especificação de documento XML do Exemplo 18.

5.5 Mapeamento de Modelos

O mapeamento entre os modelos da Ontologia médica e do Laudo anotado foi feito com base no RadLex. Não foi realizado um mapeamento com a estrutura hierárquica do UMLS pois está agrega diversos vocabulários médicos diferentes, portanto as diversas hierarquias são mantida na base do UMLS, o que a torna confusa.

5.6 Trabalhos a Serem Concluídos

5.6.1 Anotação Semântica de Textos

Através do aparato descrito nos itens anteriores, os textos de Raios-X de Tórax foram processados e anotados com rótulos que identificam um código de conceito, seja ele do RadLex ou UMLS.

Identificado um conceito, através da análise da posição onde se encontra na hierarquia do sistema de terminologias e dos conceitos adjacentes, o conceito é mapeado para uma das classes de rótulos pertencentes a estrutura do laudo.

Exemplificando, suponhamos que em um texto encontramos a expressão “aorta alongada”. A palavra aorta existe no vocabulário UMLS com o código de conceito C0003483, porém a palavra alongada não aparece em conceito algum do UMLS. Considerando o conceito C0003483 está abaixo do conceito C0002807, referente à anatomia, poderíamos concluir que a aorta refere-se a uma parte anatômica e é qualificada pela palavra alongada. Considerando a estrutura dos laudos, muito provavelmente ele estaria relacionado a uma observação e portanto a um achado. O conceito “aorta” não faria sentido como um diagnóstico.

Para esta abordagem, a estrutura criada para formatação do documento do laudo deve ser previamente relacionada com a estrutura hierárquica da ontologia médica utilizada. Este cruzamento de informações permite a identificação de quais ramos da árvore da ontologia utilizada podem preencher os rótulos da estrutura criada para descrição do laudo. Neste ponto é necessário o mapeamento entre ontologias.

Anotado um laudo com a informação dos conceitos, categorizados através de rótulos, segundo uma ontologia específica para o laudo, futuramente no momento de uma busca por laudos, poderíamos expandir esta busca para os conceitos de uma sub-árvore da hierarquia ou então com o auxílio dos rótulo, por exemplo, poderíamos encontrar todos os documentos que associam uma anatomia a uma determinada doença.

Anotado o laudo, este deve ser armazenado em uma base de dados relacional que seja capaz de mapear um documento estruturado no formato XML, pois apenas continuar

armazenado os documentos como CLOBs pode não ser muito eficiente do ponto de vista computacional.

Esta última fase de programação não foi implementada, ficando para desenvolvimento futuro.

6 RESULTADOS

6.1 Ontologias

Os resultados referentes as Ontologias se dividem em dois grupos: Modelo de Representação dos Laudos e Organização das Ontologias Médicas.

6.1.1 Modelo de Representação dos Laudos

Como já descrito no capítulo 4, referente a análise e projeto do trabalho, foi proposta uma estrutura de dados para ser preenchida pelo sistema de extração de informação segundo os moldes já descritos.

O modelo foi projetado através do estudo de bibliografias da área, da análise das ontologias utilizadas neste trabalho, tanto o UMLS como o RadLex e do auxílio de um profissional da área médica.

Este modelo é genérico o suficiente para ser usado em outras áreas da radiologia, caso não seja requisitado um maior grau de detalhamento do conteúdo. A estrutura criada permite que a informação mais detalhada seja fornecida através dos códigos de conceitos, que são agrupadas em uma forma mais adequada para facilitar o processamento por um algoritmo de busca ou indexação, por exemplo. No caso de ser necessário acrescentar mais informações ao modelo, este pode ser ampliado, mas de forma a manter a compatibilidade com as versões anteriores.

6.1.2 Organização das Ontologias Médicas

Os resultados referentes às ontologias médicas tem caráter organizacional. Com relação ao UMLS, além de obter-se uma avaliação do vocabulário pertinente a área de Raio-X de Tórax, avaliação esta descrita mais adiante, houve um trabalho de organização de dados para a porção de dados em português. Esta organização gerou uma metodologia que pode ser reproduzida para outras versões de UMLS ou para a ampliação do vocabulário.

Com relação ao Radlex, os resultados obtidos também possuem caráter organizacional, tendo sido adotada uma metodologia de tradução do vocabulário e organização de dados que beneficia aplicações de PLN.

Além de trabalhar a estrutura dos vocabulários, a análise e utilização destes permitiu a formulação de algumas críticas tanto a estrutura como ao conteúdo de ambos os sistemas de Ontologia.

O UMLS por misturar diversos vocabulários acaba por perder uniformidade e dificultar a visualização da hierarquia dos conceitos que ele agrega. Acaba por ser um vocabulário genérico que atende a requisitos como indexar arquivos, por exemplo, mas para uma aplicação mais específica, muitas vezes falha na definição dos conceitos requisitados. A vantagem, é que por ser um agregado de outras ontologias, sempre a possibilidade de ampliação do vocabulário com algum sistema que corresponda a necessidade de uma aplicação particular, porém ainda resta ao desenvolvedor avaliar se vale pena manter um vocabulário tão grande, ou apenas o vocabulário específico para a aplicação trabalhada.

O RadLex, inicialmente era uma boa promessa para suprir as necessidades de uma aplicação de PLN voltada a radiologia, no entanto em sua versão inicial não satisfaz as necessidades da aplicação por estar incompleto, além de faltar uma padronização no formato das definições dos termos. A versão mais recente parece ter resolvido muitos dos problemas, porém possui apenas o vocabulário em inglês, o que requer o trabalho de tradução para o português.

Infelizmente não foi possível trabalhar com a versão mais nova do Radlex para a conclusão desta tese, devido a recente liberação, ficando para trabalhos futuros a adoção desta versão.

6.2 Avaliação do Algoritmo de Busca

O algoritmo de busca por conceitos em uma base de dados de terminologias é baseado na semelhança de duas cadeias de palavras. Para tal avaliação, deve-se considerar não apenas as recuperações relevantes, mas também as respostas negativas, isto é, se o conceito procurado não existe na base de dados, a busca deve retornar uma resposta negativa. As respostas negativas são importantes para que não sejam atribuídos códigos de conceitos erroneamente a cadeias de palavras, considerando o objetivo da busca por conceitos na base de dados de terminologias. Conforme o objetivo da aplicação que utiliza a busca, a precisão pode ser mais rigorosa ou não. Para maiores detalhes sobre o processo de avaliação, consultar o capítulo 3, sobre a metodologia utilizada no trabalho.

Para este trabalho, dada uma cadeia de palavras o objetivo da busca é identificar qual o conceito médico que melhor representa a cadeia dada, isto é, se a cada busca for obtido como resultado pelo menos um conceito e este estiver correto ou se não forem retornados conceitos, caso não haja um conceito adequado, a busca é considerada bem sucedida. Sob este enfoque, a recuperação não é importante, apenas a precisão.

6.2.1 Conjunto de Testes

Foram realizados testes com 4 grupos de conceitos, escolhidos e organizados conforme os métodos já descritos no item 3.2.2 Metodologia de Avaliação do UMLS (página 60).

As tabelas a seguir listam os conceitos utilizados nos testes realizados para avaliação da busca por conceitos. A primeira coluna fornece o código UMLS do conceito sorteado (ou pesquisado manualmente), a segunda coluna a cadeia de palavras a ser pesquisada na base de dados de terminologias e a terceira coluna a quantidade real de ocorrências de cada conceito na base de dados de terminologias. Os conceitos não encontrados na pesquisa manual receberam uma numeração sequencial de identificação em cada experimento para facilitar seu processamento posteriormente.

Doravante, os experimentos serão referenciados como teste 1, teste 2, teste 3 e teste 4.

Tabela 19: Lista de conceitos utilizados nos testes 1 e 2.

CUI	Descrição do conceito	Ocorrência
C1002021	acetobacter xylinum	2
C0013900	ácido benzoárico	2
C0031885	ácidos picolínicos	1
C0154198	afecções hipofisárias iatrogénicas	1
C0001905	Alaska	1
C0019686	antígenos do hiv2	10
C0004811	artéria basilar	1
C0330193	árvore-da-vida	4
C0079086	ascensão na carreira	1
C0004692	Balantidíase	1
C0201942	banda oligoclonal no lcr	2
C0005137	Bertilase	1
C0004925	Besouros	4
C0006542	Bissinose	1
C0006006	Boranos	2
C0006131	Branquioma	5
C0854928	cancro da trompa de falópio estágio iii	2
C0006625	Caquexia	3
C0323529	carrapatos-de-galinha	2
C0058361	catepsina c	3
C0339753	celulite da orelha externa	2
C0040493	compostos de tosil	2
C0425058	cônjuge deficiente	2
C0156979	contração do estreito inferior da pélvis, parto consumado	1
C0017259	conversão gênica	1
C0010305	Cresóis	1
C0011546	dependência (psicologia)	1
C0542064	diminuição das células de kupffer	2

C0008445	displasia epifisária punctata	4
C0001947	dissuasores de álcool	2
C0318332	ehrlichia risticii	2
C0155754	embolismo e trombose de outra artéria especificada	1
C0085362	emprego com apoio especial	4
C0740415	encefalite lúpica	2
C0013500	Equidna	3
C1276801	Erisipelóide	2
C0015008	Etídio	1
C0015533	fator xiii	4
C0031441	Fenoxibenzamina	1
C0042510	fibrilacao ventricular	5
C0016253	Florida	1
C0376571	gene brca1	2
C0026879	Genotoxinas	2
C0029446	glicoproteína sparc	2
C0151672	glicosúria falso-positiva	4
C0085654	granuloma no local de injeção	2
C0153087	hepatite delta sem referência a doença de hepatite b activa sem referência a coma hepático	1
C0020244	hidrocarbonetos alicíclicos	1
C0085341	Hipotríquios	2
C0027615	imunidade neonatal adquirida da mãe	1
C0004048	Inalação	1
C0021313	infecções do rim	5
C0158485	joelho varo (adquirido)	1
C0022983	laminectomia vertebral	2
C0160971	lesão por esmagamento da região do ombro	1
C0160358	lesão traumática do cólon ascendente sem ferida aberta para a cavidade	1
C0235309	mal-estar do estômago	6
C0399523	maloclusão de angle classe i	2
C0155902	manifestações pulmonares agudas de radiações	1
C0004799	membrana basal	1
C0025853	Metoclopramida	1
C0006132	moraxella (branhamella) catarrhalis	3
C0153172	nevrite retrobulbar sífilítica	1
C0029515	outra e ne não absorção pós-cirúrgica	1
C0030305	pancreatite ne	2
C0030596	Partenogênese	1
C0030667	patologia clínica	1
C0325148	peixes-boi	4
C0153226	pericardite gonocócica	1
C0030435	Periquitos	1
C0154610	perturbação explosiva isolada	1
C0154459	perturbação hipomaníaca crónica da personalidade	1
C0031886	Picomaviridae	1
C0033416	Promielócitos	1
C0033439	Propanidida	1
C0949987	proteínas de fermento de padeiro	5
C0073423	proteínas de ligação ao capacete de rna	2
C0042706	proteínas nucleares virais	2
C0033914	psicologia educacional	1
C0161359	queimadura (qquer grau) c/ compromisso 60-69% da superfície corporal c/ queimadura 3º grau em 40-49%	1
C0153052	queratoconjuntivite do sarampo	1
C0073337	quinases de proteína 19935	4
C0009977	radioisótopos de cobre	1
C0151735	reação ne no local de injeção	4
C0034826	receptores muscarínicos de acetilcolina	2
C0185500	redução fechada de articulação com luxação	1
C0198457	reparação de hérnia diafragmática	2
C0021950	repetição terminal invertida	1
C0035475	Rhizobium	1
C0035554	Ribostamicina	1
C0549263	sintomas de abstinência de opiáceos	2
C0037205	Sirenomelia	2
C0339789	surdez hereditária	3
C0919720	tratamento de fractura	2
C0041006	Trihalometanos	1
C0041223	trypanosoma lewisi	1
C0153718	tumor ou linfoma de burkitt com compromisso do baço	1
C0153719	tumor ou linfoma de burkitt com compromisso dos gânglios linfáticos de localizações múltiplas	1
C0042267	Vaginite	3
C0022382	venenos de medusa	1

Tabela 20: Lista de conceitos utilizados no teste 3.

CUI	Descrição do conceito	Ocorrência
C0005388	Bile	1
C0006826	Câncer	9
C0007097	Carcinoma	7
C0007117	carcinoma basocelular	11
C0007137	carcinoma epidermóide	12
C0007133	carcinoma papilífero	1
C0242637	Cardiomioplastia	2
C0086543	Catarata	18
C0015354	Cec	2
C0040124	cisto tireoglosso	8
1	comissurotomias valvares	0
C0036508	dermatite seborréica	21
C0162589	desfibrilador implantável	2
C0018213	doença de graves	12
C0014591	Epistaxe	14
2	espícula óssea	0
C0037995	Esplenectomia	3
3	esquistossomose hepatoesplênica	0
C0193125	Fistulectomia	3
C0017163	Gastroenterologia	1
C0018417	Ginecologia	1
C0018916	Hemangioma	14
C0019116	Hemostasia	1
4	Hemoterapia	0
C0206620	Higroma	8
C0020541	hipertensão portal	6
C0020676	Hipotireoidismo	7
5	implante de marcapasso cardíaco	0
C0035127	lesão por esforços repetitivos	15
C0024204	Linfonodo	2
C0023343	mal de hansen	10
C0494165	metástase hepática	8
C0027712	Nefrologia	1
6	Neonatologia	0
C0029087	Oftalmologia	1
C0086747	Otorrinolaringologia	1
8	pavilhão auricular	0
C0030755	Pediatria	1
C0026267	prolapso de válvula mitral	8
C0013961	Os	3
C0033873	Psiquiatria	1
C0195328	punção-biópsia	1
C0043299	radiografia simples	7
C0034599	Radiologia	1
9	revascularização do miocárdio	0
0	sonda nasointestinal	0
C0023911	transplante hepático	8
C0040590	Traqueostomia	3
C0520997	Trombolises	2
C0190112	Valvoplastia	2

Tabela 21: Lista de conceitos utilizados no teste 4

CUI	Descrição do conceito	Ocorrência
C0014848	acalasia esofágica	7
C0085846	biópsia de aspiração com agulha fina	3
C0006142	câncer de mama	8
C0007137	carcinoma epidermóide metastático	7
C0035439	cardiopatia reumática	4
C0031038	cisto pericárdico congênito	2
C0041234	doença de chagas	5
C0014741	eritema indurado de bazin	11
C0035436	febre reumática	4
C0434205	lesão dos plexos simpáticos	1
C0560672	lesão dos plexos simpáticos **	0
C0007193	miocardiopatia dilatada	4
1	miocardiopatia dilatada induzida por taquicardia ventricular incessante	0
2	nódulos eritemato-violáceos	0
C0032298	pneumonia lipídica	2
C0001175	síndrome da imunodeficiência adquirida	9
3	Taquicardiomiopatia	0
C0587044	trombos na cavidade do VE *	0
C0041328	tuberculose renal	4
C0026265	valvopatia mitral isolada	4
C0225892	via de saída do ventrículo direito *	0

* Conceitos identificados através do Methatesaurus em inglês.

** O conceito "Lesão dos Plexos Simpáticos" aparece na base do UMLS com dois códigos distintos, sendo ambos listados.

Para realizar os teste, foi criado um programa em linguagem *Python*, que lê um arquivo de entrada com os conceitos a serem pesquisados e grava como saída um arquivo no formato CSV com resposta do sistema para cada conceito pesquisado. Cada conceito pesquisado possui como resposta zero ou mais conceitos encontrados, acompanhados de uma pontuação de semelhança. Nos teste 1, 2 e 3 a pontuação de semelhança foi cortada em 0,70, ou seja, o sistema ofereceu como saída apenas os conceitos com pontuação de semelhança maior que 0,70. O teste 4 teve uma pontuação de corte de 0,60, apenas para ilustrar algumas possíveis perdas na identificação de conceitos. O valor de 0,70 foi escolhido empiricamente. Através da observação dos resultados retornados pelo programa de busca notou-se que normalmente itens com pontuação muito abaixo de 0,70 são irrelevantes. Como pode acontecer de haver algum item relevante com pontuação inferior a 0,70, o teste 4 foi executado com pontuação de corte igual a 0,60.

Os resultados obtidos foram importados para um banco de dados, sendo parte dos cálculos feita através de consultas SQL. Os resultados intermediários foram então importados para uma planilha de dados para realização do restante dos cálculos. A variação da pontuação de corte foi simulada através de filtros nas consultas SQL sobre os resultados obtidos ou através de filtros diretamente nas fórmulas da planilha.

A avaliação do algoritmo se deu através de dois métodos distintos. O primeiro método utilizado é baseado em medidas tradicionais usadas em Recuperação de Informação, como a

recuperação e a precisão. O segundo método é baseado em uma curva denominada ROC, muito utilizada em avaliação conjunto de dados em medicina.

As medidas de recuperação e precisão são úteis na avaliação da quantidade e relevância dos itens recuperados por um sistema, mas falham quando é necessário avaliar a ausência de itens relevantes na base pesquisada. A medida de *fall-out* poderia ser usada como complemento, uma vez que é baseada no número de itens não relevantes recuperados, no entanto para o experimento em questão, o *fall-out* sempre é muito baixo pois a proporção de itens irrelevantes recuperados com relação à quantidade de itens não relevantes existentes na base é muito pequena.

A solução encontrada para complementar a avaliação do sistema foi a adoção da curva ROC. A utilização da curva ROC permite a avaliação em termos de itens recuperados e identificados positivamente ou negativamente e como falsos positivos ou negativos. Em complemento a curva é calculada a medida de acurácia, que quantifica o quanto o algoritmo utilizado é exato, isto é, quantifica a capacidade do algoritmo de oferecer respostas verdadeiramente corretas.

6.2.2 Avaliação da Recuperação e da Precisão

O primeiro passo da avaliação foi o cálculo de medidas tradicionais usadas em recuperação de informação, cujos resultados encontram-se nas Tabela 23, Tabela 24, Tabela 25 e Tabela 27. Foram usadas as medidas de recuperação, precisão, *fall-out* e *f-Measure*. A Tabela 22 lista os valores das constantes usadas para os cálculos baseados nas medidas tradicionais de recuperação de informação.

Tabela 22: Constantes para cálculo de avaliação da RI

Variável	Valor
Pontuação de corte mínima do algoritmo para os testes 1, 2 e 3	0,70
Pontuação de corte mínima do algoritmo para o teste 4	0,60
Total de conceitos diferentes na base UML (versão 2005AA) em português	67.409
β	2,00

Em cada teste realizado foram computadas as medidas já citadas para cada conceito pesquisado, variando-se a pontuação de corte no algoritmo de busca, uma vez que o algoritmo usado devolve como resposta uma coleção de *strings*, onde cada *string* possui uma pontuação de semelhança referente ao *string* pesquisada, permitindo que as respostas abaixo da pontuação de corte selecionada sejam eliminadas.

Para cada pontuação de corte foram extraídos o desvio padrão e a média dos valores de cada medida calculada. Considera-se que um conceito foi encontrado quando o código CUI encontrado pelo programa é igual ao código CUI encontrado manualmente ou determinados previamente no sorteio dos conceitos, no caso dos testes 1 e 2.

Para o cálculo do *fall-out*, o número de itens irrelevantes na base (espúrios) foi considerado como sendo o total de itens da base menos o total de itens relevantes existentes para cada conceito. Assim o *fall-out* foi calculado como a divisão entre a quantidade de falsos positivos pela quantidade de itens espúrios. Como o total de itens espúrios para cada busca é muito maior que o total de falsos positivos, a medida de *Fall-out* obtida é próxima de zero.

Os falsos positivos foram considerados como sendo a quantidade total recuperada para cada item menos a quantidade de verdadeiros positivos recuperados. As medidas de recuperação, precisão e *f-measure* foram calculadas através das fórmulas tradicionais.

Tabela 23: Resultados médios para a recuperação, precisão, *fall-out* e *f-measure* para o teste 1.

Corte	Recuperação	σ_{rec}	Precisão	$\sigma_{precisão}$	Fall-out	$\sigma_{fall-out}$	F-measure	$\sigma_{f-measure}$	Falsos	σ_{falsos}
0,70	0,7667	0,3020	0,6479	0,3744	0,0001	0	0,6293	0,2930	5	13
0,75	0,7472	0,3064	0,7655	0,3395	0,0000	0	0,6602	0,2842	2	6
0,80	0,7372	0,3062	0,7825	0,3290	0,0000	0	0,6666	0,2858	2	5
0,85	0,7222	0,3052	0,9212	0,2255	0,0000	0	0,7130	0,2855	1	2
0,90	0,7222	0,3052	0,9399	0,1894	0,0000	0	0,7277	0,2809	0	1
0,95	0,7102	0,3096	0,9725	0,1226	0,0000	0	0,7281	0,2828	0	0
1,00	0,7102	0,3096	0,9775	0,1129	0,0000	0	0,7298	0,2840	0	0

Gráfico 1: Curva para a Recuperação X Precisão para o teste 1

Gráfico 2: Pontuação de corte X Precisão / Recuperação e *F-Measure* para o teste 1

Para o teste 1, a análise dos valores de recuperação e precisão não deixa claro que todos os conceitos obtiveram pelo menos um item recuperado e verdadeiro para qualquer pontuação de corte. Através do Gráfico 2 percebemos que a partir da pontuação de corte de 0,85 os resultados referentes a precisão são aceitáveis, sendo que os valores médios para a precisão estão acima de 0,90 para uma pontuação de corte acima de 0,85. A quantidade média de falsos positivos para o corte em 0,85 é igual a 1. Para corte acima de 0,90, a média de falsos positivos cai para zero.

Ainda no teste 1, este apresentou valores precisão satisfatórios (lembrando que neste trabalho há maior interesse na precisão), no entanto devemos lembrar que os conceitos buscados foram extraídos da própria base pesquisada, o que justifica a obtenção de valores mais altos. Neste caso uma comparação simples de *strings* seria capaz de recuperar os conceitos pesquisados, porém o algoritmo se mostrou capaz de recuperar variações ortográficas de alguns conceitos. A questão da recuperação das variações ortográficas está ligada à recuperação. Teoricamente, a recuperação tende a ser menor quando a precisão aumenta, o que foi verificado, no entanto esta variação foi pequena, mantendo a recuperação média sempre na casa dos 70%. O desempenho sob o enfoque da recuperação é afetado pela existência de conceitos na base definidos através de sinônimos. Este problema é de difícil trato, pois depende da existência de dicionários de sinônimos para substituição de termos durante a busca.

A forma como os conceitos são definidos na base de dados de terminologias apresenta uma certa heterogeneidade, referentes à utilização de sinônimos, siglas e diferentes origens dos termos definidos (lembrando que UMLS é um agrupamento de diversos sistemas de vocabulário médicos). Os termos que possuem poucas definições tendem a aumentar a recuperação média, enquanto que os termos com muitas definições tendem a abaixar o valor da recuperação média, pois geralmente possuem sinônimos e siglas. A variação grande da recuperação para cada item aumenta o valor do desvio padrão, uma vez que a distribuição de frequência dos termos é bem heterogênea e o algoritmo usado é capaz de recuperar apenas alguns tipos de variações ortográficas, como flexões de gênero, número e grau, por exemplo.

Considerando os objetivos deste trabalho, que pretende utilizar o algoritmo de comparação de cadeias de palavras e de busca para encontrar qual o conceito médico ao qual uma cadeia de palavras se refere, o teste 1 mostrou-se satisfatório, devido a obtenção de um valor alto de precisão.

Na realidade uma avaliação booleana seria o suficiente para provar que o programa de busca de conceitos satisfaz os requisitos deste trabalho. Para cada conceito recuperado da base de dados de terminologias basta saber se ele é um positivo verdadeiro, um positivo falso, um negativo verdadeiro ou um negativo falso. Para este tipo de análise a curva ROC se apresenta mais adequada. Mais adiante será apresentada a análise através da curva ROC.

Tabela 24: Resultados médios para a recuperação, precisão, *fall-out* e *f-measure* para o teste 2.

Corte	Recuperação	Precisão	Fall-out	F-measure	Falsos	Desv. Padrão falsos
0,70	0,0000	0,0000	0	0,0000	5	13
0,75	0,0000	0,0000	0	0,0000	2	6
0,80	0,0000	0,0000	0	0,0000	2	5
0,85	0,0000	0,0000	0	0,0000	1	2
0,90	0,0000	0,0000	0	0,0000	0	1
0,95	0,0000	0,0000	0	0,0000	0	0
1,00	0,0000	0,0000	0	0,0000	0	0

Para o teste 2, os conceitos selecionados no primeiro teste, foram excluídos da base de dados de terminologias, sendo assim o programa não deve recuperar nada relevante. Para o uso do algoritmo mais adiante neste trabalho, é importante o fato de que uma resposta vazia do sistema significa que nada foi encontrado realmente. A análise do teste 2 por meio das medidas tradicionais de avaliação de Recuperação de Informação não oferece recursos suficientes para avaliar a ausência de itens relevantes em uma base de dados.

Como já foi comentado anteriormente, o *fall-out* também não oferece uma medida boa neste caso para validar os objetivos do teste. Resta fazer a análise sobre a quantidade de itens

recuperados e de falsos positivos, no entanto esta análise será deixada para mais adiante pois é feita através da curva ROC.

Tabela 25: Resultados médios para a recuperação, precisão, *fall-out* e *f-measure* para o teste 3.

Corte	Recuperação	σ_{rec}	Precisão	$\sigma_{precisão}$	Fall-out	$\sigma_{fall-out}$	F-measure	$\sigma_{f-measure}$	Falsos	σ_{falsos}
0,70	0,3499	0,3991	0,3447	0,3981	0,0001	0	0,2683	0,3072	0	0
0,75	0,3422	0,4002	0,3590	0,4086	0,0001	0	0,2623	0,3074	8	24
0,80	0,3422	0,4002	0,3590	0,4086	0,0001	0	0,2623	0,3074	7	24
0,85	0,3193	0,4002	0,6414	0,4086	0,0000	0	0,3336	0,3074	7	24
0,90	0,3193	0,3920	0,6414	0,4739	0,0000	0	0,3336	0,3906	0	24
0,95	0,3139	0,3945	0,6500	0,4765	0,0000	0	0,3275	0,3933	0	1
1,00	0,3139	0,3945	0,6500	0,4765	0,0000	0	0,3275	0,3933	0	0

Gráfico 3: Curva para a Recuperação X Precisão para o teste 3.

Gráfico 4: Pontuação de corte X Precisão / Recuperação e *F-Measure* para o teste 3.

O teste 3 foi realizado para validar a busca através do teste de itens, que em um primeiro momento, não é conhecida sua existência na base e que foram escritos de uma forma que não é necessariamente igual às definições contidas no UMLS.

No caso do teste 3, em um total de 50 itens testados, foram recuperados 33 conceitos corretos independentemente da pontuação de corte. No caso de uma pontuação de corte mais elevada, a precisão é favorecida. Dos 17 itens não identificados, 9 podem realmente não existir na base de dados de terminologias, não tendo sido identificados manualmente e constituindo os verdadeiros negativos. Os 8 itens restantes tratam-se de falsos negativos. Considerando que dos 50 itens testados 8 itens não foram recuperados, teríamos um erro de cerca de 16%.

O que se observou inicialmente a respeito dos itens não recuperados, foi a existência de erros de ortografia na base de dados de terminologias, erros com relação ao *Snowball Stemmer* e o não tratamento de sinônimos e abreviações.

Tabela 26: Conceitos não recuperados no teste 3.

CUI	Conceito	Item procurado	Motivo da perda
C0013961	pronto-socorro	ps	Abreviação
C0007133	carcinoma papilar	carcinoma papilífero	<i>Stemmer</i>
C0040124	cisto tiroglossal	cisto tireoglosso	Ortografia
C0023343	Hanseníase	mal de hansen	Sinônimo
C0043299	Radiografia	radiografia simples	Pontuação
C0162589	desfibriladores implantáveis	desfibrilador implantavel	<i>Stemmer</i>
C0190112	valvuloplastia cardíaca	valvoplastia	Ortografia
C0193125	excisão de fistula anal	fistulectomia	Sinônimo

A Tabela 26 mostra as prováveis causas da não recuperação de itens existentes na base de dados de terminologias através do algoritmo testado.

Como já discutido anteriormente, o *Snowball Stemmer* é um algoritmo baseado em regras, portanto é suscetível a erros na obtenção das raízes das palavras. Palavras terminadas em “l”, “ão”, “ões” são processadas incorretamente gerando raízes inadequadas. Os erros no algoritmo do *Snowball Stemmer* afetam principalmente a comparação entre um conceito definido no plural com um conceito escrito no singular. Esses erros podem ser corrigidos através da revisão de algumas regras do algoritmo usado pelo *Snowball Stemmer*, mas é importante ter em mente que o algoritmo do *Snowball Stemmer* sempre apresentará uma taxa de erros superior aos algoritmos de *stemming* baseados em dicionários. No caso do conceito C0007133, as palavras “papilar” e “papilífero” não possuem o mesmo radical, por isso a pontuação de semelhança ficou abaixo do corte estabelecido para os teste.

Como o algoritmo de busca por conceitos não faz correção ortográfica, ele não é capaz de identificar conceitos que estejam definidos na base com diferenças ortográficas em relação ao item pesquisado. Por exemplo, tireoglossa teria a mesma raiz que tireoglossal, mas não a mesma raiz que tiroglossal. Uma possibilidade para melhorar o desempenho neste caso é a realização de busca fonética.

Por fim, o item C0043299 teve uma pontuação de semelhança menor que o corte utilizado no teste 3, por isso não foi recuperado. As soluções possíveis seriam: alterar os pesos referentes a comprimento e interseção de palavras no momento da comparação entre *strings* ou adicionar um novo fator de pontuação, baseado na classe gramatical das palavras. Por exemplo, assumindo que o portador do sentido principal em uma frase é sempre um substantivo, poderíamos atribuir acréscimos de pontuação para itens comparados que possuem substantivos idênticos. A atribuição de pontos referentes a classe gramatical das palavras depende da rotulação prévia da base de conceitos e do item procurado na base, através do uso de um *POS Tagger*.

Não considerando para os cálculos dos valores médios os itens que não foram identificados manualmente na base de dados de terminologias, temos os novos valores médios mostrados na Tabela 27.

Tabela 27: Resultados médios da recuperação, precisão, *fall-out* e *f-measure* para o teste 3, considerando apenas os itens encontrados manualmente.

Corte	Recuperação	σ_{rec}	Precisão	$\sigma_{precisão}$	Fall-out	$\sigma_{fall-out}$	F-measure	$\sigma_{f-measure}$	Falsos	σ_{falsos}
0,70	0,4267	0,4020	0,4204	0,4019	0,0001	0,0004	0,3271	0,3096	9	26
0,75	0,4173	0,4051	0,4378	0,4113	0,0001	0,0004	0,3199	0,3112	9	26
0,80	0,4173	0,4051	0,4378	0,4113	0,0001	0,0004	0,3199	0,3112	9	26
0,85	0,3894	0,4003	0,7822	0,4028	0,0000	0,0000	0,4069	0,3954	0	1
0,90	0,3894	0,4003	0,7822	0,4028	0,0000	0,0000	0,4069	0,3954	0	1
0,95	0,3828	0,4045	0,7927	0,4027	0,0000	0,0000	0,3993	0,4000	0	0
1,00	0,3828	0,4045	0,7927	0,4027	0,0000	0,0000	0,3993	0,4000	0	0

Mais uma vez a avaliação somente pela recuperação e precisão não deixa claro qual a quantidade de itens recuperados corretamente. A precisão obtida para a pontuação de corte acima de 0,85 é razoável, porém a precisão pode ser ainda melhorada com as correções descritas anteriormente.

Tabela 28: Resultados médios para a recuperação, precisão, *fall-out* e *f-measure* para o teste 4.

Corte	Recuperação	σ_{rec}	Precisão	$\sigma_{precisão}$	Fall-out	$\sigma_{fall-out}$	F-measure	$\sigma_{f-measure}$	Falsos	σ_{falsos}
0,60	0,3881	0,3442	0,1876	0,3231	0,0008	0,0016	0,2103	0,2546	55	107
0,65	0,2801	0,3175	0,3381	0,3839	0,0000	0,0001	0,2541	0,2709	3	5
0,70	0,1879	0,2652	0,2779	0,3625	0,0000	0,0000	0,1879	0,2600	2	3
0,75	0,1778	0,2606	0,2847	0,3723	0,0000	0,0000	0,1823	0,2599	1	2
0,80	0,1611	0,2133	0,2847	0,3723	0,0000	0,0000	0,1680	0,2187	1	2
0,85	0,1444	0,1835	0,3583	0,4329	0,0000	0,0000	0,1583	0,1953	0	1
0,90	0,1444	0,1835	0,3619	0,4345	0,0000	0,0000	0,1588	0,1962	0	1
0,95	0,1139	0,1346	0,4500	0,5071	0,0000	0,0000	0,1334	0,1571	0	0
1,00	0,1139	0,1346	0,4500	0,5071	0,0000	0,0000	0,1334	0,1571	0	0

Gráfico 5: Curva para a Recuperação X Precisão para o teste 4.

Gráfico 6: Pontuação de corte X Precisão / Recuperação e *F-Measure* para o teste 4.

O teste 4 foi idealizado para verificar o comportamento da busca para identificação de conceitos a partir de itens extraídos de textos reais da área médica. Trata-se de um experimento um pouco menor que os demais e que apresentou resultados piores no que se refere à recuperação e a precisão.

Dos 20 conceitos pesquisados, 3 não foram identificados manualmente e 2 conceitos foram identificados apenas em inglês, portanto assumiu-se que não existem na base de dados de terminologias em português. Para uma pontuação de corte de 0,60, 14 itens foram encontrados restando um item não recuperado. Considerando que de um total de 20 itens, 5 não existem na base e um não foi recuperado, teríamos um erro de 5% para este teste.

Para uma pontuação de corte de 0,80, dos 20 conceitos pesquisados, 9 conceitos foram encontrados restando 6 conceitos que não foram recuperados. Considerando que de um total de 20 itens, 6 não foram recuperados, teríamos um erro de 30% para este teste.

Tabela 29: Conceitos não recuperados no teste 4, para pontuação de corte igual a 0,80..

CUI	Conceito	item procurado	Motivo da perda
C0032298	peumonia lipóide (ou peumonia lipídica)	peumonia lipoldica	<i>Stemmer</i>
C0007137	carcinoma epidermóide	carcinoma epidermóide metastático	Pontuação
C0014741	eritema endureado de bazin	eritema indurado de bazin	Ortografia
C0031038	cisto pericárdico	cisto pericárdico congênito	Pontuação
C0434205	lesão traumática dos nervos simpáticos	lesão dos plexos simpáticos	Pontuação
C0026265	valvulopatia mitral	valvopatia mitral isolada	Ortografia

No caso do teste 4, para uma pontuação de corte maior que 0,80, a maioria dos itens não recuperados estavam abaixo da pontuação estipulada. Para solucionar este problema ou deve ser feita uma alteração na atribuição de pontuação, como já sugerido, ou então abaixar a pontuação de corte para o teste. A desvantagem de abaixar a pontuação de corte é um aumento muito grande da recuperação e consequentemente perda de precisão. Em termos práticos, se um item não é encontrado, há a possibilidade de abaixar a pontuação de corte, porém a aplicação que utiliza os dados deve prevenir o usuário da perda de precisão, ou manter algum tipo de anotação a respeito da precisão ou da pontuação obtida para os itens recuperados.

Apesar dos conceitos C0434205 e C0560672 (lesão dos plexos simpáticos) expressarem o mesmo significado, foram mantidos no experimento como entradas separadas, para o caso de ambos serem encontrados pelo algoritmo de busca por conceitos, porém o fato de não ter sido recuperado nada com o código 'C0560672' não foi computado como item não recuperado. Na verdade verificou-se manualmente a inexistência da definição do código C0560672 em português.

6.2.3 Curva ROC

A curva ROC (*Receiver Operating Characteristic*) é muito usada em medicina para medir a exatidão de diagnósticos. Neste trabalho a curva ROC é utilizada para avaliar a resposta do algoritmo de busca por conceitos em uma base de dados de terminologias, de forma a avaliar se as respostas dadas pelo sistema são relevantes ou não. Optou-se por utilizar a curva ROC, mais um conjunto de parâmetros associados a ela, em adição as medidas de recuperação e precisão, pois estas não fornecem uma forma de avaliar a inexistência de itens relevantes na base pesquisa.

Como já explicado, a variação dos pontos de corte para a montagem da curva foi obtida no momento dos cálculos, isto é, na própria fórmula que realiza a contagem de positivos e negativos, uma vez que os dados foram importados para uma planilha de dados.

A área estimada sob a curva ROC foi calculada através da soma das áreas dos trapézios formados entre cada intervalo de pontuação de corte no gráfico obtido.

A Tabela 30 descreve as legendas para os itens calculados para a avaliação através da curva ROC. As tabelas a partir da Tabela 31 até a Tabela 33 e os gráficos a partir do Gráfico 7 até o Gráfico 14 se referem aos testes 1, 2, 3 e 4 sob o enfoque da curva ROC.

Tabela 30: Legendas (avaliação através da curva ROC)

VP	Verdadeiro positivo
FP	Falso positivo
VN	Verdadeiro negativo
FN	Falso negativo
FVP	Fração de verdadeiro positivos
FVN	Fração de verdadeiro negativos
FFP	Fração de falso positivos
FFN	Fração de falso negativos
Acurácia	Acurácia
P(A)	Fração da população estudada que são realmente positivos
P(N)	Fração da população estudada que são realmente negativos
Gold	Valores de referência ou <i>Gold standard</i>
PPV	Previsão de positivos (Positive Predictive Value)
NPV	Previsão de negativos (Negative Predictive Value)
P(A T⁺)	Regra de Bayes (Bayes Rule)
Área estimada	Área estimada sob a curva ROC.

Em um teste onde tivéssemos o máximo desempenho, a área sob a curva ROC teria o valor um. Podemos observar que dos quatro experimentos realizados, os testes 1 e 3 foram os que obtiveram melhores resultados fornecendo uma área próxima a um. O teste 2, por não apresentar valores de verdadeiros positivos, tem sua análise baseada em outras medidas

associadas à curva ROC, como a medida de acurácia, da precisão dos negativos e da contagem de negativos verdadeiros e falsos. Apenas como método auxiliar, foi montado um gráfico de FVN x FFP para o teste 2. O teste 4 foi o que apresentou pior desempenho, assim como na análise por recuperação e precisão, porém agora será possível visualizar a quantidade de conceitos recuperados corretamente para cada item pesquisado.

Tabela 31: Dados para curva ROC, Teste 1

Corte	1,00	0,95	0,90	0,85	0,80	0,75	0,70
Identificados	100	100	100	100	100	100	100
Não identificados	0	0	0	0	0	0	0
Inexistentes	0	0	0	0	0	0	0
VP	105	105	109	109	112	115	120
FP	6	7	30	58	181	217	482
VN	476	475	452	424	301	265	0
FN	15	15	11	11	8	5	0
FVP	0,8750	0,8750	0,9083	0,9083	0,9333	0,9583	1,0000
FDN	0,9876	0,9855	0,9378	0,8797	0,6245	0,0104	0,0000
FFN	0,1250	0,1250	0,0917	0,0917	0,0667	0,0417	0,0000
FFP	0,0124	0,0145	0,0622	0,1203	0,3755	0,9896	1,0000
PPV	94,5946	93,7500	78,4173	65,2695	38,2253	34,6386	19,9336
NPV	96,9450	96,9388	97,6242	97,4713	97,4110	98,1481	0,0000
P(A T ⁺)	0,9459	0,9375	0,7842	0,6527	0,3823	0,1943	0,1993
Acurácia	0,9651	0,9635	0,9319	0,8854	0,6860	0,1993	0,1993
Gold	Total de positivos:		120				
	Total de negativos:		482				
	Total de itens achados:		602				
P(A):	0,1993						
P(N):	0,8007						
Área estimada	0,9267

Gráfico 7: Curva ROC para o Teste 1

Gráfico 8: Acurácia X Corte para o Teste 1

Para o teste 1, observamos que a diferença entre a área ideal sob a curva ROC e a área obtida é de 7%, o que indica que os resultados obtidos estão razoavelmente próximos do ideal. Apesar da existência de alguns falsos positivos e falsos negativos, todos os conceitos pesquisados foram recuperados, como o previsto para o teste, pois os conceitos realmente existiam na base de dados de terminologias. Os falsos positivos estão ligados a expressões que contêm os mesmos radicais de palavras, em decorrência dos erros de *stemming*, como por exemplo, “florida” e “flor”. Os falsos negativos estão ligados a fatores já descritos como sinônimos, abreviações ou problemas com o algoritmo do *Snowball Stemmer*.

No teste 1 para uma pontuação de corte acima de 0,85 o PPV (Previsão de positivos), que indica a eficiência do programa para acertar os valores positivos, é de aproximadamente 78,40% e o NPV (Previsão de negativos), que indica a eficiência do programa para acertar os negativos é de aproximadamente 97,47%. A acurácia obtida, medida que resume as quantidades de positivos e negativos, foi de 93,19%. O valor de $P(A | T^+)$, que indica a probabilidade de um item classificado como verdadeiro positivo ser realmente o conceito procurado foi de 78,42%. Estes valores indicam que uma pontuação de corte maior ou igual a 0,85 fornece bons resultados para o teste 1, o que fica mais fácil de visualizar através do Gráfico 8.

A única ressalva a ser feita é que os falsos positivos, embora tenham uma escala pequena, precisam ser identificados com relação aos verdadeiros positivos de alguma forma,

como por exemplo sempre adotar a maior pontuação, ou então discriminar como saída o erro médio do programa.

Tabela 32: Dados para curva ROC, Teste 2

Corte	1,00	0,95	0,90	0,85	0,80	0,75	0,70
Identificados	0	0	0	0	0	0	0
Não identificados	100	100	100	100	100	100	100
Inexistentes	100	100	100	100	100	100	100
VP	0	0	0	0	0	0	0
FP	6	7	30	58	181	217	479
VN	521	520	497	469	346	310	48
FN	0	0	0	0	0	0	0
FVP	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
FVN	0,9886	0,9867	0,9431	0,8899	0,6565	0,0000	0,0000
FFN	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
FFP	0,0114	0,0133	0,0569	0,1101	0,3435	1,0000	1,0000
PPV	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
NPV	100,0000	100,0000	100,0000	100,0000	100,0000	100,0000	100,0000
P(A T ⁺)	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Acurácia	0,9867	0,9848	0,9413	0,8883	0,6553	0,0000	0,0000
Gold	Total de positivos:						0
	Total de negativos:						527
	Total de itens achados:						527
P(A):	0						
P(N):	0,9981						
Área estimada	0

Gráfico 9: FVN x FFN para o Teste 2

Gráfico 10: Acurácia X Corte para o Teste 2

O teste 2, serve para avaliar a questão dos itens inexistentes na base de dados de terminologias, portanto ele é composto apenas de negativos. O Gráfico 10, nos mostra que uma pontuação de corte menor que 0,80 não é recomendável, pois apresenta uma acurácia próxima de zero. Os valores de acurácia aceitáveis estão na faixa de pontuação de corte maiores ou iguais a 0,85.

A FVN é uma boa forma de visualizar o sucesso do sistema em reconhecer que os itens pesquisados não foram encontrados, para pontuações de corte maiores ou iguais a 0,85.

Tabela 33: Dados para curva ROC, Teste 3

Corte	1,00	0,95	0,90	0,85	0,80	0,75	0,70
Identificados	33	33	33	33	33	33	33
Não identificados	8	8	8	8	8	8	8
Inexistentes	9	9	9	9	9	9	9
VP	46	46	50	50	57	57	61
FP	1	1	8	8	387	387	397
VN	411	411	404	404	25	25	15
FN	15	15	11	11	4	4	0
FVP	0,7541	0,7541	0,8197	0,8197	0,9344	0,9344	1,0000
FVN	0,9976	0,9976	0,9806	0,9806	0,0607	0,0097	0,0000
FFN	0,2459	0,2459	0,1803	0,1803	0,0656	0,0656	0,0000
FFP	0,0024	0,0024	0,0194	0,0194	0,9393	0,9903	1,0000
PPV	97,8723	97,8723	86,2069	86,2069	12,8378	12,8378	13,3188
NPV	96,4789	96,4789	97,3494	97,3494	86,2069	86,2069	100,0000
$P(A T^+)$	0,9787	0,9787	0,8621	0,8621	0,1284	0,1226	0,1290
Acurácia	0,9662	0,9662	0,9598	0,9598	0,1734	0,1290	0,1290
Gold	Total de positivos:						61
	Total de negativos:						412
	Total de itens achados:						473
P(A):	0,1290						
P(N):	0,8710						
Área estimada	0,8781

Gráfico 11: Curva ROC para o Teste 3

Gráfico 12: Acurácia X Corte para o Teste 3

No teste 3, mais uma vez a pontuação de corte de 0,85 se mostrou como um divisor para a relevância dos resultados para os conceitos pesquisados. Verifica-se que dos 50 itens pesquisados, 33 conceitos foram recuperados corretamente e 9 não deveriam existir realmente na base de conceitos, restando 8 itens não identificados. Percebemos que neste teste a diferença entre a área de uma curva ideal para a curva achada aumentou para aproximadamente 12%. A acurácia, o PPV e o NPV indicam que o programa teve uma boa

eficiência para identificar positivos e negativos. O $P(A | T^+)$ também é satisfatório indicado a capacidade do programa reconhecer os verdadeiros positivos.

Exemplo 24: Dados para curva ROC, Teste 4

Corte	1,00	0,95	0,90	0,85	0,80	0,75	0,70	0,65	0,60
Identificados	9	9	9	9	9	9	10	13	14
Não identificados	6	6	6	6	6	6	5	2	1
Inexistentes	5	5	5	5	5	5	5	5	5
VP	18	18	21	21	22	23	25	31	42
FP	0	1	9	10	19	19	45	65	1179
VN	1184	1183	1175	1174	1165	1165	1139	1119	5
FN	24	24	21	21	20	19	17	11	0
FVP	0,4286	0,4286	0,5000	0,5000	0,5238	0,5476	0,5952	0,7381	1,0000
FVN	1,0000	0,9992	0,9924	0,9916	0,9840	0,0160	0,0144	0,0093	0,0000
FFN	0,5714	0,5714	0,5000	0,5000	0,4762	0,4524	0,4048	0,2619	0,0000
FFP	100,00	94,7368	70,0000	67,7419	53,6585	54,7619	35,7143	32,2917	3,4398
PPV	98,0132	98,0116	98,2441	98,2427	98,3122	98,3953	98,5294	99,0265	100,00
NPV	1,0000	0,9474	0,7000	0,6774	0,5366	0,0194	0,0210	0,0257	0,0343
$P(A T^+)$	1,0000	0,9474	0,7000	0,6774	0,5366	0,0194	0,0210	1,0000	0,0000
Acurácia	0,9804	0,9796	0,9755	0,9747	0,9682	0,0343	0,0343	0,0253	0,0343
Gold	Total de positivos: 42								
	Total de negativos: 1184								
	Total de itens achados: 1226								
P(A):	0,0343								
P(N):	0,9657								
Área estimada	0,5384

Gráfico 13: Curva ROC para o Teste 4

Gráfico 14: Acurácia X Corte para o Teste 4

No teste 4, a diferença entre as áreas ideal e obtida subiu para aproximadamente 46%, que já é uma diferença bastante significativa, porém para um corte acima de 0,90, o programa ainda apresenta acurácia, PPV e o NPV bons. Este teste em particular deve ser analisado mais cuidadosamente, por ser de menor porte e envolver a seleção manual de conceitos para teste. Notamos aqui que uma redução da pontuação de corte possibilita a recuperação de praticamente todos os itens pesquisados, mas há um preço a se pagar que é o aumento da recuperação e queda da precisão. É importante ressaltar que a maioria dos conceitos identificados como positivos, são verdadeiramente positivos, o que explica os valores de PPV e $P(A | T^+)$ mais elevado para o corte superior a 0,95. Considerando os valores obtidos, os resultados do teste 4 podem ser considerados razoáveis.

6.2.4 Considerações Gerais Sobre a Avaliação

Para o cálculo do *fall-out* foram considerados espúrios todos os *strings* contidos na base que representam os conceitos que não são de interesse para cada busca. Como a base contém 67409 conceitos diferentes e o número máximo de repetições de um mesmo conceito na base é 39, para todas as medidas tomadas o número de conceitos espúrios é próximo do total de conceitos diferentes existentes na base e de uma grandeza muito maior que a quantidade de conceitos recuperados em qualquer uma das buscas testadas, o que explica por que os valores obtidos para o *Fall-out* foram tão baixos.

Considerando a diferença de grandeza entre os conceitos irrelevantes recuperados e a quantidade de conceitos irrelevantes na base de dados de terminologias em cada busca, percebemos que o algoritmo se comporta bem com relação à recuperação de informação irrelevante, isto é, recupera uma fração bastante pequena de informação irrelevante.

Os valores da precisão e do *f-measure* poderiam nos levar a crer que o algoritmo testado tem um desempenho médio, porém a análise através da curva ROC e as medidas associadas a ela, mostram que para a maioria das buscas, o conceito procurado foi recuperado com sucesso, considerando que para o propósito do algoritmo basta encontrar um único conceito que satisfaça a condição de pontuação o mais próximo de um.

Quando a pontuação encontrada é igual a zero, significa que certamente a cadeia de palavras não existe na base de dados de terminologias, porém isso não significa que o conceito não exista, pois o algoritmo de busca não considera fatores como sinônimos, siglas ou mesmo a necessidade de algumas correções no *Snowball Stemmer*, como já explicado anteriormente.

Em geral, o algoritmo satisfaz o requisito para os qual ele foi projetado, dado uma cadeia de palavras, encontrar o código do conceito médico que a cadeia de palavras representa.

6.3 Avaliação do Vocabulário UMLS

O vocabulário médico contido no *Metathesaurus* do UMLS é bastante vasto, no entanto através de inspeção inicialmente manual, percebeu-se que alguns conceitos pertencentes aos laudos de raio-x de tórax não estavam presentes neste vocabulário. Com a desconfiança de que o vocabulário não seria adequado para o processamento dos laudos, foi elaborado um teste para verificar tal adequação.

O teste consiste em selecionar um conjunto de conceitos a partir dos laudos de raio-x de tórax e fazer a busca por estes conceitos com o algoritmo de busca descrito neste trabalho.

Foram selecionados manualmente para teste 174 conceitos diferentes a partir do banco de dados de frases que constituem os laudos, para então serem executadas três buscas com pontuações de corte diferentes. O corte mínimo escolhido foi de 0,85, pois se verificou na fase

de teste que este valor apresentou uma precisão mínima razoável. Além do corte em 0,85, foram rodados testes com corte igual a 0,90 e 0,95.

A Tabela 34 mostra um resumo dos resultados obtidos com os teste para avaliação do uso no UMLS neste trabalho.

Tabela 34: Resultado dos testes executados para avaliação do UMLS

Teste	Corte	Total recuperado	Itens identificados	Porcentagem do total
1	0,85	90	36	20%
2	0,90	90	36	20%
3	0,95	34	27	15%

Para a pontuação de corte de 0,85 e de 0,90, os resultado obtidos foram idênticos, sendo que doravante a avaliação será feita com os dados obtidos com a pontuação de corte igual a 0,85.

Através de inspeção manual, observou-se que para o corte de 0,95, os itens encontrados eram os mesmo que os procurados, isto é, a taxa de acerto para os itens encontrados foi de 100%. Para as pontuações de 0,90 e 0,85, 53 itens recuperados não eram exatamente o conceito procurado havendo apenas um item recuperado corretamente para a pontuação de 0,90, porém para o mesmo item também houve um item recuperado com pontuação maior que 0,95.

Considerando o teste 3, de 174 itens foram recuperados apenas 27, cerca de 15% dos itens procurados. Mesmo sabendo que no pior caso do algoritmo de busca, exposto no item 6.2, houve uma taxa de 30% de itens não recuperados em um experimento pequeno. Apesar de no pior caso a taxa de itens não recuperados erroneamente ser um pouco mais elevada, os valores obtidos para avaliação do UMLS são um indicativo de que a base do *Metathesaurus* não é a melhor escolha para servir como léxico para algoritmos de PLN na área de raio-x de tórax, devido a taxa baixa de recuperação obtida neste teste.

Os resultados obtidos com os testes realizados, nos levam a crer que o vocabulário contido no UMLS não é satisfatório para o objetivo deste trabalho. A alternativa talvez mais precisa de avaliação seria a busca manual dos itens selecionados para teste. A avaliação feita manualmente possui como principais desvantagens a quantidade de pessoas necessárias para a execução do trabalho, o tempo maior despendido e a necessidade de profissionais especializados na área testada. Vale lembrar que mesmo a avaliação humana possui uma taxa de erros relacionada a itens não recuperados.

Apesar de algumas imprecisões, a avaliação automática ainda é capaz de fornecer indícios do que é buscado, além de ser reproduzível caso seja necessário avaliar outros vocabulários médicos. Esses indícios são importantes para um melhor planejamento do trabalho como um todo.

Com a avaliação dos resultados obtidos neste teste, optou-se inicialmente por trocar o vocabulário utilizado para o RadLex, um vocabulário específico para radiologia. Em seu estado implementação inicial, já possuía a definição de parte dos termos referentes a modalidade de Raios-X de tórax, porém precisou ser traduzido para o português.

Foi prevista a necessidade de alguma ampliação do vocabulário do Radlex, para este trabalho e apesar de esta ampliação não ser padrão, visa aumentar o desempenho do reconhecimento de conceitos mais tarde. A preferência pelo *RadLex* se dá não apenas pelo fato de possuir as definições referentes a raio-x de tórax, mas também por ser uma ontologia estruturada especificamente para radiologia, cuja estrutura semântica promete oferecer mais benefícios para este trabalho. O ideal seria utilizar a versão mais recente do Radlex, no entanto ainda é necessário um trabalho de tradução para o português.

6.4 Avaliação do *Brill Tagger*

6.4.1 Anotação Sintática dos Laudos para Treinamento

Para o treinamento do *Brill Tagger* em língua portuguesa, parte dos laudos foi separada para constituir o *Corpus* de treinamento do projeto. O *Corpus* de treinamento foi anotado manualmente, sendo que os rótulos usados são um subconjunto dos definidos para o *Brown Corpus*, no entanto foi necessário fazer uma pequena adaptação. Como não há rótulo no *Brown Corpus* para adjetivos no plural, foi usado um rótulo já existente, que não tinha função para a língua portuguesa para representar os adjetivos no plural. A saber, foi usado o rótulo JJS para adjetivos no plural. A lista de rótulos utilizados para anotação do *Corpus* de treinamento encontra-se no Anexo 11.4.

Optou-se por usar os rótulos do *Brown Corpus*, ainda que estes não cobrissem todas as necessidades da língua portuguesa, para que o trabalho se mantenha o mais próximo possível dos padrões utilizados na área de PLN.

O Exemplo 25 mostra uma frase extraída de um laudo que foi anotado segundo os rótulos descritos anteriormente. Foram marcados os sinais de pontuação, pois muitos rotuladores os consideram como uma palavra a ser contada no modelo de n-grama, além de que a informação pode ser utilizada para auxiliar na delimitação de sentenças.

Exemplo 25: Exemplo de frase extraída de um laudo anotado.

abaulamento/NN do/IN arco/NN médio/JJ./.

As frases anotadas pertencem a uma primeira remessa de laudos de raio-x de tórax extraídas da base hospitalar. Esta primeira remessa possui 1132 laudos com conteúdo. Cada laudo possui 10 frases em média, sendo que o menor laudo possui apenas uma frase e o maior possui 15 frases.

Os 1132 laudos são constituídos por 907 frases de grafia diferente. A repetição da mesma grafia entre frases em laudos diferentes se dá devido a utilização de um dicionário comum entre os radiologistas da instituição. Das 907 frases diferentes, foram anotadas manualmente 508.

A avaliação do *Brill Tagger* foi realizada em dois passos:

O primeiro passo envolveu a submissão do *Corpus* de treinamento ao procedimento de treinamento do *Brill Tagger*. Após treinar o algoritmo, as frases de treinamento foram limpas, isto é, os rótulos sintáticos foram removidos, voltando o texto para seu estado original. O texto limpo foi então submetido ao *Brill Tagger*, desta vez para ser rotulado automaticamente. Os arquivos foram então comparados através de um programa escrito especificamente para a avaliação do rotulador, sendo que os resultados obtidos podem ser observados na Tabela 35.

Tabela 35: Resultados do primeiro passo de avaliação do *Brill Tagger*

Total de sentenças no <i>Corpus</i> de treinamento:	508
Quantidade média de tokens/sentença:	7,45
Total de tokens no <i>Corpus</i> de treinamento:	3786
Total de tokens incorretos:	2
Erro	0,0528%

Como era de se esperar, há uma taxa de erros ainda que baixa, pois o algoritmo usa regras e alguma estatística para o treinamento, o que gera um resultado que não é perfeito, mas bastante aceitável.

O segundo passo consistiu em submeter o *Corpus* de teste ao *Brill Tagger* e corrigir manualmente os erros, gerando um padrão de comparação. O arquivo gerado no segundo

passo, foi corrigido pela mesma pessoa que rotulou o *Corpus* de treinamento. Os resultados obtidos no segundo passo podem ser observados na Tabela 36.

Tabela 36: Resultados do primeiro passo de avaliação do *Brill Tagger*

	Padrão	Teste (original)	Teste (corrigido)
Total de sentenças:	864	864	864
Quantidade média de tokens/sentença:	8,0833	8,1516	8,0833
Total de tokens:	6984	7043	6984
Erros de digitação (palavra quebrada com espaço):	0	59	0
Total de tokens incorretos:	0	473	552
Total de tokens incorretos corrigido:	0	631	-
Erro:	-	8,9593%	7,9038%

Algumas das sentenças do *Corpus* de teste, possuíam palavras quebradas por espaços em branco, não sendo possível determinar se essas palavras estavam “quebradas” já no texto original do laudo, ou se algum passo da aquisição dos dados gerou os erros no texto. Para gerar o arquivo padrão para a avaliação, essas palavras foram corrigidas, porém esta correção gerou uma dificuldade na avaliação. As sentenças corrigidas do arquivo padrão não possuíam mais o mesmo número de *tokens* que o arquivo do *Corpus* de teste, o que dificulta o alinhamento das palavras nas sentenças padrão e de teste, para avaliação.

Para resolver este problemas, adotou-se o seguinte procedimento. Como o número de palavras quebradas não é grande, elas foram corrigidas no arquivo gerado pelo *Brill Tagger*, tendo sido atribuído o rótulo correto a palavra corrigida. O arquivo corrigido foi submetido ao programa de comparação para totalização de erros. Considerando que cada palavra quebrada gera dois erros (um para cada pedaço de palavra), foram somados dois erros para cada uma das palavras quebradas ao total de erros obtidos pelo programa de comparação. Sendo assim, o erro total obtido pode ser calculado da seguinte forma:

Equação 14: Cálculo do erro do *Brill Tagger*

$$\text{Erro} = \frac{\text{N}^\circ \text{ tokens incorretos no arquivo de teste original} + (2 \times \text{N}^\circ \text{ palavras quebradas})}{\text{N}^\circ \text{ de tokens no arquivo padrão}}$$

Segundo [NLTK 2004] (capítulo 4.3 - *Evaluating Taggers*), o desempenho dos rotuladores estatísticos depende da qualidade do conjunto de dados de treinamento, sendo que em particular o tamanho do conjunto de treinamento tem grande influência no desempenho final do rotulador.

Apesar do *Brill Tagger* usar uma abordagem de Aprendizado Supervisionado, é de se esperar que o tamanho do *Corpus* também influencie no seu desempenho. Em uma abordagem de Aprendizado não Supervisionado, [Brill 1999] relata ter obtido uma taxa de acertos 95,5% para uma fatia de 120.000 palavras do *Brown Corpus* usadas como *Corpus* de treinamento. Para uma fatia do *Brown Corpus* de 300.000 palavras, a taxa de acertos do *Brill Tagger* subiu para 96%.

Considerando que o tamanho do *Corpus* de treinamento neste trabalho é pequeno (cerca de 80 vezes menor), e que o desempenho dos rotuladores em geral costumam estar ligado ao tamanho do *Corpus*, a taxa de acertos de aproximadamente 91% é bastante satisfatória.

6.5 Avaliação de Algoritmos de PLN

6.5.1 Avaliação da Segmentação de Sentenças

Para a avaliação do algoritmo de segmentação de sentenças, foram submetidos ao programa o *Corpus* de teste, devidamente rotulado pelo *Brill Tagger*, e o respectivo arquivo corrigido manualmente, que serviu como padrão de comparação para a avaliação do *Brill Tagger*. Ambos os arquivos usados para avaliação nesta fase geraram saídas em formato xml com as sentenças segmentadas. Cada arquivo possui um total de 856 sentenças.

Através de um programa que utiliza o módulo *Random* existente na distribuição do *Python*, foram sorteadas 86 sentenças através do número da sentença nos arquivos, resultando em dois arquivos para avaliação: um arquivo com as sentenças sem a correção dos rótulos sintáticos e outro com as sentenças devidamente corrigidas. O arquivo sem correção dos rótulos sintáticos será denominado Arquivo Teste e o arquivo com correção dos rótulos sintáticos será denominado Arquivo Padrão.

Ambos os arquivos foram submetidos ao segmentador, cada um gerando uma saída segmentada em formato xml. A saída gerada de cada arquivo foi avaliada manualmente para verificar a ocorrência de segmentos incorretos.

A Tabela 37 mostra os critérios usados para a avaliação dos segmentos em cada arquivo de teste.

Tabela 37: Critérios para a avaliação de segmentos

1) O segmento faz sentido como uma unidade (conceito) Ex: "Arco médio", "partes moles", "escoliose dorsal".
a) Existência de palavras a mais no segmento, comprometendo o sentido do conceito como unidade. Ex: "discreta escoliose dorsal" onde a palavra discreta está apenas qualificando o conceito que deveria ser "escoliose dorsal"
b) Existência de palavras a menos no segmento, comprometendo o sentido do conceito como unidade. Ex: "escoliose" quando deveria ser "escoliose dorsal"
c) Existência de expressões classificadas como segmento, mas que não o são. Ex <SEG>presença/NN</SEG> não faz sentido como conceito
d) Perda de um segmento que não foi reconhecido. Ex: se a expressão "Arco médio" não fosse reconhecida como segmento.

A regra 1 é o ponto principal da avaliação manual, pois através dela as demais são interpretadas. Doravante o caso (a) é chamado de Super (super estimou o tamanho do segmento), o caso (b) é chamado de Sub (subestimou o tamanho do segmento), o caso (c) é chamado de Adicional (segmento que não deveria existir) e o caso (d) é chamado de Perdido (segmento não identificado).

A análise de desempenho foi realizada para os dois testes com o intuito de verificar a influência do rotulador nos resultados da fase de segmentação. A Tabela 38 descreve as características gerais obtidas como saída para cada arquivo segmentado.

Tabela 38: Descrição dos arquivos Teste e Padrão (para as frases sorteadas)

	Teste	Padrão
1) Quantidade de sentenças	86	86
2) Estimativa do N° segmentos esperados	244	246
3) Quantidade de segmentos encontrados	248	253
4) Média de segmentos por sentença	2,88	2,94
5) Desvio da média de segmentos/sentença	1,03	1,11

Tabela 39: Descrição dos arquivos Teste e Padrão (completos).

	Teste	Padrão
1) Quantidade de sentenças	856	856
2) Estimativa do N° segmentos esperados	2388	2477
3) Quantidade de segmentos encontrados	2424	2465
4) Média de segmentos por sentença	2,81	2,90
5) Desvio da média de segmentos/sentença	1,14	1,13

Como não há um padrão de referência que possa ser usado para a avaliação do algoritmos de segmentação, pois a criação de um arquivo de referência requer a criação manual do mesmo ou correção do arquivo segmentado automaticamente, foi necessário fazer uma estimativa da quantidade de segmentos que deveria existir em cada um dos arquivos testados (item 2 da Tabela 38 e da Tabela 39), para que se possa calcular a taxa de recuperação e a precisão do algoritmo. A estimativa foi baseada nos seguintes critérios:

- ✓ Todo substantivo deve estar contido em um segmento, portanto o número de segmentos é no mínimo igual ao número de substantivos;
- ✓ Todo advérbio deve estar em um segmento. No pior dos casos, existiria um segmento para cada advérbio;
- ✓ Os adjetivos devem estar em segmentos acompanhando substantivos, advérbios ou sozinhos. Provavelmente os adjetivos que não sejam adjacentes a substantivos ou advérbios devem formar segmentos sozinhos. Em uma contagem no *Corpus* de teste submetido ao *Brill Tagger*, verificou-se que cerca de 0,7% dos adjetivos não são adjacentes a substantivos ou advérbios, então assumiu-se esta quantidade de segmentos formados só por adjetivos.

Seguindo os critérios descritos, temos então que o número estimado de segmentos em cada arquivo de teste é dado por:

Equação 15: Cálculo da estimativa de segmentos nos arquivos de teste do algoritmos de segmentação

$$\text{Estimativa do N}^\circ \text{ segmentos} = \text{Total de substantivos} + \text{total de advérbios} + 0,7 * \text{Total de adjetivos}$$

Os erros obtidos no arquivo Teste e Padrão, foram avaliados manualmente segundo os tipos de erro já descritos. Como a avaliação foi realizada sobre uma amostra de dados e não sobre o arquivo segmentado inteiro, foi utilizado o teste de hipótese para estimar o erro do conjunto inteiro de dados segmentados tanto para o arquivo de Teste como para o arquivo Padrão. A Tabela 40 e a Tabela 41 mostram os erros obtidos para os arquivos de Teste e Padrão respectivamente. A coluna nomeada como “Outros Erros” refere-se a erros ocorridos na segmentação devido a comportamento inesperado do programa.

Tabela 40: Resultados da segmentação para o arquivo Teste

	Adicional	Super	Sub	Perdido	Outros Erros	Total
Total	7	15	15	5	5	47
%	2,82%	6,05%	6,05%	2,02%	2,02%	18,95%
Média de erros por sentença	0,0843	0,1744	0,1744	0,0581	0,0581	0,5465
Desvio médio (erros/sentença)	0,1544	0,2880	0,3042	0,1122	0,1095	0,6609

Tabela 41: Resultados da segmentação para o arquivo Padrão

	Adicional	Super	Sub	Perdido	Outros Erros	Total
Total	7	12	9	5	4	37
%	2,77%	4,74%	3,56%	1,98%	1,58%	14,62%
Média de erros por sentença	0,0864	0,1481	0,1111	0,0617	0,0494	0,4302
Desvio médio (erros/sentença)	0,1579	0,2524	0,2058	0,1158	0,0939	0,5703

A Tabela 42 mostra os resultados obtidos como estimativa do erro para os arquivos de Teste e Padrão.

Tabela 42: Teste de hipótese para os arquivos Teste e Padrão

	$\hat{p}$	α	Z_{α}	Significância	Erro estimado
Teste	0,1895	0,4495	1,6400	5%	24%
Padrão	0,1462	0,4495	1,6400	5%	19%

Os valores de Z_{α} e α foram obtidos em [McClave 2000], Apêndice A, Tabela IV, página 809 (*Normal Curve Areas*). As fórmulas usadas para o cálculo do teste de hipótese foram obtidas em [McClave 2000] (capítulo 8, pg. 349, *Large-Sample Test of Hypothesis About p*). O valor de $\hat{p}$ corresponde à proporção de erro na amostra, sendo obtido segundo a Equação 16.

Equação 16: Cálculo da proporção do erro na amostra ($\hat{p}$)

$$\hat{p} = \frac{(N^{\circ} \text{ de segmentos incorretos na amostra})}{(\text{Total de segmentos na amostra})}$$

Para se encontrar o erro estimado, inicialmente foi assumida uma distribuição normal do erro. O próximo passo foi determinar a significância aceita para o teste, no caso 5%. Através da tabela *Normal Curve Areas* [McClave 2000], foram encontrados os valores de Z_{α} e α para o nível de significância assumido. Definidos os valores das variáveis envolvidas, estes foram substituídos nas fórmulas já referenciadas para se obter o erro estimado da população, isto é, o erro estimado caso a avaliação tivesse sido realizada em todo o arquivo de Teste e Padrão.

Segundo os valores estimados para o arquivo Teste, a taxa de erros dos algoritmos de segmentação de sentenças é menor que 24%, com uma significância de 5%. Para o arquivo Padrão, temos que a taxa de erros estimada é menor que 19%, com 5% de significância.

Para completar a análise de resultados dos algoritmos de segmentação de sentenças, foram calculados os valores da Recuperação e Precisão, para ambos os arquivos, o Teste e o Padrão. Os valores obtidos foram calculados sobre a amostra verificada manualmente.

Tabela 43: Valores obtidos para a Recuperação e a Precisão no algoritmo de segmentação

	Recuperação	Precisão
Arquivo Teste	0,8180	0,8105
Arquivo Padrão	0,8791	0,8538

Em um trabalho que usa uma abordagem híbrida entre regras e probabilística para a identificação de sintagmas nominais, [Chen 1994] relata a obtenção de 96% de Recuperação e 95% de Precisão. Considerando que a abordagem deste trabalho é puramente baseada em regras e que o algoritmo ainda pode ser melhorada através do refinamento de algumas regras, os valores de Recuperação e Precisão encontrados são satisfatórios.

Apesar das abordagens probabilísticas oferecerem melhores resultados, possuem a desvantagem de necessitarem de um *Corpus* de treinamento previamente anotado. A montagem de um *Corpus* para treinamento, requer uma quantidade grande de material e um trabalho manual extenso para anotação para cada área que se deseje processar textos, pois os algoritmos probabilísticos normalmente são sensíveis ao *Corpus*, isto é, o desempenho seria muito baixo se tentássemos processar um laudo de Raio-X de Tórax com um algoritmo treinado com o CETENFolha⁴ [CETENFolha], por exemplo.

Com relação as diferenças obtidas entre o desempenho do arquivo Teste e Padrão, era de se esperar que os erros da fase de rotulação sintática diminuíssem o desempenho da fase de segmentação. Notou-se uma diferença significativa de desempenho, na casa dos 5%, considerando o erro estimado através do teste de hipótese, no entanto, uma vez que obteve-se um segundo arquivo padrão da fase de rotulação sintática, o *Brill Tagger* pode ser treinado novamente, com um *Corpus* maior, o que deve melhorar o desempenho de toda a cadeia de processamento.

Apesar de as taxas de erros serem de 24% e 19% para os arquivos Teste e Padrão respectivamente, notamos que a maioria dos erros são do tipo Super ou Sub. Os erros do tipo Super podem ser facilmente compensados na próxima fase de processamento. Os erros do tipo Sub também podem ser compensados, porém com um maior esforço. Para que haja a compensação dos erros do tipo Sub, primeiro é preciso identificar os conceitos menores encontrados, e tentar junta-los através da árvore semântica da hierarquia, porém sempre haverá uma taxa de segmentos subestimados que não será compensada.

⁴ CETENFolha (Corpus de Extractos de Textos Electrónicos NILC / Folha de S. Paulo): *Corpus* em português constituído de artigos extraídos do jornal Folha de São Paulo.

No casos de segmentos adicionais, isto é, trechos de texto identificados como segmentos erroneamente, estes segmentos dificilmente serão associados a algum conceito na hierarquia. Os segmentos perdidos são a maior fonte de problemas, pois um segmento não identificado significa informação perdida.

6.5.2 Avaliação da Identificação de Conceitos

A avaliação da identificação de conceitos consiste em verificar se os segmentos marcados na fase anterior de processamento (fase de segmentação) foram realmente identificados como conceitos em algum sistema de terminologias médico.

O procedimento de avaliação foi realizado sobre uma amostra de dados retirada da saída do arquivo Padrão da fase anterior tendo sido calculado o devido teste de hipótese para estimar o erro do teste como um todo. Quanto aos cálculos da precisão e da recuperação nesta fase pode-se assumir duas abordagens.

Poderíamos realizar os cálculos considerando a quantidade de conceitos relevantes e irrelevantes existentes na base de dados das ontologias usadas, porém julgou-se que esta abordagem é mais adequada para avaliar as próprias ontologias do que o algoritmo de rotulação de conceitos.

A outra abordagem diz respeito a avaliar a quantidade de segmentos identificados no arquivo de entrada do algoritmo e verificar a quantidade de conceitos recuperados e com que taxa de acerto. Esta foi a abordagem adotada.

Nesta fase temos que um segmento equivale ao segmento de texto identificado na fase anterior, um termo equivale a um conceito identificado em um dos sistemas de ontologias e uma sentença é uma frase completa pertencente a coleção de frases distintas do conjunto de laudos.

A escolha da amostra foi realizada através de um *script* escrito em *Python* que através do módulo *random* sorteou 130 sentenças, sendo que para cada sentença foi sorteado 1 segmento. Caso a sentença não possuísse nenhum termo identificado, computou-se como um segmento não encontrado.

Tabela 44: Características gerais da amostra para avaliação da rotulação de conceitos

	Valor	Proporção termos/segmentos
Quantidade de sentenças na amostra	130	-
Total de segmentos de entrada	360	-
Total de termos identificados referente ao total de sentenças	201	55,83%

A Tabela 44 descreve as características gerais do teste utilizado para a avaliação do rotulador de sentenças. Note que o total de segmentos testados não é o mesmo que o total do arquivo de entrada. Na realidade foi selecionado um segmento por sentença, tendo sido mantida a sentença completa apenas como referência para a avaliação da saída do programa, feita manualmente.

A Tabela 45 mostra os valores obtidos como resultado do teste. Percebemos que houve a identificação apenas de 60% dos termos testados, porém considerando os resultados anteriores de avaliação de ontologias, já não se esperava um resultado muito alto para o reconhecimento de conceitos, devido a pobreza dos vocabulários usados como referência. Mesmo utilizando o Radlex, um vocabulário voltado à radiologia, a versão utilizada é uma versão de desenvolvimento ainda, sendo contendo apenas cerca de 1300 definições.

Percebemos a pobreza do Radlex em relação ao UMLS ao verificar que a quantidade de termos que foram identificados através do UMLS é bem maior que a taxa de termos identificados através do RadLex. Se o UMLS não é adequado para o processamento de laudos de Raio-X de Tórax, o RadLex, em sua versão inicial, demonstrou-se também inadequado.

Tabela 45: Valores obtidos na avaliação do rotulador de termos

	Valor	Proporção com o total de termos
Total de segmentos	130	-
Termos testados	130	-
Quantidade de termos identificados	78	60,00%
Quantidade de segmentos não identificados com termo algum	52	40,00%
Quantidade de termos identificados e corretos	64	49,23%
Quantidade de termos identificados cuja quantidade de palavras do segmento é igual à do termo.	34	26,15%
Quantidade de termos identificados e corretos cuja quantidade de palavras do segmento é igual à do termo.	34	26,15%
Quantidade de termos Radlex identificados	25	19,23%
Quantidade de termos UMLS identificados	53	40,77%

As causas que levaram a 40% de segmentos não serem identificados são várias e podem assumir critérios subjetivos. A causa mais óbvia é a deficiência dos vocabulários usados como referência, no entanto se poderia atribuir parte dos problemas à forma como os conceitos são definidos nas ontologias em comparação a forma como o texto é segmentado.

Por exemplo, notamos que cerca de metade dos segmentos identificados foram reconhecidos na forma exata como entraram no programa de rotulação, a outra metade sofreu cortes na sequência original de palavras até que pudessem ser identificados com algum conceito. Esses conceitos acabam por agregar um valor mais genérico ao trecho de texto. Um exemplo seria o caso de se identificar o conceito “escoliose” ao invés de identificar “escoliose dorsal”. A questão de o dicionário de termos médicos definir o termo “escoliose dorsal” em complemento aos conceitos “escoliose” e “dorso”, é uma questão de interpretação e modelagem de quem construiu o sistema de terminologias.

Apesar de cerca de metade dos termos terem sido identificados em uma forma mais simples, isto é, com menos palavras, a próxima fase de processamento deveria resolver problemas como o exemplificado com o conceito “escoliose dorsal”, pois esta fase deveria associar os conceitos que acabaram sendo rotulados individualmente.

Tabela 46: Proporção segmentos X termos

	Valor
Segmentos idênticos a termos relativo ao total identificado	43,59%
Segmentos idênticos a termos e corretos, relativo ao total identificado	43,59%

Tabela 47: Taxa de acerto/recuperação do rotulador de conceitos para a amostra

	Valor	Proporção
Taxa de recuperação	0,6000	60,00%
Taxa de acerto	0,8205	82,05%
Taxa de erro	0,1795	17,95%

Tabela 48: Teste de hipótese a avaliação do rotulador de conceitos

	$\hat{p}$	α	$Z\alpha$	Significância	Erro estimado
Padrão	0,1795	0,4495	1,6400	5%	25%

Fechando a análise, obteve-se uma taxa de erro na amostra de 17,95%, considerando apenas termos identificados, isto é, dos termos rotulados com o código de um conceito, 17,95% receberam rótulos incorretos. Considerando o teste de hipótese, chegamos a um erro estimado menor que 25%, com cerca de 5% de significância para o experimento como um todo.

Trata-se de um resultado modesto, mas que ainda pode ser melhorado com o aprimoramento das fases anteriores, das ontologias usadas como referência de terminologias e dos próprios algoritmos implementados para esta tese.

No caso do algoritmo de rotulação de conceitos, algumas melhorias que se tem em vista são:

- ✓ Implementar a aglutinação de segmentos próximo para busca, no caso de segmentos subestimados. Aqui poderíamos considerar algumas preposições por exemplo ou fazer um melhor tratamento de conjunções como “e” ou “ou”.
- ✓ Melhorar o método de quebra dos segmentos superestimados, que é feita no momento com o foco nos substantivos e adjetivos;
- ✓ Fazer um melhor tratamento da escolha da melhor resposta retornada pelo algoritmo de busca. Atualmente a resposta escolhida para rotular um termo é baseado na melhor pontuação obtida pelo algoritmo de comparação entre *strings*, não havendo critério de desempate no caso de dois conceitos com a mesma pontuação. O ultimo conceito processado acaba por ser escolhido. Neste ponto se poderia ao invés de escolher apenas um conceito por termo, selecionar uma lista de termos e a partir da lista, escolher o conceito final através de um processo de contextualização baseado na informação semântica da hierarquia de conceitos. A contextualização deve ser feita na fase final de processamento.

Apenas a título de visualização da saída da rotulação de conceitos, o Exemplo 26 mostra a forma final de algumas sentenças.

Exemplo 26: Exemplo de rotulação de conceitos

Entrada	<pre> <SENT ORIG="acentuação da cifose e osteófitos dorsais."> <SEG>acentuação/NN</SEG><COMP>da/TN</COMP> <SEG>cifose/NN</SEG><COMP>e/CC</COMP> <SEG>osteófitos/NNS dorsais/JJS</SEG><COMP>./.</COMP> </SENT> </pre>
Saída:	<pre> <SENT ORIG="acentuação da cifose e osteófitos dorsais."> <TERM ID="C0022821" SRC="umls" SEARCH="cifose/NN" FOUND="cifose" SCORE="1.0"></TERM> <TERM ID="C0004600" SRC="umls" SEARCH="dorsais/JJS" FOUND="dorso" SCORE="1.0"></TERM> </SENT> </pre>
Entrada	<pre> <SENT ORIG="aorta alongada , tortuosa com calcificações parietais na croça arco médio normal."> <SEG> <SEG>aorta/NN alongada/JJ</SEG><COMP>./.</COMP> <SEG>aorta/NN tortuosa/JJ</SEG> </SEG> <COMP>com/TN</COMP> <SEG>calcificações/NNS parietais/JJS</SEG> <COMP>na/TN</COMP> <SEG>croça/NN</SEG> <SEG><SEG>arco/NN médio/JJ</SEG> normal/RB</SEG> <COMP>./.</COMP> </SENT> </pre>
Saída:	<pre> <SENT ORIG="aorta alongada , tortuosa com calcificações parietais na croça arco médio normal."> <TERM ID="C0003483" SRC="umls" SEARCH="aorta/NN" FOUND="aorta" SCORE="1.0"></TERM> <TERM ID="C0741171" SRC="umls" SEARCH="aorta/NN tortuosa/JJ" FOUND="aorta tortuosa" SCORE="1.0"> <TERM ID="RL906" SRC="radlex" SEARCH="tortuosa/JJ" FOUND="tortuosidade" SCORE="1.0"></TERM> </TERM> <TERM ID="RL867" SRC="radlex" SEARCH="calcificações/NNS" FOUND="calcificação" SCORE="1.0"></TERM> </SENT> </pre>

6.6 Avaliação Geral

Apesar de não ter sido possível chegar a última fase de processamento dos laudos, devido ao tempo, os resultados obtidos são animadores, podendo ainda ser melhorados através de ajustes em cada fase do processamento, desde a rotulação sintática que pode ser melhor treinada, passando pelo ajuste das regras de segmentação até alcançar a rotulação de conceitos, que requer o uso de um sistema de terminologias mais adequado.

O desempenho mais baixo, referente à taxa de recuperação, na última fase implementada em grande parte se dá à pobreza dos vocabulários com relação às necessidades de uma aplicação de processamento de laudos de Raio-X de Tórax. Na verdade não importa a área ou a linguagem escolhidas para trabalhar com algoritmos de PLN, o desempenho da aplicação sempre estará fortemente ligado à qualidade do léxico utilizado, pois se o léxico não contempla o vocabulário dos textos processado, não há como reconhecer o texto por melhor que sejam os algoritmos utilizados.

Apesar de não ter sido implementada a última fase de processamento, a fase de rotulação de conceitos já agrupa alguns termos, como podemos observar na segunda sentença do Exemplo 26.

Nesta tese optou-se por uma abordagem baseada em regras em todos os passos do processamento dos laudos, desde a extração das raízes das palavras, da rotulação sintática, da extração de segmentos até a rotulação dos mesmos. Apesar de as abordagens baseadas em regras normalmente apresentarem um desempenho mais baixo, elas possuem uma grande vantagem, por não precisarem de *Corpus* de treinamento. O único ponto deste trabalho onde foi necessário um *Corpus* de treinamento, foi na utilização do *Brill Tagger*, porém este algoritmo se mostrou eficaz, mesmo sendo treinado com um *Corpus* pequeno.

Considerando todas as fases do processamento implementadas, em nenhuma obteve-se um resultado inferior aos 70% de acerto, um resultado modesto mas animador, se considerada a abordagem baseada totalmente em regras e a dificuldade de obtenção de ferramentas e dados para o trabalho na língua portuguesa.

7 DISCUSSÃO

7.1 Avaliação do Desenvolvimento

No capítulo de metodologias foram descritas algumas premissas, sugeridas por [Friedman 1997] para o desenvolvimento de aplicações de PLN, por se julgar que tais premissas são importantes para a organização de um trabalho e para a obtenção de resultados mais fidedignos. No entanto, nem todas as premissas puderam ser obedecidas devido a restrições no projeto. Apesar de se ter desviado em alguns pontos das orientações de Friedman, buscou-se ao longo do trabalho condições de contorno que minimizem o impacto sobre o resultado final do trabalho.

A primeira premissa de Friedman determina que o desenvolvedor não deve participar do projeto da avaliação de um sistema de PLN, bem como da própria execução da avaliação, além de não dever ter acesso ao conjunto de dados de teste. Na prática, com uma equipe reduzida é muito difícil respeitar tal premissa, exigindo do desenvolvedor maior capacidade para se auto-policiar durante o trabalho para que sejam obtidos resultados de maior valor científico.

Se por um lado se perde a capacidade de gerar um produto mais genérico, por outro, podemos perder a possibilidade de gerar uma série de otimizações quando o objetivo é bem específico. Para se evitar que os prováveis vícios do produto final, causados pelo conhecimento prévio dos dados, torna-se necessário um processo de avaliação mais rigoroso e elaborado. Mesmo que a avaliação seja feita por pessoas envolvidas no desenvolvimento, ela ainda pode ser submetida a terceiros para apreciação.

Como o desenvolvedor acaba por interagir em todas as fases, desde implementação até avaliação, é importante que todos os métodos e critérios sejam bem definidos no decorrer do trabalho. A utilização de métodos e critérios bem definidos, torna mais fácil identificação dos pontos onde o desenvolvimento como um todo pode ter falhado ou provocar falhas. É claro os métodos e critérios são importantes em qualquer trabalho com qualquer equipe, porém em uma situação onde vícios podem ser introduzidos devido a situação de trabalho não ser ideal, sua importância adquire uma dimensão muito maior, pois é através deles que um observador externo poderá identificar os pontos críticos do trabalho.

Com relação ao estabelecimento de um padrão de referência, o tamanho da equipe envolvida no projeto e a disponibilidade de recursos, considerando a língua portuguesa, também são fatores que aumentam a distância entre a realidade e as premissas estabelecidas por Friedman. Procurou-se aqui utilizar outros trabalhos e a adoção de sistemas de terminologias padronizados já existentes para minimizar o impacto referente ao tamanho da equipe sobre o trabalho final.

A descrição dos métodos de avaliação foi introduzida no capítulo de metodologias, procurando-se deixar claro os passos dados para a obtenção de resultados e também para o estabelecimento de um roteiro inicial do trabalho. A obtenção dos dados para teste foi descrita mais detalhadamente no capítulo de discussão de resultados, com a esperança de que tenha ficado claro ao leitor o processo de avaliação dos resultados finais.

A avaliação de resultados levou em consideração uma série de testes realizados de forma a mostrar não só os pontos fortes do sistema desenvolvido, mas também os pontos falhos com a sugestão das possíveis melhorias. Para a análise de dados, em alguns momentos procurou-se utilizar mais de uma metodologia, com a finalidade de cobrir pontos não tratados em um método ou outro. Como exemplo cita-se a utilização em paralelo das avaliações tradicionais em PLN e da Curva ROC.

Em alguns pontos onde não foi possível proceder com uma avaliação dos dados por completo, optou-se por usar amostragens, validando os experimentos através dos Testes de Hipótese. Apesar de não ser o método mais perfeito, pois o que acabamos obtendo são aproximações do erro de cada programa testado, este tipo de abordagem é uma alternativa ao trabalho manual mais extenso de criação de padrões de referência (*Gold Standards*) ou *Corpus* previamente anotados. Em um trabalho que possua uma equipe maior e mais tempo, a criação dos Padrões de Referência ou de *Corpus* anotados previamente, leva a resultados mais precisos.

7.2 Sobre Ontologias

Uma ontologia é um modelo que representa a realidade sob um determinado ponto de vista em um determinado momento. Como todo modelo, dificilmente a ontologia é completa estando amarrada à visão de mundo dos indivíduos que construíram este modelo em determinado ponto da linha do tempo. Sob este ponto de vista fica mais simples entender o

por que sistemas de terminologias como o UMLS não contemplam satisfatoriamente um vocabulário tão específico como o de radiologia de tórax, por exemplo. No caso do UMLS ele não representa o ponto de vista dos radiologista simplesmente por não agregar vocabulário algum específico para esta área. Na realidade o UMLS é um apanhado geral de vocabulários, o que o torna genérico, portanto superficial quando pensamos em uma área tão específica como Raio-X de Tórax.

Para quem desenvolve sistemas de computação em geral, não apenas em PLN e EI, existe a noção clara de que é muito difícil obter qualquer coisa genérica, que atenda as necessidades de qualquer tipo de usuário ou que desempenhe diversas tarefas em níveis e escopos diferentes satisfatoriamente. Não poderia ser diferente para um sistema de terminologias em medicina. Cada área possui muitas particularidades, sendo difícil abranger todas em um único sistema.

Passamos então a construção de uma ontologia própria, porém é importante estar ciente de que esta ontologia ainda estará amarrada ao processo desenvolvido no InCor, pois deve assimilar os hábitos e a linguagem usada costumeiramente neste hospital. Apesar da construção de uma ontologia própria ser o mais adequado para processar os próprios dados, este é um trabalho árduo que requer o envolvimento de diversos especialistas, o que não foi possível realizar para esta tese por limitações de tempo, equipe ou mesmo financeiras. Apenas para se ter uma idéia entre a decisão de se começar a utilizar o RadLex e a liberação de uma versão mais completa, passou-se mais de um ano, apesar de que o RSNA possui uma equipe dedicada a este trabalho, com o fator agravante de se tratar de uma ontologia definida em língua inglesa. O tempo de desenvolvimento não desmerece o trabalho do RadLex, pois aqui o que conta é a criação de um sistema que padronize as terminologias usadas em radiologia.

Resta agora o trabalho de traduzir e organizar o RadLex para a língua portuguesa e para o uso em PLN, pois na forma como é distribuído, precisa de alguma reorganização para que se possa utilizar o seu conteúdo em programas de computadores voltados à EI e RI. O UMLS possui a vantagem de ser projetado já com esta intenção, porém todo o conjunto de ferramentas voltados ao PLN, existe apenas em inglês.

Apesar de todo trabalho ainda necessário para se obter um sistema de ontologias em língua portuguesa e que se adeque às necessidades das aplicações de PLN, neste trabalho procurou-se dar os primeiros passos, através de metodologias e geração de material, que

dependendo apenas da autorização dos proprietários do conteúdo original, pode vir a ser disponibilizado para outros pesquisadores na área de PLN e aplicações médicas que trabalham com língua portuguesa.

7.3 Sobre Formatação e Restrição de Vocabulários

Os profissionais de informática sempre tendem a querer criar formulários com campos formatados e fixos enquanto entre profissionais de saúde a manutenção de textos livres é um requisito importantíssimo. Como profissionais de informática, enxergamos que uma formatação mais rígida trás diversos benefícios do ponto de vista computacional, porém nos esquecemos que ao impormos uma formatação mais rígida estamos restringindo a liberdade de expressão. Como os profissionais de saúde não abrem mão da liberdade de expressão, com toda a razão, resta-nos partir para a aplicação de técnicas PLN para obter a formatação que precisamos para satisfazer as necessidades de sistemas computacionais.

Além da padronização de um vocabulário ser importante para melhorar a qualidade e a clareza dos textos médicos, em nosso caso os laudos de Raio-X de Tórax, ainda é importante não restringir os profissionais a um vocabulário predefinido, o que poderia implicar a restrições no conteúdo semântico do texto. Por isso, um sistema de terminologia deve servir como vocabulário recomendado e não obrigatório. Neste ponto a utilização de sistemas de terminologias ou ontologias passa a ser importante, pois fornece a base para o processamento de linguagem natural e a padronização necessária aos profissionais.

Infelizmente um dos pontos onde sistemas de extração de informação ou classificação automática falham é na fuga do vocabulário recomendado, no entanto os pontos onde há falhas nos sistemas automatizados não devem ser encarados como problemas que simplesmente atrapalham nossos resultados, mas como indicativos de que alguma revisão deve ser feita na ontologia usada ou nos próprios algoritmos.

Friedman aponta outros pontos que influenciam no desempenho dos sistemas de PLN, pontos estes ligados à representação, complexidade, variação da expressividade, agrupamento da informação, inconsistência e informação implícita [Friedman 1997].

7.4 Sobre PLN em Português

A criação de ferramentas de PLN em língua portuguesa do ponto de vista de algoritmos básicos se encontra bem resolvida, percebemos este fato ao encontrar algoritmos prontos como O *Snowball Stemmer* ou o *Brill Tagger* que neste trabalho provaram ter um desempenho bom em português, apesar da necessidade de algumas pequenas alterações ou adaptações.

Na prática, o maior problema que enfrentamos ao trabalhar com PLN em língua portuguesa é a falta de material devidamente formatado. Se computarmos o custo de montagem de um *Corpus* para cada área da medicina que se deseje processar laudos, comparado ao custo de se mudar algumas regras dentro de um *software*, o segundo é bem menor se o *software* for projetado de forma flexível às adaptações. A montagem de um *Corpus* requer trabalho manual extenso de uma equipe de profissionais especializada na área de trabalho, além de uma grande quantidade de material de alta qualidade.

Fugindo da montagem dos *Corpus*, ainda resta a necessidade de criação de léxicos e ontologias para a área escolhida para o processamento de textos. Este é um trabalho árduo e que consome bastante tempo, que não pode ser evitado através de abordagem como *softwares* baseados em regras. Pode existir algum nível de automatização neste processo, porém grande parte do trabalho de organização de uma ontologia e de um léxico é manual.

Pode-se optar por utilizar sistemas já prontos, porém apesar de existirem diversos sistemas de terminologias médicas prontos, estes acabam por ser projetados segundo a visão e a necessidade de terceiros, o que acarreta em algum trabalho de adaptação para as próprias necessidades de cada aplicação e para a língua que se deseje trabalhar. O que ainda não exime os desenvolvedores de aplicações de PLN em medicina de algum trabalho manual.

Apesar das dificuldades encontradas na utilização de ontologias prontas, elas ainda são muitos úteis, pois, seguindo o conselho de [Noy 2001], é sempre melhor projetar uma ontologia a partir de trabalhos já existentes do que começar do zero.

Se pensarmos que o conhecimento é cumulativo e que a estrutura de dados deste trabalho nasceu da própria exploração do RadLex ou do UMLS, além da consulta a bibliografia e profissionais da área, percebemos por que a sugestão de [Noy 2001] é importante. Se não houvesse a definição prévia em uma ontologia de conceitos médicos, teria

o projetista deste trabalho visualizado qual a informação desejável de se extrair em um laudo em radiologia? As ontologias médicas acabam por fornecer o conhecimento sobre a organização do conteúdo médico e o material para a montagem dos dicionários e léxicos necessários ao PLN.

7.5 Sobre as Ferramentas Desenvolvidas

Tanto ferramentas como modelos desenvolvidos para esta tese foram estruturados com o objetivo de diminuir o esforço para adaptação do uso para outras áreas da medicina. Para avaliar se este objetivo foi alcançado, seria necessário proceder com a adaptação, medindo os esforços necessários para tal tarefa.

Com relação ao desempenho, foram alcançados resultados satisfatórios para a rotulação sintática, apesar do *Corpus* de treinamento reduzido, e para a segmentação de sentenças. Com relação ao rotulador sintático, este pode ser treinado novamente, agora usando além do *Corpus* de teste inicial, o padrão de referência para avaliação de desempenho do algoritmo. Com relação a segmentação de sentenças, pode-se ainda refinar as regras de segmentação. Este refinamento pode levar a um melhor desempenho com relação a tempo de execução e com relação a identificação dos segmentos.

Com relação à rotulação de conceitos, esta depende muito da melhoria da ontologia usada como vocabulário de referência e da melhoria de algumas regras que dizem respeito a unificação de segmentos subestimados. Os segmentos superestimados já recebem um tratamento, que pareceu satisfatório, visto que muitos dos conceitos reconhecidos nesta fase tinha comprimento em quantidade de palavras menor que o segmento inicial.

7.6 Sobre os Objetivos Alcançados

Seguindo a proposta inicial do trabalho o objetivo principal era gerar uma aplicação capaz de anotar laudos de Raio-X de Tórax adicionando informação semântica que viesse a ser utilizada para auxiliar posteriormente na utilização de algoritmos de RI. O objetivo final foi alcançado parcialmente. Apesar dos resultados buscados como objetivo inicial terem sido alcançados parcialmente, obteve-se diversos resultados derivados. Resumidamente os resultados finais obtidos foram:

- ✓ Desenvolvimento de um método de avaliação de Ontologias com relação a uma área de conhecimento específica;
- ✓ Obtenção de um *Corpus* anotado sintaticamente de laudos de Raio-X de Tórax;
- ✓ Projeto da ontologia e da estrutura de dados para anotação de laudos e mapeamento da ontologia de laudos para a ontologia de conceitos médicos;
- ✓ Implementação e avaliação dos algoritmos de processamento intermediário para a anotação final dos laudos em português, sem que haja a necessidade de organização de material de treinamento e padrões de referência;
- ✓ Desenvolvimento de uma metodologia e de aplicativos para a estruturação de um léxico baseado no vocabulário de um sistema de ontologias;
- ✓ Desenvolvimento de uma metodologia para tradução e adaptação de uma ontologia para a língua portuguesa.

Apesar de não ter sido possível implementar a última fase de processamento de laudos, como foi proposto inicialmente, toda a base para seu desenvolvimento se encontra pronta ou encaminhada.

Do ponto de vista de desempenho o trabalho alcançou resultados satisfatórios, considerando ainda a criação de toda uma estrutura e de metodologias não apenas de desenvolvimentos de aplicações PLN para o português, mas também de adaptação de material publicado originalmente em inglês.

Do ponto de vista das ontologias, apesar destas se demonstrarem inicialmente pobres com relação à área de conhecimento escolhida para o desenvolvimentos desta tese, ainda há a perspectiva de evolução das mesmas, restando o trabalho de adaptação e tradução para o português, através dos métodos e programas desenvolvidos para esta tese.

7.7 Continuidade do Trabalho

Os dois pontos principais para a continuidade desta tese é a implementação da fase final de processamento para anotação dos laudos e a adaptação da nova versão do RadLex para o Português.

O trabalho de estruturação da nova versão do RadLex para o português requer não apenas a tradução, mas algumas pequenas adaptações a serem feitas na hierarquia de

conceitos para melhorar o desempenho dos programas desenvolvidos para esta tese. Uma importante adaptação sugerida é a adição de informação sintática aos conceitos, o que poderia melhorar o resultados de algoritmos de busca ou mesmo ajudar a remover a ambigüidade entre palavras ou conceitos. Esta adaptação requer um estudo ainda a ser realizado e deve resultar na revisão do mapeamento entre ontologia de laudos e ontologia de conceitos médicos. Tendo estruturado a nova versão do RadLex, será necessário preencher novamente as estruturas de dados como o índice de palavras normalizadas por exemplo.

A implementação da fase final de processamento requer que a ontologia e o léxico, em sua nova versão, sejam finalizados. Pretende-se manter a abordagem baseada em regras para mais esta fase, tendo sido estudada inicialmente a utilização de algoritmos de busca como *Forward Chaining* ou *Backward Chaining* para varrer a hierarquia e fazer a inferência sobre a organização dos conceitos e escolha dos rótulos XML adequados para cada trecho de laudo.

Concluídas as duas tarefas descritas, pode-se passar a pensar na adaptação de todo o aparato desenvolvido para a anotação de laudos em novas áreas médicas.

Por fim, os resultados obtidos em cada fase intermediária ainda podem ser refinados, segundo critérios de melhoria dos algoritmos já descritos na análise de resultados de cada um deles.

8 CONCLUSÕES

No início da pesquisa sobre PLN em língua portuguesa, foi descoberta uma literatura abrangente sobre a área e a existência de diversas ferramentas e algoritmos implementados para a língua inglesa. Toda a literatura e material disponível na área a princípio gerou bastante expectativa com relação ao desenvolvimento de uma pesquisa na área de PLN para a língua portuguesa, uma combinação pouco explorada. O fato de não existirem muitas ferramentas em português para PLN acabou por se revelar uma das maiores dificuldades do trabalho.

Segundo Bolshakov e Gelbukh [Bolshakov 2004], o maior esforço no desenvolvimento de aplicações de PLN não está na criação do *software* e sim no trabalho extenso e tedioso de extração de dados para criação de dicionários e tabelas gramaticais.

A veracidade da afirmação de Bolshakov e Gelbukh se constatou quando foi percebido que a simples adoção de um sistema de terminologias não era suficiente para alimentar os algoritmos de PLN com dados de referência para o processamento dos laudos. A partir deste momento se iniciou um processo de análise e organização dos dados contidos nas ontologias médicas adotadas. Este processo envolveu a concepção de uma estrutura de armazenamento e acesso aos dados pertencentes às ontologias, a criação de diversos programas auxiliares para extração e formatação deste dados e de um método de avaliação de conteúdo. Tudo isso formalizado através dos passos de uma metodologia que possa ser reproduzida de forma a permitir a ampliação dos trabalhos atuais ou a adaptação para o processamento de textos em outras áreas de conhecimento dentro da medicina.

Tendo extraído e reorganizado os dados necessários aos dicionários para a aplicação de PNL voltada ao processamento de laudos, ainda restava uma segunda fase também de extração e organização de dados. Se fosse adotada uma abordagem probabilística no desenvolvimento de algoritmos, seria necessária a criação de um conjunto de *Corpus* de treinamento para cada fase de processamento expostas neste trabalho. Não bastando o fato de que a criação de um *Corpus* de referência ou treinamento de algoritmos ser tediosa e requerer uma grande dose de meticulosidade, o desempenho de algoritmos probabilísticos é normalmente afetado pelo tamanho do *Corpus*. Em linhas gerais, quanto maior o *Corpus*, melhor o desempenho. Não só o tamanho mas também a qualidade tanto dos dados quanto da estrutura que os encapsula são importantes para o desempenho.

Considerando a realidade desta tese, o investimento na criação de um conjunto de *Corpus* apresenta um custo elevado, pois requer uma grande quantidade de material e pessoas com conhecimento especializado que estejam aptas a classificar e anotar esses dados. O material pode ser obtido na base hospitalar. Considerando que se realiza cerca de 2000 exames de Raio-X de Tórax por mês e a base do InCor armazena laudos desde 1996 com acesso *on line* ou em algum tipo de mídia digital, teríamos o potencial de criar um *Corpus* com cerca de 240.000 laudos. Fica evidente que o problema não é necessariamente extrair dados em quantidade suficiente, mas sim em ter recursos humanos e computacionais que possam fazer o trabalho de organização.

Ao recuperar uma modesta quantidade de laudos notou-se mais um fato muito importante. Os laudos possuem um vocabulário e um conjunto de expressões bastante restritos. Considerando todo este contexto, resolveu-se por adotar abordagens puramente baseadas em regras para o desenvolvimento deste trabalho, com a vantagem de não ser necessária a criação de *Corpus* para treinamento de algoritmos, mas com a desvantagem de normalmente apresentarem um desempenho inferior. Aqui nos deparamos com uma questão puramente de engenharia referente aos custos e benefícios de cada alternativa. Optou-se pelo menor custo em trabalho manual, mas alcançando-se benefícios satisfatórios no que diz respeito ao desempenho tanto de cada fase isolada de processamento dos laudos quanto do processamento como um todo.

Apesar da abordagem baseada em regras, o único ponto no processo onde foi necessária a organização de um *Corpus* foi para o treinamento de um rotulador sintático, no caso o *Brill Tagger* fornecido com a distribuição do NLTK. Apesar deste algoritmo requerer um treinamento, ele também é baseado em regras geradas automaticamente na fase de treino.

O fato de os laudos possuírem um conjunto de frases limitado, levou a escolha por uma estratégia de minimização de esforço, onde a coleção de frases foi extraída do conjunto dos laudos e utilizada para criação do único *Corpus* necessário neste trabalho e para desenvolvimentos dos métodos de teste e avaliação, baseados nas frases ao invés dos laudos. Processar somente as frases é menos custoso que processar cada laudos individualmente, tanto do ponto de vista computacional como do ponto de vista do trabalho de avaliação manual utilizado. Os laudos processados podem ser montados através do reagrupamento das sentenças processadas. Se pensarmos nos 240.000 laudos em potencial que o *Corpus* possa assumir, podemos obter um grande ganho de processamento com esta abordagem.

Como em alguns pontos foi adotada uma abordagem de avaliação baseada em amostragem, procurou-se validar os testes nas amostras com métodos estatísticos, no caso o teste de hipótese, usado para estimar o erro do algoritmo avaliado com relação ao experimento completo.

O *Brill Tagger* treinado com um *Corpus* pequeno alcançou uma taxa de aproximadamente 91% de acerto, apesar de a literatura apontar o tamanho do *Corpus* utilizado como fator crítico para o treinamento de algoritmos em PLN. Esse resultado é animador, pois indica que os esforços necessários para a criação de um *Corpus* de treinamento para uma outra área médica, além de Raio-X de Tórax, não é grande.

O algoritmo de busca por *strings* de conceitos na base de dados das terminologias usadas alcançou uma precisão de 97%, no melhor dos casos onde todos os conceitos buscado realmente existiam na base de dados. No pior dos casos, um experimento de porte muito pequeno e menor relevância, o algoritmo alcançou 45% de precisão. O caso com o pior resultado foi obtido com o primeiro experimento realizado, que necessitava de aperfeiçoamento além de ter sido subdimensionado.

Para a segmentação de sentenças o erro estimado é menor que 24%, com significância de 5%. Um acerto de 76% para a segmentação de sentenças é bastante significativo, ainda mais que a porção maior de erros na classificação dos segmentos se dá por superestimar (6,05% dos erros na amostra) ou subestimar (6,05% dos erros na amostra) o tamanho de um segmento, erros estes que podem ser compensados em fases posteriores de processamento.

Para a fase de atribuição de códigos de conceitos à segmentos de texto, obteve-se um erro estimado menor que 25% com significância de 5%, valor considerado satisfatório. Apesar de existirem aplicações como o MedLEE, que possui um desempenho superior (ver [Mendonça 2003] e [Hripcsak 2002]), deve-se considerar que esta ferramenta é escrita para processamento de língua inglesa, além de utilizar um dicionário próprio, o que contribui para o aumento do desempenho.

A comparação entre uma ferramenta escrita para processamento de língua portuguesa com outra escrita para língua inglesa é complexa, pois a simples comparação de números como recuperação e precisão, não reflete as diferenças entre a quantidade de material existente para o Inglês e não existente para o Português. O simples fato de uma ferramenta ou processo possuir alguns anos a mais de desenvolvimento, onde *Corpus* e dicionários puderam

ser construídos, resulta em uma grande vantagem por parte da ferramenta mais antiga, quando comparada com a mais recente.

Com relação aos sistemas de ontologias adotados, estes se demonstraram ainda imaturos para o uso em língua portuguesa, porém como são sistemas em constante aperfeiçoamento, espera-se que em algum momento estes sistemas ofereçam pelo menos um conteúdo mínimo para utilização em aplicações de PLN voltadas a medicina e em outras línguas além da inglesa. É importante ressaltar que através do desenvolvimento de pesquisas e publicações que envolvam as ontologias utilizadas, espera-se contribuir com discussões que venham a aperfeiçoar estas ontologias. Ressalta-se também como um resultado dos trabalhos com o RadLex, a utilização da estrutura criada sobre a ferramenta Protégé, que pode ser utilizada em um outro trabalho de pós-graduação, que gerou uma publicação no congresso anual promovido pela [RSNA] (ver [Castilla 2006]), a entidade responsável pelo RadLex.

Apesar de não ter sido possível concluir a última fase de processamento dos laudos para obtenção do laudo anotado com informações semânticas, com os resultados obtidos o alcance da fase final de processamento é plausível, ficando assim a implementação da última fase de processamento para trabalhos futuros.

Alguns pontos que podem trazer melhorias ao trabalho desenvolvido, dizem respeito a adoção de correção gramatical automática, adição de informação sintática a estrutura das ontologias usadas e a criação de um conjunto de *Corpus* que sirvam como referência para a avaliação dos algoritmos implementados. A adição de informação sintática às ontologias pode ser útil para remover ambigüidades de conceitos e melhorar o algoritmo de busca por *strings* na base de dados de terminologias.

As contribuições desta tese foram:

- ✓ Criação de um *Corpus* de laudos de Raio-X de Tórax em português anotado sintaticamente para treinamento de algoritmos. Dependendo apenas da verificação de questões legais, este *Corpus* poderia ser disponibilizado para a comunidade científica;
- ✓ Desenvolvimento de um método para a avaliação de Ontologias segundo uma área de conhecimento predefinida;

- ✓ Criação de ferramentas de extração de dados e de uma estrutura de dados para armazenamento e acesso do conteúdo de um sistema de ontologias, considerando as necessidades de ferramentas voltadas ao PLN e um ambiente multilíngue. No caso as ontologias trabalhadas foram o RadLex e o UMLS;
- ✓ Construção de uma aplicação de PLN baseada em ferramentas padrão, disponíveis na *Internet*, complementada pela criação de ferramentas próprias. A saber: algoritmo de busca por conceitos na base de terminologias, algoritmo de segmentação de sentenças e algoritmo de rotulação de segmentos segundo conceitos médicos.

Espera-se que os resultados obtidos possam se juntar aos esforços na criação de ferramentas PLN para a língua portuguesa, como os projetos Linguatca [Linguatca] ou o *Corpus* anotado do CETENFolha, voltados a PLN em português de Portugal e do Brasil.

Como trabalhos futuros resultantes desta tese, deixa-se a proposta de:

- ✓ Importar a nova versão do RadLex para a estrutura de dados utilizada neste trabalho e com uma versão traduzida para o português;
- ✓ Completar a implementação da anotação de laudos com a implementação da última fase de processamento de laudos;
- ✓ Dependendo da viabilidade legal, disponibilizar o material organizado para esta tese para *download* na *Internet*. O que inclui, o *Corpus* de laudos e futuramente a versão portuguesa do RadLex; As ferramentas desenvolvidas também podem ser disponibilizadas;
- ✓ Criação de um *Corpus* anotados com informações sobre sintagmas nominais ou com a rotulação de conceitos, que possam vir a servir como referência para avaliação e treinamento de aplicativos para grupos que desenvolvam pesquisa em PLN.

Concluindo, as dificuldades encontradas acabaram por servir como incentivo para que esta tese não seja apenas um conjunto de ferramentas de anotação de laudos mas também uma descrição de todo o processo de desenvolvimento para facilitar a criação de novas aplicações ou a expansão do trabalho para novas áreas médicas.

9 REFERÊNCIAS BIBLIOGRÁFICAS

- [**Bolshakov 2004**] I. A. Bolshakov, A. Gelbukh, Computational Linguistics: Models, Resources, Applications, IPN-UNAM-FCE, First Edition, 2004.
<http://www.gelbukh.com/clbook/> (Ultimo acesso 07/01/2007)
- [**Brachman 2004**] R. J. Brachman, H. J. Levesque, Knowledge Representation and Reasoning, Morgan Kaufman, 2004.
- [**Brill 1992**] E. Brill, A Simple Rule-Based Part of Speech Tagger, Proceedings of the Third Annual Conference on Applied Natural Language Processing, ACL, 1992.
- [**Brill 1999**] E. Brill, Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging, Natural Language Processing Using Very Large Corpora, Springer, January 1999.
- [**Brown 1979**] W. N. Francis, H. Kucera, Brown Corpus Manual, Manual of Information to accompany a Standard Corpus of Present-Day Edited American English, for use with Digital Computers, Revised and Amplified 1979.
- [**Castilla 2005**] A. Castilla, A. Bacic, S. Furuie, Machine Translation on the Medical Domain: The Role of BLEU/NIST and METEOR in a Controlled Vocabulary Setting, MT Summit X, September 12-16 2005.
- [**Chen 1994**] Kuang-hua Chen, Hsin-Hsi Chen, Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic Evaluation, Proceedings of the 32nd ACL Annual Meeting, 1994, pp. 234-241.
- [**Chonsky 1956**] N. Chonsky, Three Models for The Description of Language, IEEE Transactions on Information Theory, Vol. 2, Issue 3, Sept. 1956.
- [**Chu 2000**] W. W. Chu, D. B. Johnson, H. Kangarloo, A Medical Digital Library to Support Scenario and User-Tailored Information Retrieval, IEEE Transactions on Information Technology in Biomedicine, Vol. 4, No2, Jun. 2000.
- [**Cowie 1996**] J. Cowie, W. Lehnert, Information Extraction, Communications of the ACM, Vol. 39, No 1, Jan. 1996.
- [**Farquhar 1996**] A. Farquhar, R. Fikes, J. Rice, The Ontolingua Server: a Tool for Collaborative Ontology Construction, KAW96. Nov. 1996.
- [**Farquhar 1997**] A. Farquhar, Ontolingua Tutorial, Knowledge Systems Lab, Stanford University, 1997.
- [**Friedman 1998**] C. Friedman, G. Hripcsak, Evaluating Natural Language Processors in the Clinical Domain, Methods of Information in Medicine 1998, Vol. 37 (4-5).
- [**Hobbs 1996**] J. R. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, M. Tyson, A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text, 1996.
<http://www.isi.edu/~hobbs/> (Ultimo acesso: 20/05/2007)
- [**Hripcsak 2002**] G. Hripcsak, J. H. M. Austin, P. O. Alderson, C. Friedman, Use of Natural Language Processing to Translate Clinical Information from a Database of 889,921 Chest Radiographic Reports, Radiology, Vol. 224, No 1, July 2002.
- [**Jackson 2002**] P. Jackson, I. Moulinier, Natural Language Processing for On Line Applications – Text Retrieval. Extraction and Categorization, John Benjamins Publishing Company, 2002.
- [**Jurafsky 2000**] D. Jurafsky, J. M. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice Hall, 2000.
- [**Langlotz 2002**] C. P. Langlotz, S. A. Caldwell, The completeness of Existing Lexicons for Representing Radiology Report Information, Journal of Digital Imaging, Vol 15, Suppl 1, 2002.
- [**Manning 2003**] C. D. Manning, H. Schütze, Foundations of Statistical Natural Language Processing, The MIT Press, Sixth Printing, 2003.
- [**McClave 2000**] J. T. McClave, T. Sincich, Statistics, Prentice Hall, 2000, 8th Edition
- [**Mendonça 2000**] E. A. Mendonça, Using Extraction from the Medical Record to Access Biomedical Literature, PhD Thesis for Graduate School of Arts and Sciences of Columbia University, 2002.

[Mendonça 2003] E. A. Mendonça, J. Haas, L. Shagina, E. Larson, Carol Firedman, Extracting Information on Pneumonia in Infants Using Natural Language Processing of Radiology Reports, Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine, Vol. 13, Sapporo, Japan, 2003.

[Minsky 1974] Marvin Minsky, A Framework for Representing Knowledge, MIT-AI Laboratory Memo 306, June, 1974.

<http://web.media.mit.edu/~minsky/papers/Frames/frames.html> (Último acesso: 28/2/2006)

[Miorelli 2001] S. T. Miorelli, ED-CER: Extração do Sintagma Nominal em Sentenças em Português, Dissertação apresentada ao Programa de Pós Graduação em Ciência da Computação, Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2001/JULHO

[MUCHMORE 2001] MUCHMORE - Multilingual Concept Hierarchies for Medical Information Organization and Retrieval, "State of the Art in Cross-lingual Information Access for Medical Information", January 2001.

<http://muchmore.dfki.de/deliverables.htm> (Último acesso: 20/05/2007)

[NLTK 2004] S. Bird, E. Loper, NLTK Tutorials, 2004.

[Noy 2001] N. F. Noy, D. L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology, Stanford University, Stanford, 2001

<http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html>

(Último acesso: 08/02/2007)

[Papineni 2002] K. Papineni, S. Roukos, T. Ward, W. J. Zhu, BLEU: a Method for Automatic Evaluation of Machine Translation, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.

[PEP 2003] E. Massad, H. F. Marin, R. S. Azevedo N., O Prontuário Eletrônico Do Paciente Na Assistência, Informação e Conhecimento Médico, OPAS/OMS 2003.

[Perini 2005] M. A. Perini, Gramática Descritiva do Português, Editora Ática, 4ª Edição, 8ª Impressão, 2005

[Porter 1980] M.F.Porter, An algorithm for suffix stripping, Program, 14, no. 3, pp 130-137, July 1980.

<http://www.tartarus.org/~martin/PorterStemmer/def.txt> (Último acesso 17/06/2006)

[Porter 2001] M.F. Porter, Snowball: A language for stemming algorithms, October 2001,

<http://snowball.tartarus.org> (Último acesso 20/05/2007).

[RadLex 2003] RadLex: Overview of Lexicon Organization, Radiological Society of North America, 2003.

[Randall 1993] D. Randall, H. Shrobe, P. Szolovits, What is Knowledge Representation?, AI Magazine, 14 (1), 17-33, 1993.

[Rangayyan 2005] R. M. Rangayyan, The Biomedical Engineering Series: Biomedical Imaging Analysis, CRC Press, 2005.

[UMLS LEXICON 2000] A. C. Browne, A. T. McCray, S. Srinivasan, The SPECIALIST LEXICON, Lister Hill National Center for Biomedical Communications, National Library of Medicine, Bethesda, Maryland, REVISED: JUNE 2000.

[UMLS 2004AB] National Library of Medicine, "Unified Medical Language System Documentation", 15th Edition, July Release, 2004AB DOCUMENTATION

[Weizenbaun 1966] J. Weizenbaun, ELIZA – A Computer Program for the Study of Natural Language Communication Between Man and Machine, Communications of the ACM, Vol. 9, No 1, Jan. 1966.

[Yates 1999] R. Y. Yates, B. Ribeiro N., Modern Information Retrieval, Addison-Wesley Pub. Co., 1st Edition, 1999.

10 ENDEREÇOS ELETRÔNICOS

[BVS] Biblioteca Virtual em Saúde, <http://www.bireme.br> (Último acesso em 20/05/2007)

[CETENFolha] Corpus de Extractos de Textos Electrónicos NILC / Folha de S. Paulo
<http://acdc.linguateca.pt/cetenfolha/> (Último acesso em 20/05/2007)

[DECS] Descritores de Ciências e Saúde, <http://decs.bvs.br> (Último acesso em 20/05/2007)

[Dic. Médico HCFMUSP] Dicionário Médico para o Público: <http://www.hcnet.usp.br/dicionario/index.htm>
(Último acesso em 20/05/2007)

[ICD] History of the development of the ICD, <http://www.who.int/classifications/icd/en/HistoryOfICD.pdf>
(Último acesso em 20/05/2007)

[LILACS] Metodologia LILACS, <http://bvsmodelo.bvsalud.org/site/lilacs/> (Último acesso em 20/05/2007)

[Linguateca] Linguateca, <http://www.linguateca.pt/> (Último acesso em 20/05/2007)

[MedLEE] A Medical Language Extraction and Encoding System, <http://lucid.cpmc.columbia.edu/medlee/>
(Último acesso em 20/05/2007)

[MESH] Medical Subject Headings, <http://www.nlm.nih.gov/mesh/> (Último acesso em 20/05/2007)

[NLM] National Library of Medicine: <http://www.nlm.org/> (Último acesso em 20/05/2007)

[NLTK] Natural Language Toolkit, <http://sourceforge.net/projects/nltk> (Último acesso em 20/05/2007)

[Protégé] Protege user documentation, Stanford University, Stanford Medical Informatics.
<http://protege.stanford.edu/doc/users.html> (Último acesso 20/05/2007).

[PyStemmer] PyStemmer: <http://sourceforge.net/projects/pystemmer/> (Último acesso em 20/05/2007)

[RadLex] A Lexicon for Uniform Indexing and Retrieval of Radiology Information Resources,
<http://www.rsna.org/RadLex/> (Último acesso em 20/05/2007)

[ROC] The Receiver Operating Characteristic (ROC) Curve, Cita: Metz CE, Basic principles of ROC analysis, Sem. Nuc. Med. 8 283-298 (1978), <http://www.mlahanas.de/MOEA/Med/ROC21.htm> (Último acesso em 20/05/2007)

[SNOMED CT] SNOMED Clinical Terms, <http://www.snomed.org/> (Último acesso em 20/05/2007)

[Snowball] Snowball Stemmer: <http://www.snowball.tartarus.org> (Último acesso em 20/05/2007)

[Systran] Systran – Language Translation Technologies, <http://www.systransoft.com/index.html> (Último acesso em 20/05/2007)

[UMLS] Unified Medical Language System, <http://www.nlm.nih.gov/research/umls/> (Último acesso em 20/05/2007)

11 ANEXOS

11.1 Constantes e Operadores para Definição de Expressões Regulares Segundo o Padrão Posix.

Exemplo 27: Operações e constantes usadas em expressões regulares segundo o padrão Posix

Reconhecimento de caracteres	
.	Reconhece qualquer caractere.
[...]	Reconhece qualquer caractere listado entre os colchetes.
[^]	Reconhece qualquer caractere exceto os listados entre os colchetes.
Qualificadores	
?	Reconhece qualquer caractere zero ou uma vez.
*	Reconhece qualquer caractere zero ou mais vezes.
+	Reconhece qualquer caractere uma ou mais vezes.
{ <i>n</i> }	Reconhece o elemento precedente <i>n</i> vezes.
{ <i>min</i> , <i>max</i> }	Reconhece o elemento precedente no mínimo <i>min</i> vezes e no máximo <i>max</i> vezes.
Âncoras	
^	Faz o reconhecimento no começo da linha.
\$	Faz o reconhecimento no fim da linha.
\<	Faz o reconhecimento no começo da palavra.
\>	Faz o reconhecimento no fim da palavra.
\b	Faz o reconhecimento no começo ou no fim da palavra.
\B	Faz o reconhecimento de qualquer caractere nem no começo nem no fim da palavra.
Caracteres de escape	
\	Reverte o significado de um caractere especial para seu literal a ser reconhecido.
Alternando	
	Faz a operação <i>OU</i> lógica entre expressões regulares.
Escopos	
()	Os parênteses são usados para delimitar o escopo de uma expressão.
Referências prévias (Back references)	
(<i>expressão</i>) \1	Um padrão pode ser armazenado em buffer para posterior uso através de back references. O \1 passa a armazenar o resultado do reconhecimento obtido com a expressão entre parênteses.

Exemplificando uma expressão regular com a notação fornecida. Suponhamos que seja escrito um programa para buscar por todos os e-mails cujo destinatário ou remetente seja o joao.silva@email.com. A expressão regular teria a seguinte forma:

```
(^To | ^From):( joao.silva@email.com)
```

Segundo a expressão acima, devem ser reconhecidas as palavras *To* ou *From*, no início de uma linha, seguidas pelo caractere ':' e pelo endereço de e-mail procurado. É interessante lembrar que o reconhecimento faz diferença para letras maiúsculas ou minúsculas.

11.2 Hierarquia de Chomsky

Na década de 50 Noam Chomsky definiu uma hierarquia para as **gramáticas formais** que geram **linguagens formais** (ver [Chomsky 1956]). Essa hierarquia ficou conhecida como “Hierarquia de Chomsky”. Por se tratar de um estudo sobre as gramáticas que geram linguagens formais, a teoria das linguagens formais estuda apenas a forma da linguagem, mas não sua interpretação. Segundo a Hierarquia de Chomsky temos quatro tipos e linguagens formais, portanto gramáticas. A saber:

Tabela 49: Hierarquia de Chomsky⁵

Classe	Tipo	Descrição
3	Regular	As gramáticas regulares podem ser descritas por gramáticas regulares, que por sua vez podem ser descritas por expressões regulares e representadas através de autômatos finitos.
2	Livre de contexto	Gramáticas que possuem regras de produção na forma $A \rightarrow a$, onde A é um não terminal e a é uma combinação de terminais e não terminais. Não podem ser representadas por expressões regulares, portanto não podem ser representadas por AEFN ou AEFN, mas podem ser representadas por Autômatos Push-Down. Os autômatos Push-Down são semelhantes aos AEFN, porém possuem uma pilha que serve como memória para o reconhecimento de cadeias de entrada.
1	Sensível a contexto	As linguagens sensíveis a contexto permitem a existência de mais de um símbolo ao lado esquerdo da regra de produção, devendo ser obedecidas as seguintes regras: * CS1: O Número de símbolos ao lado esquerdo da expressão não pode exceder o número de símbolos ao lado direito. * CS2: Não são permitidas regras do tipo $A \rightarrow \epsilon$ e a não ser que A seja um símbolo inicial e não ocorra ao lado direito de regra alguma. São reconhecidas por Autômatos Lineares Restritos (Linear-Bounded Automata), que são similares aos AEFN ou AEFN, mas são capazes de armazenar símbolos em uma lista.
0	Irrestrita ou livre	Não há restrições em suas regras de produção, exceto que deve haver pelo menos um não terminal ao lado esquerdo da expressão. Podem ser reconhecidas por Máquinas de Turing.

Segundo a hierarquia dada, quanto mais poderosa uma linguagem, menos restritiva é sua gramática, sendo assim, as classes são geradas segundo a imposição de restrições. Cada classe de gramática engloba as classes mais restritivas.

Identificar a qual classe uma linguagem pertence é importante para que as ferramentas computacionais mais adequadas possam ser escolhidas. Para identificar uma linguagem como

⁵ Extraído de ‘The Chomsky Hierarchy’: <http://www.cs.may.ie/~jpower/Courses/parsing/node21.html> (último acesso em 12/1/2006).

pertencente a uma das classes propostas por Chomsky, é necessário conhecer sua gramática para então determinar a que classe pertence essa gramática.

As linguagens de programação como Java ou Python, por exemplo, podem ser enquadradas como classe 2 (Livre de contexto), o que torna possível a construção de compiladores ou interpretadores.

No caso das linguagens naturais é difícil as categorizar em uma classe, a maioria dos autores trabalham na direção de provar que uma determinada linguagem (alemão ou inglês, por exemplo) não pertence a uma determinada classe. Segundo Jurafsky, as linguagens naturais não são regulares ([Jurafsky 2000], capítulo 13, página 485-488) nem Livres de contexto (capítulo 13, página 488-491).

11.3 Organização dos Arquivos/Tabelas do UMLS

Os dados referentes à Rede Semântica se encontram em uma coleção de arquivos texto cujos campos são separados por vírgula. Estes arquivos podem ser importados para tabelas em um banco de dados relacional. A Tabela 50 descreve as tabelas relacionais da Rede Semântica.

Tabela 50: Tabelas relacionais da Rede Semântica [UMLS 2004AB]

Tabela	Descrição
SRDEF	Informações básicas sobre os Tipos Semânticos e relacionamentos
SRSTR	Estrutura da Rede Semântica
SRSTRE1	Conjunto de heranças dos relacionamentos (UI, Unique Identifier)
SRSTRE2	Conjunto de heranças dos relacionamentos (nomes)
SRFIL	Descrição de cada tabela
SRFLD	Descrição de cada campo em cada tabela.

Além das tabelas específicas da Rede Semântica, no Metathesaurus a tabela MRHIER contém informações sobre os Tipos semânticos e sua hierarquia.

Os trechos de dados a seguir exemplificam a formatação e o conteúdo das tabelas da Rede Semântica.

Tabela 51: Exemplo de dados da tabela SRSTR [UMLS 2004AB]

Argumento 1 do relacionamento	Relacionamento	Argumento 2 do relacionamento	Link Status
Acquired Abnormality	co-occurs_with	Injury or Poisoning	D
Acquired Abnormality	isa	Anatomical Abnormality	D
Acquired Abnormality	result_of	Behavior	D
Activity	isa	Event	D
Age Group	isa	Group	D
Alga	isa	Plant	D

Tabela 52: Exemplo de dados da tabela SRSTR1 [UMLS 2004AB]

Argumento 1	Relacionamento	Argumento 2
T020	T186	T190
T020	T186	T017
T020	T186	T072
T052	T186	T051
T052	T165	T090

Tabela 53: Exemplo de dados da tabela SRSTR2 [UMLS 2004AB]

Argumento 1	Relacionamento	Argumento 2
Acquired Abnormality	isa	Anatomical Abnormality
Acquired Abnormality	isa	Anatomical Structure
Acquired Abnormality	isa	Physical Object
Acquired Abnormality	isa	Entity
Acquired Abnormality	affects	Alga
Acquired Abnormality	affects	Amphibian
Acquired Abnormality	affects	Animal
Acquired Abnormality	affects	Bacterium
Acquired Abnormality	affects	Bird
Activity	isa	Event
Age Group	isa	Group
Age Group	isa	Conceptual Entity
Age Group	isa	Entity
Alga	isa	Plant
Alga	isa	Organism
Alga	isa	Physical Object
Alga	isa	Entity

Tabela 54: Exemplo de dados da tabela SRDEF [UMLS 2004AB]

Tipo de registro (sty=Semantic Type, RL=Relation)	UI (Unique Identifier do tipo semântico ou relação)	Nome do STY ou RL	Índice na árvore do STY ou RL	Definição do STY ou RL	Exemplos no Metathesaurus	Notas de uso apenas para o STY	Non-Human Flag (somente STY)	Abreviação	Relação inversa (somente RL)
STY	T020	Acquired Abnormality	A1.2.2.2	An abnormal structure, or one that is abnormal in size or location, found in or deriving from a previously normal structure. Acquired abnormalities are distinguished from diseases even though they may result in pathological functioning (e.g., "hernias incarcerate").	Abscess of prostate; Hemorrhoids; Hernia, Femoral; Varicose Veins				
STY	T052	Activity	B1	An operation or series of operations that an organism or machine carries out or participates in.	Social Planning; Expeditions; Information Distribution; Return Migration	Few concepts will be assigned to this broad type. Wherever possible, one of the more specific types from this hierarchy will be chosen. For concepts assigned to this type, the focus of interest is on the activity. When the focus of interest is the individual or group that is carrying out the activity, then a type from the 'Behavior' hierarchy will be chosen. In general, concepts will not receive a type from both the 'Activity' and the 'Behavior' hierarchies.			

Para maiores informações sobre os campos e a formatação dos dados consultar a documentação do UMLS [UMLS 2004AB].

11.4 Conjunto de Rótulos Adaptados do *Brown Corpus*

Tabela 55: Rótulos do *Brown Corpus* usados para o treinamento do *Brill Tagger*

Rótulo	Descrição	Exemplo
. (ponto final)	Fim de sentença	. ; ? !
(	Abre parentesis	
)	Fecha parentesis	
*	Não, sem	
,	Vírgula	
:	Dois pontos	
ABL	Pré-qualificador	discretamente
ABX	Pré-quantificador	ambas
AT	Artigo	a, o
CC	Conjunção coordenada	E / ou
CD	Número cardinal	1, 2, 3, ...
CS	Conjunção subordinada	Se
DT	Determinante no singular	Este, esta
DTI	Determinante (qualificador/quantificador)	Algum
DTS	Determinante no plural	Alguns
IN	Preposição	A
JJ	Adjetivo	Bonito
JJS	Adjetivo no plural	Bonitos
NN	Substantivo no singular	Gato
NNS	Substantivo no plural	Gatos
OD	Número ordinal	1º, 2º, ...
PP\$	Pronome possessivo	Meu, seu, ...
PPL	Reflexivo (singular)	-se
PPLS	Reflexivo (plural)	-nos
RB	Advérbio	
VB	Verbo	
VBD	Verbo no passado	
VBG	Verbo no presente, particípio ou gerúndio	
VBN	Verbo no particípio passado	
VBZ	Verbo na 3ª pessoa do singular, presente	

11.5 Publicações Resultantes da Tese

[1] A. S. Bacic, S. S. Furuie, Sharing and Archiving Medical Knowledge Through Collaborative Tools, Medical Imaging 2004: PACS and Imaging Informatics, edited by Osman M. Ratib, H. K. Huang, Proceedings of SPIE Vol. 5371 (SPIE, Bellingham, WA, 2004)

[2] A. Castilla, A. Bacic, S. Furuie, "Machine Translation on the Medical Domain: The Role of BLEU/NIST and METEOR in a Controlled Vocabulary Setting", MT Summit X, September 12-16 2005, Phuket - Thailand

[3] Castilla A. C., Bacic A. S., Furuie S. S., Performing Logical Reasoning on a RadLex Ontology. Oral Scientific Presentation - Abstract published on the conference proceedings 92nd Radiological Society of North America Scientific Assembly and Annual Meeting. 27/11/2006. Chicago, USA.

11.6 Exemplos de Laudos Rotulados

Original:

Estruturas ósseas e tecidos moles da parede torácica, sem alterações. Assimetria pleural arterial à direita. Vascularização pulmonar normal. Hilos de configuração, topografia e dimensões normais. Imagem cardíaca normal. Aorta alongada. Arco médio normal. Parênquima pulmonar de transparência normal. Mediastino sem alterações.

Anotado:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<LAUDO>
  <SENT ORIG="Estruturas ósseas e tecidos moles da parede torácica, sem
    alterações.">
    <TERM ID="C0026367" SRC="umls" SEARCH="moles/JJS"
      FOUND="molar" SCORE="1.0"></TERM>
    <TERM ID="RL414" SRC="radlex" SEARCH="parede/NN torácica/JJ"
      FOUND="parede torácica" SCORE="1.0">
    <TERM ID="C0004757" SRC="umls" SEARCH="torácica/JJ"
      FOUND="torácicos" SCORE="1.0"></TERM>
    </TERM>
  </SENT>
  <SENT ORIG="Assimetria pleural arterial à direita."></SENT>
  <SENT ORIG="Vascularização pulmonar normal.">
    <TERM ID="C0697197" SRC="umls" SEARCH="pulmonar/JJ"
      FOUND="pulmonaria" SCORE="1.0"></TERM>
  </SENT>
  <SENT ORIG="Hilos de configuração, topografia e dimensões normais."></SENT>
  <SENT ORIG="Imagem cardíaca normal."></SENT>
  <SENT ORIG="Aorta alongada."></SENT>
  <SENT ORIG="Arco médio normal."></SENT>
  <SENT ORIG="Parênquima pulmonar de transparência normal .">
    <TERM ID="C0697197" SRC="umls" SEARCH="pulmonar/JJ"
      FOUND="pulmonaria" SCORE="1.0"></TERM>
    <TERM ID="C0004282" SRC="umls" SEARCH="transparência/NN"
      FOUND="transparências" SCORE="1.0"></TERM>
  </SENT>
  <SENT ORIG="Mediastino sem alterações ."></SENT>
</LAUDO>
```

Original:

Escoliose direita e espondilose torácica. Opacidade linear no campo médio esquerdo. Hilos proeminentes. Aorta alongada.

Anotado:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<LAUDO>
  <SENT ORIG="Escoliose direita e espondilose torácica.">
    <TERM ID="C1261380" SRC="umls" SEARCH="espondilose/NN torácica/JJ"
      FOUND="espondilose torácica" SCORE="1.0">
    <TERM ID="C0004757" SRC="umls" SEARCH="torácica/JJ"
      FOUND="torácicos" SCORE="1.0"></TERM>
    </TERM>
  </SENT>
  <SENT ORIG="Opacidade linear no campo médio esquerdo.">
    <TERM ID="C1010570" SRC="umls" SEARCH="linear/JJ"
      FOUND="linaria" SCORE="1.0"></TERM>
  </SENT>
  <SENT ORIG="Hilos proeminentes."></SENT>
  <SENT ORIG="Aorta alongada."></SENT>
</LAUDO>
```

11.7 Diagrama ER para o Armazenamento dos Laudos

Figura 11: Diagrama ER para o Armazenamento dos Laudos