UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE CIÊNCIA DA INFORMAÇÃO

EDSON MARCHETTI DA SILVA

# RECUPERAÇÃO DA INFORMAÇÃO ATRAVÉS DE BUSCA COMPARADA EM DOMÍNIO ESPECÍFICO, BASEADO EM EXPRESSÕES MULTIPALAVRAS

Belo Horizonte
2013

EDSON MARCHETTI DA SILVA

# RECUPERAÇÃO DA INFORMAÇÃO ATRAVÉS DE BUSCA COMPARADA EM DOMÍNIO ESPECÍFICO, BASEADO EM EXPRESSÕES MULTIPALAVRAS

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

Linha de Pesquisa: Organização e Uso da Informação

Orientador: Prof. Dr. Renato Rocha Souza

BELO HORIZONTE

2013

Silva, Edson Marchetti.

S586r Recuperação da informação através de busca comparada em domínio específico, baseado em expressões multipalavras [manuscrito] / Edson Marchetti Silva. – 2013.
177 f. : Il: color, enc.

Orientador: Renato Rocha Souza.
Tese (doutorado) – Universidade Federal de Minas Gerais, Escola de Ciência da Informação.
Referências: f. 152-155
Apêndices: f. 156-177

1. Ciência da informação – Teses. 2. Sistemas de recuperação da informação – Teses. 3. Ferramentas de busca – Teses. 4. Linguagem documentaria – Teses. I. Título. II. Souza, Renato Rocha. III. Universidade Federal de Minas Gerais, Escola de Ciência da Informação.

CDU: 025.4.03

Ficha catalográfica: Biblioteca Profª Etelvina Lima, Escola de Ciência da Informação da UFMG

UFMG

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

**FOLHA DE APROVAÇÃO**

"RECUPERAÇÃO DA INFORMAÇÃO ATRAVÉS DE BUSCA COMPARADA EM DOMÍNIO ESPECÍFICO, BASEADO EM EXPRESSÕES MULTIPALAVRAS"

Edson Marchetti da Silva

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

Tese aprovada em: 25 de abril de 2013.

Por:

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

Prof. Dr. Manoel Palhares Moreira - PUC/MG

Prof. Dr. Luiz Cláudio Gomes Maia - FUMEC

Profa. Dra. Lídia Alvarenga - ECI/UFMG

Prof. Dr. Maurício Barcellos Almeida - ECI/UFMG

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

Aprovada pelo Colegiado do PPGCI

Profa. Renata Maria Abrantes Baracho Porto
Coordenadora

Versão final Aprovada por

Prof. Renato Rocha Souza
Orientador

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

UFMG

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

ATA DA DEFESA DE TESE DE **EDSON MARCHETTI DA SILVA**, matrícula: 2010654905

Às 10:00 horas do dia 25 de abril de 2013, reuniu-se na Escola de Ciência da Informação da UFMG a Comissão Examinadora aprovada *ad referendum* pela Coordenadora do Programa de Pós-Graduação em Ciência da Informação em 08/03/2013, para julgar, em exame final, o trabalho intitulado ***Recuperação da informação através de busca comparada em domínio específico, baseado em expressões multipalavras***, requisito final para obtenção do Grau de DOUTOR em CIÊNCIA DA INFORMAÇÃO, área de concentração: Produção, Organização e Utilização da Informação, Linha de Pesquisa: Organização e Uso da Informação. Abrindo a sessão, o Presidente da Comissão, Prof. Dr. Renato Rocha Souza, após dar conhecimento aos presentes do teor das Normas Regulamentares do Trabalho Final, passou a palavra ao candidato para apresentação de seu trabalho. Seguiu-se a argüição pelos examinadores com a respectiva defesa do candidato. Logo após, a Comissão se reuniu sem a presença do candidato e do público, para julgamento e expedição do resultado final. Foram atribuídas as seguintes indicações:

| | |
|---|---|
| Prof. Dr. Renato Rocha Souza - Orientador | APROVADO |
| Prof. Dr. Manoel Palhares Moreira | APROVADO |
| Prof. Dr. Luiz Cláudio Gomes Maia | APROVADO |
| Profa. Dra. Lídia Alvarenga | APROVADO |
| Prof. Dr. Maurício Barcellos Almeida | APROVADO |
| Profa. Dra. Renata Maria Abrantes Baracho Porto | APROVADO |

Pelas indicações, o candidato foi considerado APROVADO.

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

Belo Horizonte, 25 de abril de 2013

Prof. Dr. Renato Rocha Souza
FGV/RJ

Prof. Dr. Manoel Palhares Moreira
PUC/MG

Prof. Dr. Luiz Cláudio Gomes Maia
FUMEC

Profa. Dra. Lídia Alvarenga
ECI/UFMG

Prof. Dr. Maurício Barcellos Almeida
ECI/UFMG

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

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

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

# DEDICATÓRIA

A meu pai Synésio (*in memorian*), que nos deixou durante o período do meu doutoramento, e sempre me serviu de inspiração. Ele foi um homem sábio, com quem aprendi que apenas através do esforço e determinação conquistamos nossos objetivos.

À minha esposa Márcia e minhas filhas Cínara e Maiara, que tiveram de conviver com a minha ausência, não apenas durante o período de doutorado, mas durante todo esse percurso por quase uma década para chegar até aqui.

À minha mãe Ephigênia que sempre acreditou em mim.

À minha irmã Eliane que me incentivou, desde o início desse caminho, e agora, sou eu que a incentivo para buscar essa mesma meta.

Ao meu irmão Eduardo que acabou de se graduar como engenheiro civil.

À minha irmã Elizabete (*in memorian*) que também nos deixou durante esse percurso.

A todos demais familiares e amigos.

## AGRADECIMENTOS

Em primeiro lugar, gostaria de agradecer de forma bem especial o meu orientador prof. Dr. Renato Rocha Souza, pois sem o apoio dele nada disso teria sido possível. Sei o quanto ele foi importante, apoiando-me e incentivando-me desde o início. Pelo seu esforço e dedicação, pelos e-mails enviados após a meia noite e todo o trabalho com as infindadas correções dos textos publicados.

À Universidade Federal de Minas Gerais (UFMG) da qual tive todo o apoio como bolsista CAPES-Reuni e como bolsista PDSE-CAPES, o que possibilitou o meu estágio de doutoramento na Universidade de Strathclyde em Glasgow no Reino Unido.

Também agradecer à ex-coordenadora da Pós-Graduação Profa. Dra. Gercina Ângela Borém de Oliveira Lima e à atual profa. Renata Maria Abrantes Baracho Porto e a todos os demais professores dos quais tive a oportunidades de receber seus ensinamentos.

Aos meus colegas do PPGCI, e também aos funcionários, em especial à secretária Nely que sempre se prontificou a me ajudar a organizar toda documentação necessária.

Ao prof. Dr. Dmitri Roussinov, meu co-orientador estrangeiro na universidade de Strathclyde em Glasgow, o qual me acolheu de forma generosa e muito proveitosa para o meu processo de formação.

Ao Centro Federal de Educação Tecnológica de Minas Gerais (CEFET), que me concedeu redução da carga horária para viabilizar o meu processo de aprendizado com o de ensino.

E principalmente aos meus colegas da coordenação de informática que precisaram arcar com o ônus da minha ausência.

“*In the middle of difficulty lies opportunity*”

*Albert Einstein*

## RESUMO

Normalmente, as ferramentas de busca em bases de dados utilizam-se de palavras-chave fornecidas pelo usuário para realizar a identificação de documentos. Este trabalho visa propor uma alternativa adicional que possa ser agregada aos Sistemas de Recuperação da Informação (SRI) para auxiliar o usuário em seu processo de busca pela informação em base de documentos. Essa alternativa possibilita a realização de uma busca automatizada baseada em um documento fornecido pelo usuário que serve de referência. Nesse contexto, delimitou-se como objeto de estudo a extração de expressões multipalavras (EM) do documento para servir como descritores da busca em um *corpus* específico. As EM são obtidas através de um método determinístico proposto que considera as características da estrutura física do documento e compara o resultado com o obtido por treze diferentes medidas de associação estatísticas produzidas pelo *software* Ngram Statistics Package (NSP) que considera o texto como um conjunto de palavras do inglês *bag of words*. Os resultados demonstram que o método proposto possibilita uma melhor representação semântica do documento trazendo ganhos qualitativos no conjunto de EM extraídas e que contribui positivamente para o resultado da Busca Comparada. A partir desses experimentos propôs-se e implementou-se um protótipo de ferramenta de Busca Comparada e apresentam-se os resultados obtidos com o seu uso.



**Palavras-chave:** extração de expressões multipalavras, recuperação da informação, medidas de associação estatísticas, busca comparada.

## ABSTRACT

Normally, the search engines in databases is performed using keywords provided by the user to perform the documents identification. This study aims to propose an additional alternative that can be aggregated to Information Retrieval Systems (IRS) to assist the user in the process of information search. This alternative allows the realization of an automated search based on a document supplied by the user which serves as a reference. In this context the object of study was the extraction of Multi Word Expressions (MWE) of the document to serve as descriptors of the search in a specific corpus. The MWE are obtained by a deterministic method which proposed that considers the characteristics of the physical structure of the document and compares the result with that obtained for thirteen different measures of association statistics produced by Statistics Ngram Package (NSP), which considers the text as a set of bag of words. The results demonstrate that the proposed method provides a better semantic representation of the document bringing together qualitative gains in MWE extracted and that it contributes positively to the results of the search compared. From these experiments we have proposed and implemented a prototype of a compared search tool and it was present the results obtained with its use.



**Key-words:** multi word expression extraction, information retrieval, measures of association statistics, compared search

# LISTA DE FIGURAS

# LISTA DE TABELAS

# LISTA DE GRÁFICOS

# LISTA DE ABREVIATURAS

| | | |
|---|---|---|
| ANCIB | – | **A**ssociação **N**acional de Pesquisa e Pós-graduação em **C**iência da **I**nformação |
| CC | – | **C**iência da **C**omputação |
| CE | – | **C**oeficiente de **E**strutura |
| CGI | – | **C**ommom **G**ateway **I**nterface |
| CI | – | **C**iência da **I**nformação |
| CSLI | – | **C**entre for the **S**tudy of **L**anguage of **I**nformation |
| CSV | – | **C**osine **S**imilarity **V**ector |
| DCG | – | **D**efinite **C**lause **G**ramma |
| DLL | – | **D**inamic **L**ink **L**ibrary |
| EM | – | **E**xpressões **M**ultipalavras |
| EMICO | – | **E**nhanced **M**utual **I**nformation and **C**ollocation **O**ptimization |
| ENANCIB | – | **E**ncontro **N**acional **ANCIB** |
| GNU | – | **G**eneral **P**ublic **L**icense |
| GT | – | **G**rupos **T**emáticos |
| IA | – | **I**nteligência **A**rtificial |
| IP | – | **I**nternet **P**rotocol |
| HTTP | – | **H**iper**T**ext **T**ransfer **P**rotocol |
| NSP | – | **N**grams **S**tatistics **P**ackage |
| PDF | – | **P**ortable **D**ocument **F**ormat |
| PDF-TET | – | **P**ortable **D**ocument **F**ormat – **T**ext **E**xtraction **T**oolkit |
| PLN | – | **P**rocessamento de **L**inguagem **N**atural |
| POS | – | **P**arts of **S**peech |
| RI | – | **R**ecuperação da **I**nformação |
| SGBD | – | **S**istemas **G**erenciadores de **B**anco de **D**ados |
| SRI | – | **S**istemas de **R**ecuperação da **I**nformação |
| SQL | – | **S**tructured **Q**uery **L**anguage |
| TCC | – | **T**ermo **T**écnico-**C**ientífico |
| TCP-IP | – | **T**ransmission **C**ontrol **P**rotocol – **I**nternet **P**rotocol |
| TF-IDF | – | **T**erm **F**requency – **I**nverse **D**ocument **F**requency |
| TM | – | **T**ermo **M**ultipalavras |
| URL | – | **U**niform **R**esource **L**ocator |

**SUMÁRIO**

# 1 INTRODUÇÃO

Desde que surgiram os primeiros computadores, um de seus principais propósitos tem sido o de coletar, armazenar e processar grandes volumes de dados a fim de produzir informações. Cabe aos sistemas computadorizados receber esses dados, organizá-los e classificá-los, de tal forma que possam ser recuperados e apresentados ao usuário requisitante a fim de suprir a demanda de informação desejada.

Desde a década de 1960 alguns modelos foram propostos e implementados para gerir o processo de manutenção e recuperação de dados estruturados. Dentre eles, podemos citar o Modelo de Redes, o Modelo Hierárquico, Modelo Relacional e o Modelo Objeto-Relacional. Todos eles demandam que um esquema estrutural seja projetado para receber os dados criando uma forte aderência semântica entre o dado e o exato local onde ele será armazenado, ou seja, o metadado. Nesse tipo de solução, para garantir que a extração das informações seja determinística, os dados necessariamente, precisam ser organizados de forma estruturada e agrupada de acordo com suas características intrínsecas e semânticas.

Portanto, esses modelos são propícios apenas quando lidamos com dados que podem ser organizados de forma estruturada, como é o caso dos atuais sistemas de informações, que armazenam os seus dados apoiados pelas tecnologias disponibilizadas pelos Sistemas Gerenciadores de Banco de Dados Relacionais e suas extensões. Entretanto, a maioria das informações geradas pelo homem não estão na forma estruturada, pois é através da linguagem, principalmente na forma escrita, que elas são registradas. O grande desafio, que ainda apresenta muitas questões em aberto, está em aproximar o computador com a forma humana em lidar com a informação, ou seja, através do tratamento da linguagem natural.

A demanda por uma maior interação entre o homem e o computador se intensificou ao longo das últimas décadas, devido ao processo de popularização dos computadores dado ao crescente aumento da capacidade de armazenamento e de processamento, ao mesmo tempo em que o custo tornou-se cada vez menor. É natural que tais fatores tenham corroborado para popularização dos computadores nas empresas e principalmente no uso doméstico. Mas, talvez o que mais tenha contribuído para essa disseminação foi o incremento de suas funcionalidades através da interconexão possibilitada pelo acesso à internet. A internet surgiu como uma nova mídia de acesso e troca de informação convergindo diversas outras mídias, permitindo a interatividade entre os usuários de uma forma totalmente nova, ampliando as possibilidades de interação. A franca adesão a essa

nova mídia fez crescer exponencialmente a quantidade de dados e informações digitais existentes na grande rede criando novos desafios em como armazenar e recuperar esse crescente volume informacional semi-estruturado e não-estruturado.

A informação disponibilizada em meio digital, em grande parte, se apresenta na forma textual em linguagem natural através de documentos tais como: artigos científicos, teses, livros dentre outros. Mas o computador, ao processar dados expressos em linguagem natural, não tem a capacidade de interpretá-los de forma semântica. Afinal um texto para um computador é uma sequência de bytes em que não há nenhum sentido. Ampliar a capacidade das máquinas para extrair significado de informações semi-estruturadas ou até mesmo não estruturadas é um desafio que vem instigando pesquisadores das mais diversas áreas do conhecimento.

Apesar do interesse comum da Ciência da Informação (CI) e da Ciência da Computação (CC) na informação, a abordagem no trato da mesma é bastante distinta. Para a CI o termo informação está associado à semântica, pois segundo Tálamo (1977), o objeto da CI, a informação, aparece como produto de um processo intencional, como algo construído. Portanto, o propósito é promover a adequação significativa dos conteúdos. Já para a CC, a informação se caracteriza de forma mais abstrata, pois, segundo Setzer (2001, p. 242-243), não é possível processar informação diretamente em um computador. Para isso é necessário reduzi-la a dados. Não obstante às dificuldades, pesquisadores de diversas áreas buscam dominar a complexidade inerente à linguagem, porque é através da linguagem escrita ou falada que a maior parte das informações são registradas e transmitidas entre os seres humanos. Atribuir significado a esses conteúdos possibilitará expressivos ganhos no processo de recuperação automatizada da informação a partir da semântica intrínseca contida nos documentos.

A busca por construir uma máquina capaz de se comunicar com o homem de forma natural através da linguagem falada ou escrita é algo que a Inteligência Artificial (IA) vem buscando há décadas. A IA é uma área de pesquisa que, segundo Russell & Norvig (2004 p. 3-4), teve sua gênese com John McCarthy em 1956 e que, historicamente, vem trabalhando em duas frentes: a primeira focada em sistemas que pensam e agem como os seres humanos e a segunda focada em sistemas que pensam e agem apenas de forma racional. As pesquisas com o foco na primeira abordagem, mostraram-se muito mais complexas do que pareciam ser. Já a segunda abordagem, que trabalha com a racionalidade, faz o que é certo, considerando os dados que têm, e é, portanto, bem mais exitosa, apesar de limitada, por representar apenas alguns aspectos da natureza humana.

Segundo Manning & Schütze (2003, p. 4-7), duas correntes de pensamento predominaram nos estudos da linguagem. A primeira, dos empiristas, entre as décadas de 1920 e 1960, postulava que a experiência é única, ou senão, pelo menos a principal forma de construção do conhecimento na mente humana. Eles acreditavam que a habilidade cognitiva estava no cérebro e que nenhum aprendizado é possível a partir de uma *tabula rasa,* portanto, o cérebro tinha *a priori* alguma capacidade de associação, reconhecimento de padrão e generalização, que aliada à rica capacidade sensorial humana possibilitavam o aprendizado da linguagem. A segunda, dos racionalistas, entre os anos de 1960 e 1985, postulava que significante parte do conhecimento da mente humana não é derivado dos sentidos, mas estabelecido previamente, presumivelmente por herança genética. Essa corrente de pensamento se baseou na teoria da faculdade inata da linguagem proposta por Noam Chomsky, a qual considera as estruturas iniciais do cérebro como responsáveis por fazerem com que cada indivíduo a partir de sua percepção sensorial siga certos caminhos e formas para organizar e generalizar as informações internamente.

Atualmente buscam-se a partir das mais diversas áreas do conhecimento avanços na capacidade das máquinas em representar e recuperar as informações. Nessa busca, um dos principais aspectos é desenvolver a capacidade de interpretação de documentos atribuindo valor semântico ao texto escrito. Destaca-se a área da Engenharia da Linguagem e do Processamento de Linguagem Natural (PLN) através de estudos da morfologia, análise sintática e análise semântica e dos processamentos estatísticos que buscam o reconhecimento de padrões probabilísticos, a fim de predizer comportamentos no conteúdo do texto. Uma das possibilidades adotadas pela PLN é tratar o texto através de uma abordagem estatística, a qual tem mostrado bons resultados práticos no aprendizado automatizado e na desambiguação.

Todas essas questões ainda são um campo profícuo para as ciências. Existe uma incessante busca em articular formas para representar o conhecimento nas máquinas a fim de reduzir as diferenças entre o processamento computacional e a capacidade simbólica do pensar humano. Tudo isso nos leva a uma primeira questão: A partir de qual perspectiva deve-se tratar esse tema? Esse é um relevante e complexo debate, travado pelas mais diversas áreas, desde as humanas, sociais e exatas. Nesse sentido, destaca-se a abordagem realista de Smith e Ceusters (2010) uma tentativa de automatizar a representação do conhecimento textual a partir da aplicação da lógica na linguagem. Decorre dessa abordagem um contra senso, pois se a linguagem antecede a lógica, como usar lógica para expressar toda a semântica possibilitada pela linguagem? Como a linguagem não teve sua fundamentação baseada na lógica, mesmo aumentando a capacidade de expressividade da lógica a partir de novos operadores e relações que

possam vir a ser criados, ainda assim será muito pouco provável que se consiga esgotar todas as nuances da linguagem e ter êxito na sua representação pela lógica. A linguagem é simbólica e nem mesmo existe uma equivalência direta entre os signos criados na mente e uma palavra que expresse o seu significado nos diversos idiomas falados pelo mundo. Nesse sentido recorremos ao conceito de linguagem apresentado por Berger & Luckmann.

> A linguagem constrói campos semânticos ou zonas de significação linguisticamente circunscritas. O vocabulário, a gramática e a sintaxe estão engrenadas na organização desses campos semânticos. Assim a linguagem constrói esquemas de classificação para diferenciar os objetos em gênero ou número; formas para realizar enunciados da razão por oposição a enunciados do ser; modos de indicar o grau de intimidade social, etc. (BERGER & LUCKMANN, 2003 p. 61)

A mente humana é uma visão particular de um indivíduo formada pela convivência social constituindo o que vulgarmente chama-se de personalidade, a qual compõe o seu conjunto próprio de crenças e valores. Somando-se a isso existem as relações pessoais e o acúmulo de dados e informações retidos na mente que formam o conhecimento. Nas reflexões humanas para a produção do conhecimento, ou simplesmente para produzir respostas às perguntas e necessidades sociais, a mente não processa todo o conhecimento existente no cérebro. A mente busca, por aproximação, situações similares às vividas anteriormente, produzindo inferências, criando novas relações ou buscando lembranças registradas na memória. Ou seja, é um recorte de um dado momento de um contexto cerebral. Portanto, não há garantia de exatidão nas respostas em qualquer tempo. Já a máquina digital trabalha em um contexto completamente diferente do cérebro humano, o resultado do processamento é exato e repetível. Portanto, a tecnologia atual jamais será capaz de simular a mente humana em sua plenitude. O que se pode buscar é uma aproximação de algumas das capacidades humanas. Conforme o pensamento de Vigotsky é necessário um claro entendimento das relações entre o pensamento e a língua para que se possa compreender como se dá o desenvolvimento intelectual.

> O significado das palavras é só um fenômeno de pensamento na medida em que é encarnado pela fala e só é um fenômeno lingüístico na medida em que se encontra ligado com o pensamento e por este iluminado. É um fenômeno do pensamento verbal ou da fala significante – uma união do pensamento e da linguagem. (VYGOTSKY, 1987, p. 277-278).

Considera-se que, ao direcionar os esforços da ciência na busca de representação semântica do conhecimento para a recuperação de informação através da tentativa de aproximação em simular a mente humana tal como ela é, talvez não seja o caminho que alcançará melhores resultados, pois, provavelmente, resultará nos mesmos “defeitos”, ou características da forma humana de processar informações: a incerteza, a não garantia de repetibilidade, etc. Portanto, uma forma de lidar esse problema é reduzir a linguagem às

limitações da lógica, dessa forma garantir a exatidão do que se deseja expressar, em vez de tentar a aproximação da lógica à linguagem e inserir a imprecisão.

O que se propõe como fio condutor teórico nesta tese é o tratamento do texto através da redução do conteúdo expresso em linguagem natural para um conjunto determinado de léxicos compostos que tenham maior capacidade de expressar os significados desses conteúdos textuais, as Expressões Multipalavras[1] (EM), e utilizá-las como descritores de busca em um SRI.

## 1.1 Delimitação do problema

Esta tese está embasada nos pressupostos da Ciência da Informação na subárea de Organização e Uso da Informação, com aportes nos referenciais metodológicos oriundos das Ciências da Computação e da Estatística. Nesse contexto, buscou-se delimitar como objeto de estudo a interpretação do significado do texto a partir de técnicas algorítmicas determinísticas e estatísticas que usam as características estruturais do texto e do conceito de EM, pois acredita-se possuírem uma melhor representação semântica dos documentos do que as palavras de forma isolada.

A ideia que está por trás desta tese é a de pesquisar e comparar meios de extrair informações, ou seja, identificar documentos relevantes em um *corpus* sobre um tema de interesse do usuário de forma automatizada. Nesse sentido, adicionalmente às técnicas de buscas convencionais baseadas em descritores informados pelo requisitante, propõe-se uma abordagem para agregar uma alternativa de busca baseada em um **documento de referência** fornecido pelo requisitante. Essa abordagem proposta será denominada neste trabalho como **Busca Comparada**.

A forma de extração de informação apresentada nesta tese se mostra bastante adequada para usuários, normalmente pesquisadores e estudantes, que desejam, a partir de um artigo de referência da área de estudo buscar demais publicações que tratam de problemas correlatos realizando buscas automatizadas em *corpora* científicos específicos.

Os principais mecanismos de busca utilizados atualmente tais como Google[2], Yahoo[3] e Bing[4] funcionam através de uma interface de consulta na qual o usuário informa palavras-

[1] As Expressões Multipalavras são excertos de frases formados por duas ou mais palavras que, juntas, possuem uma expressividade semântica mais forte do que quando tratadas como termos em separado.

[2] www.google.com

[3] www.yahoo.com

[4] www.bing.com

chave a serem utilizadas como referência para a localização dos *links*, ordenados por relevância, para as páginas onde os termos foram encontrados. De forma semelhante, a grande maioria dos sistemas de bibliotecas digitais, sistemas de gestão de documentos e demais sistemas afins utilizam técnicas de busca semelhante a essas, ou até mesmo mais rudimentares, para recuperar documentos contidos em suas respectivas base de dados. Nesse contexto, mais controlado e delimitado, postula-se ser possível utilizar técnicas que busquem melhoria na qualidade das respostas obtidas pelos sistemas de recuperação da informação, explorando melhor a semântica intrínseca desses conteúdos. Entretanto, na prática, ao se buscarem conteúdos nessas bases de documentos, frequentemente, os resultados surpreendem, ora muito restritos, ora muito extensos. O ideal seria que a busca garantisse maior similaridade entre o desejo do usuário e o resultado produzido pela ferramenta de busca.

O uso de palavras-chave para efetuar buscas em um *corpus* de um domínio específico, tal como: base de teses, artigos científicos ou de bibliotecas digitais, apesar de ser muito utilizada atualmente e facilitar sobremaneira o processo de busca, ainda traz consigo muitos problemas na precisão das respostas.

A proposta deste trabalho é viabilizar uma alternativa à busca de documentos em um *corpus* através de palavras-chave, que seja de forma automatizada e independente de idioma. Nesse sentido, um documento fornecido pelo usuário servirá como referência de busca. Ou seja, o processo da Busca Comparada buscará documentos similares ao utilizado como referência. Desse modo, busca-se responder se seria viável e vantajoso agregar alternativas de busca por expressões multipalavras aos programas tradicionais de recuperação da informação.

## 1.2 Objetivo geral

Propor e analisar comparativamente uma metodologia de recuperação de informação que utiliza um documento como referência para a busca em um *corpus* específico. Ou seja, a Busca Comparada, sendo a função de similaridade medida através da ocorrência de expressões multipalavra.

### 1.2.1 Objetivos específicos

Para atingir ao objetivo geral, os seguintes objetivos específicos deverão ser almejados:

- Propor e implementar uma técnica determinística e automatizada para extrair as EM do documento de referência, com o *software* específico elaborado para esse fim;
- Comparar os resultados obtidos pelo uso de técnicas estatísticas e determinísticas na extração de EM a serem utilizadas como descritores do processo de recuperação da informação avaliando os critérios de precisão e tempo de resposta computacional;
- Propor e testar uma metodologia para implementar a Busca Comparada através de um componente de *software* que seja capaz de extrair as EM do documento de referência e utilizá-las como um conjunto de descritores, *n*-gramas, da busca na coleção de documentos.
- Testar os resultados obtidos com a ferramenta proposta.

## 1.3 Justificativa

É notório o crescente aumento das bases de dados digitais com documentos científicos e da facilidade de acesso proporcionado pelas tecnologias da informação. Entretanto, ao empreenderem-se pesquisas por palavras-chave nessas coleções de documentos, normalmente, somos surpreendidos por respostas compostas por uma enorme quantidade de documentos, mas, muitas das vezes, em grande parte, não correspondem à real necessidade da busca desejada.

Dessa forma, cabe ao usuário analisar dentre esse conjunto enorme de respostas aquelas que melhor se ajustam à sua requisição. Uma tarefa que nem sempre é factível dado o volume de respostas retornado. Uma das possíveis causas dessa situação decorre das buscas exclusivas por palavras-chave comparadas aos documentos em que elas ocorrem possuírem um menor teor semântico do que outras técnicas que lidam com a semântica intrínseca contida nos documentos do *corpus.* Afinal, muitas palavras perpassam por diversas áreas do conhecimento e é comum recuperar documentos que contenham o termo utilizado na busca, mas que possuem um outro significado, ou mesmo, aplicado à área distinta da área de interesse do requisitante. Diante de tantos resultados cabe ao usuário analisá-los, no entanto, a capacidade humana em ler e interpretar todas essas

informações se mantém constante e limitada à disponibilidade de tempo. Ramisch exemplifica essas ideias.

> Mecanismos de busca especializados precisam levar em conta a terminologia do domínio, por exemplo, os termos árvore e folha possuem estatuto terminológico, mas os conceitos que representam não são os mesmos em botânica e em informática. Em resumo, com a tecnologia atualmente disponível, o usuário que exprime uma necessidade de informação específica a um domínio usando sua língua mãe precisa interrogar diversas fontes externas para traduzir as palavras-chave e os documentos retornados na busca. (RAMISCH, 2009 p. 64)

Portanto, este trabalho se justifica, pois tem como objetivo propor uma forma de facilitar o processo de seleção de documentos, ao mesmo tempo em que impõe restrições automatizadas para recuperar informações, a partir da implementação de uma metodologia de Busca Comparada. Essa metodologia possibilita ao usuário uma alternativa de busca na qual em vez de informar palavras-chave como elemento de busca, caberá ao usuário informar um documento a ser usado pelo SRI como referência para a busca. Dessa forma, serão extraídas todas as EM encontradas no documento informado pelo usuário, para serem utilizadas como descritores compostos de busca. Em outras palavras, as buscas serão feitas a partir dos *n*-gramas extraídos. Essa estratégia alternativa simplifica o trabalho do usuário, que passa a utilizar documentos conhecidos sobre o tema de seu interesse para servir como base da Busca Comparada da recuperação de documentos similares.

A Figura 1 mostra um diagrama da estrutura do protótipo de *software* proposta nesta tese que se apresenta como um módulo de Busca Comparada adicional, em destaque, que pode ser agregado aos sistemas convencionais de busca por palavras.

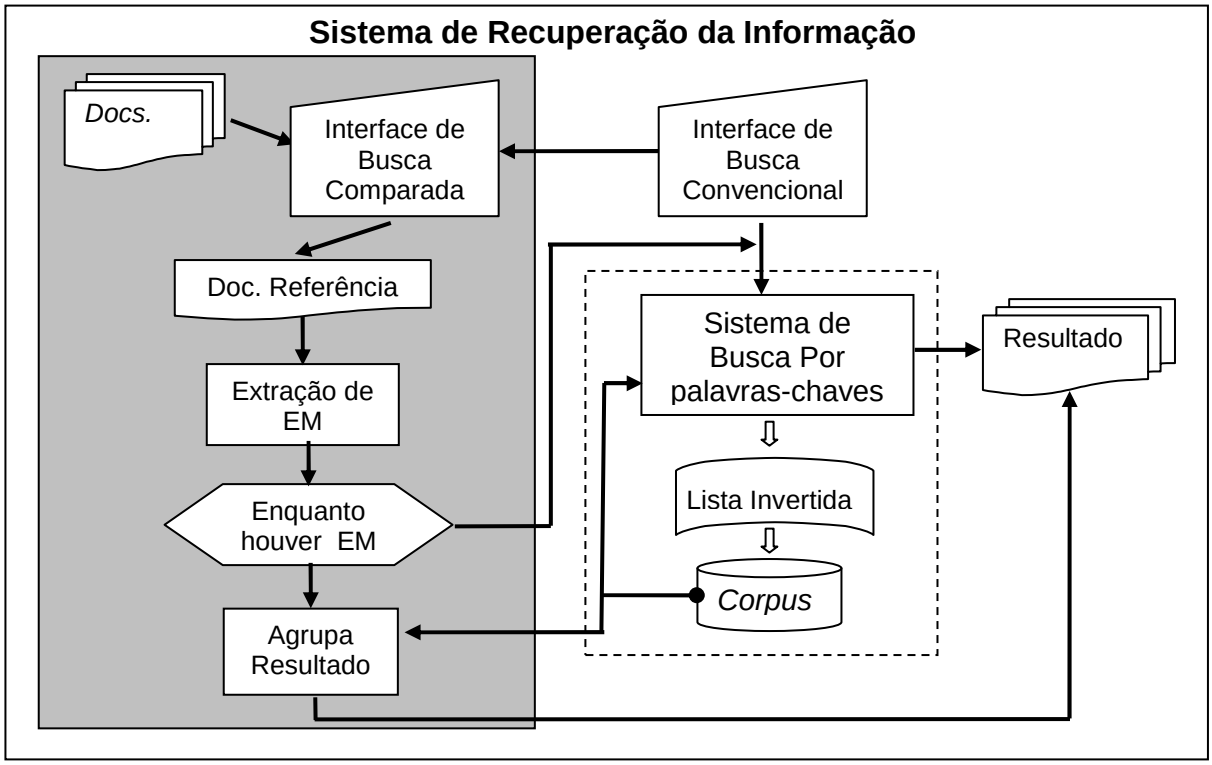



FIGURA 1 – Módulo de Busca Comparada agregado ao SRI.
Fonte: Elaborada pelo autor.

## 1.4 Estrutura da tese

Esta tese está estruturada em seis capítulos:

Nesta introdução, apresentou-se uma visão geral e evolutiva dos caminhos de pesquisa do PLN, diante da perspectiva de automatização da extração de sentido de um texto. Essa breve discussão serve apenas como forma de apresentar uma primeira delimitação do escopo deste trabalho dentro dessa grande área de pesquisa. Nesse propósito, o foco está na redução da extração de sentido de um texto, representando-o através de expressões multipalavras dependentes que co-ocorrem em uma frequência acima de um limite pré-definido. Adicionalmente, características estruturais do texto são utilizadas no processo de identificação.

No segundo capítulo, apresentam-se os fundamentos conceituais que sustentam o desenvolvimento deste trabalho, numa abordagem que parte de conceitos genéricos para os mais específicos com objetivo de seguir a construção teórica seguida pelo autor deste trabalho.

No terceiro capítulo, apresenta-se todo o roteiro metodológico empregado para construção do processo de identificação dos *n*-gramas, ou mais especificamente dos bigramas, em textos através de formas determinísticas e estatísticas.

No quarto capítulo, verificam-se vários experimentos para analisar e comparar os resultados obtidos pelo uso de ambas as técnicas e também o resultado final obtido pelo uso da ferramenta proposta.

No quinto capítulo, mostram-se as conclusões obtidas através da condução de todo o processo teórico prático desenvolvido.

Finalmente, no sexto capítulo, indicam-se algumas direções de futuras pesquisas relacionadas com o tema deste trabalho.

# 2 FUNDAMENTOS CONCEITUAIS

Por ser este um trabalho de cunho multidisciplinar, são apresentados, neste capítulo, os marcos teóricos e alguns conceitos norteadores que ultrapassam as fronteiras da área da CI, mas são necessários para o completo entendimento e construção teórica desta tese.

Na primeira seção, apresentam-se conceitos sobre os fundamentos linguísticos necessários para entendimento da estrutura e da interpretação do texto.

Na segunda seção, apresentam-se alguns fundamentos do processamento de linguagem natural e algumas das estratégias para tratar o texto escrito, numa perspectiva mais geral do tema que auxiliam no entendimento dessas tarefas.

Na terceira seção, apresentam-se os fundamentos conceituais sobre EM, sob as perspectivas estatística e linguística. Adicionalmente, apresenta-se uma delimitação da abrangência conceitual sobre EM adotada nesse trabalho.

Na quarta seção, apresentam-se os fundamentos das técnicas de medidas de associação estatísticas com o foco voltado para os processos de identificação de EM.

Na quinta seção, apresenta-se um retrospecto dos conceitos de indexação e linguagens documentárias utilizados pelos documentaristas que servem de fundamento para os processos automatizados. Em sequência, apresentam-se alguns dos principais fundamentos dos sistemas de recuperação da informação e os seus principais conceitos, com ênfase nas técnicas utilizadas na construção do ferramental de *software* utilizado neste trabalho.

Finalmente, na sexta seção, apresentam-se a descrição do estado da arte das pesquisas e trabalhos relacionados com o tema de estudo.

## 2.1 Fundamentos linguísticos

O objetivo dessa seção é apresentar alguns conceitos e definições básicas sobre a linguagem, afinal, o entendimento de alguns desses conceitos é importante para direcionar o desenvolvimento desta tese.

Cintra (1983 p. 7) apresenta o termo linguagem como sendo uma faculdade natural, enquanto que o termo língua refere-se a um caso particular de linguagem.

> A linguagem é uma representação simbólica que expressa uma função psicossocial complexa. Corresponde a uma manifestação intelectual e multiforme dos seres, que recobre inúmeras formas de significar: linguagem verbal (oral e escrita), a pictórica, a musical, a cinética, a mímica, a documentária, etc. (CINTRA, 1983 p. 7)

Por ser a linguagem a forma utilizada para mediar as relações humanas na elaboração cognitiva do pensamento e na comunicação para a troca de informações, como era de se esperar, ela também é utilizada para registrar a fala e o pensamento na forma de texto. Este trabalho abrange apenas um recorte desse tema, ao lidar somente com o texto escrito, dentro de um domínio específico, expresso em linguagem natural e armazenado no formato digital.

Cipro Neto e Infante (2009, p. 9) definem **linguagem** como sendo a capacidade de se comunicar por meio de uma língua. Eles definem **língua** como sendo um sistema de signos convencionados e utilizados por membros de uma comunidade. Já o **signo** como sendo um elemento representativo com aspectos do significante e do significado unidos por um todo indissociável. Portanto, o conhecimento de uma língua demanda conhecer a identificação de seus signos e o uso adequado de suas regras combinatórias.

Na perspectiva de Cintra (1983 p. 7-13), que apresenta uma definição em um nível mais abstrato, signo é uma unidade que está no sistema e na consciência do falante. Os signos são compostos pelo léxico da língua e pela palavra. O léxico é não-quantificável, composto das unidades que alimentam o vocabulário. Ao se criar uma nova entrada no vocabulário, tem-se um vocábulo. Os vocabulários são compostos de dois tipos de unidades: a) Morfema Lexical – contém o significado lexical, ou seja, expressam o “suporte de conceito” do mundo biossocial e b) Morfema Gramatical – contém significado gramatical, por isso mesmo é denominado “indicador de função”. Alguns autores consideram a palavra como sendo uma unidade formal composta de morfemas definidos dentro de uma língua, outros como uma unidade de texto. Na prática, não há um consenso entre os linguistas sobre qual é a definição de palavra ou termo. A linguagem pode ser estudada essencialmente perante a perspectiva gramatical, semântica e pelo sistema que relaciona ambas como as subseções subsequentes mostrarão.

### 2.1.1 Gramática

Cipro Neto e Infante (2009, p. 14-16) apresentam uma definição de **gramática** como sendo a designação para um conjunto de regras que garantem o uso modelar da língua. Ou seja, a gramática estabelece a norma culta e as regras que asseguram o uso correto da língua. O estudo da gramática é convencionalmente dividido em:

- **Fonologia** – Estuda os fonemas ou sons da língua e as sílabas que esses fonemas formam;
- **Morfologia** – Estuda a estrutura, a formação e os mecanismos de flexão das palavras, além de dividi-las em classes gramaticais;
- **Sintaxe** – Estuda as formas de relacionamentos entre as palavras ou entre orações, a qual inclui e regência, a colocação pronominal e a concordância.

Por ser a fonologia a parte que estuda os sons da língua, ela extrapola o escopo deste estudo, delimitado ao texto escrito. Portanto, tratar-se-á apenas de alguns dos aspectos da morfologia e sintaxe que são relacionados à compreensão do tema.

Conforme define Cipro Neto e Infante (2009, p. 73-74), a morfologia estuda a estrutura, formação, flexão e classificação das palavras. Sendo que cada uma delas é formada por **morfemas**, que são os elementos que a constituem. Esses elementos indecomponíveis são unidades de significação mínima que agregam significado à palavra. Segundo Faraco & Moura (1990, p. 132-138), os principais processos de formação das palavras são a derivação e a composição. A **derivação** é o processo de formação da palavra a partir de uma outra que já existe na língua. A **Composição** refere-se à junção de duas ou mais palavras ou radicais para formação de uma nova palavra. Os autores citam outros processos de formação de palavras como sendo: hibridismo, onomatopeia, siglonimização e abreviação vocabular.

Cintra (1983, p. 6) define morfologia como a disciplina que sintetiza parcialmente aspectos da semântica e da sintaxe, por se encarregar da identificação das partes da palavra e de suas condições de ocorrência.

Faraco & Moura (1990, p. 144-147) definem que cada palavra tem uma finalidade no ato de comunicação oral ou escrita. De acordo com essa finalidade as palavras se enquadram nas seguintes classes gramaticais:

- Substantivo – dá nome aos seres;
- Adjetivo – caracteriza os seres;
- Verbo – indica fato ou estado;
- Pronome – representa ou acompanha o substantivo considerando-o como pessoa do discurso;
- Numeral – indica a quantidade ou ordem dos seres;
- Artigo – acompanha o substantivo, determinando ou indeterminando-o;

- Advérbio – indica circunstância de tempo, modo, lugar, intensidade, etc;
- Preposição – liga dois termos na oração;
- Conjunção – relaciona duas orações ou termos semelhantes de uma mesma oração;
- Interjeição – expressa sentimento ou emoção.

As classes de palavras são normalmente divididas em duas. A de categoria léxica ou aberta tais como substantivos, verbos, adjetivos e advérbios os quais possuem um grande número de membros, e para os quais novas palavras são comumente adicionadas. E a categoria funcional ou fechada, que possui um número finito de palavras e tem claro uso gramatical, na qual enquadram-se os pronomes, os artigos, as preposições e as conjunções.

As palavras podem ser flexionadas mudando sua terminação para exprimir outros significados. **Flexão** é modificação sistemática da forma raiz por meio de prefixo ou sufixo para indicar distinções gramaticais tipo singular e plural. Flexão não muda a classe da palavra ou altera o seu significado, mas varia características tais como tempo, número e plural. Toda forma flexionada de uma palavra é frequentemente agrupada como manifestações de um morfema. A tipologia das flexões é relacionada a seguir:

- Flexão de número – é a mudança da terminação para indicar singular ou plural;
- Flexão em grau – é terminação utilizada para indicar tamanho nos substantivos e intensidade nos adjetivos e advérbios;
- Flexão de Tempo – existe apenas para os verbos, e indica o momento da ocorrência do fato presente, passado ou futuro;
- Flexão de modo – só existe para os verbos e serve para indicar as diferentes atitudes do emissor em relação ao fato que se quer expressar. Sendo três as possibilidades: indicativo, subjuntivo ou imperativo;
- Flexão de pessoa – permite flexionar o verbo de acordo com a pessoa gramatical: emissor, receptor, ou de que/quem se fala.

Cintra (1983, p. 6) define **sintaxe** como a disciplina que se ocupa das relações que se estabelecem a partir da organização sintagmática dos elementos e funcionamento do significado do signo, visto como elemento do sistema lexical de uma língua. Para Faraco & Moura (1990, p. 307-310) uma mensagem linguística é formada por palavras e o estudo da combinação e relação entre as palavras é denominado sintaxe. A análise sintática estuda um texto a partir de suas partes definidas a seguir:

- Frase – é um conjunto de palavras que formam o sentido completo (sentença);
- Oração – é uma frase constituída de sujeito e predicado, ou apenas predicado;
- Período – é um conjunto de orações que formam sentido completo;
- Período Simples – frase constituída de uma só oração;
- Período Composto – frase constituída de duas ou mais orações;
- Composição por Subordinação – são orações sem autonomia gramatical, isto é, as orações que funcionam como parte, integrantes ou acessórios de outra oração;
- Composição por Coordenação – são as orações que têm sentido próprio;
- Sintagma – sequência de elementos linguísticos relacionados entre si;
- Sintagma Nominal – conjunto de substantivos e seus adjuntos;
- Sintagma Verbal – conjunto de verbos e seus adjuntos;
- Sintagma Preposicional: são grupos preposicionais não ligados, independentes da noção de regência;
- Sintagma Adjetival – formado por adjetivo ou grupos de adjetivos;
- Sujeito – termo com o qual o verbo concorda;
- Predicado – tudo aquilo que se diz do sujeito;
- Complemento Verbal – palavras que integram o sentido do verbo;
- Complemento Nominal – palavras que completam o sentido de substantivo, adjetivo ou advérbio;
- Adjunto Adverbial – denota alguma circunstância do fato expresso pelo verbo, ou intensifica o sentido deste;
- Adjunto Adnominal – serve para especificar ou delimitar o significado de um substantivo;
- Aposto – é o termo que se junta a um substantivo, a um pronome, ou a um equivalente destes, a título de explicar, especificar, enumerar, ou resumir;
- Preposição: é o vocábulo que relaciona dois termos de uma oração, de tal modo que o sentido do primeiro (antecedente) é explicado ou completado pelo sentido do segundo (consequente);

A análise sintática é uma técnica empregada no estudo da estrutura de uma sentença, seus períodos e orações. É um passo importante para o entendimento (semântica) de uma sentença em linguagem natural. Somente vocábulos não garantem o entendimento de uma sentença, é importante que a sua estrutura sintática seja analisada. Na análise sintática de uma oração em português deve levar em conta os seguintes sintagmas: termos essenciais (sujeito e predicado), termos integrantes (complementos verbal e nominal) e termos acessórios (adjunto adverbial, adjunto adnominal e aposto). A análise do período, por sua vez, deve considerar o tipo de período (simples ou composto), sua composição (por subordinação, por coordenação) e a classificação das orações (absoluta, principal, coordenada ou subordinada).

### 2.1.2 Semântica

A palavra semântica é um adjetivo relacionado com sentido. Cintra (1983, p. 6) define a semântica como a disciplina que se ocupa do sentido ou da significação dos elementos. Barros (1991) corrobora com a definição apresentada anteriormente ao afirmar que semântica é tudo o que se refere ao significado. Semântica é o estudo dos mecanismos que atuam na significação dos morfemas, das palavras ou dos enunciados.

Dentre os estudos de semântica, a que está mais relacionada ao texto é a semântica estrutural que se ocupa do estudo descritivo da natureza e funcionamento do significado do signo, visto como elemento do sistema lexical de uma língua. Enquanto que a semântica gerativa estuda a competência do falante nativo, nível do significado por considerá-lo dotado de informações semânticas básicas e regras de projeção que lhe permitem produzir e reconhecer as frases e as suas ambiguidades, pois na língua portuguesa há palavras, componentes da palavra e frases com vários significados. Portanto, a questão da representação do significado apresenta diversas dificuldades. Podem-se mencionar um exemplo: a questão da ambiguidade, como no verbo tomar, “tomar de alguém”, “tomar um banho”, ou em “tomar suco”.

Russell & Norvig (2004, p. 767) afirmam que toda cadeia válida, ou seja, conteúdo expresso, baseado tanto em uma linguagem formal quanto em uma linguagem natural carrega em si um significado ou semântica. Como exemplo, eles utilizam a linguagem da aritmética que elucida bem a questão, veja-se: uma expressão aritmética $+ X Y$, não é considerada válida, pois foge às regras da gramática aritmética; já a expressão $X + Y$ é válida e tem como representação semântica a soma dos valores contidos nas variáveis $X$ e $Y$. Na medida em que se sai das restrições impostas pelas regras da aritmética e se

expande para o universo da linguagem passa-se a ter uma enorme ampliação das regras da gramática da língua, dos léxicos que podem ser utilizados e das formas de combiná-los nas sentenças e, por conseguinte chega-se a um universo, possivelmente infinito, de significações.

Para se tentar extrair a significação de um texto é importante entender como ocorre o processo de comunicação entre um elemento transmissor e outro receptor. Russell & Norvig (2004, p. 768-770) dividem esse processo nas seguintes etapas:

- Intenção – ocorre a partir da vontade do transmissor em comunicar uma proposição;
- Geração – é o planejamento do modo de transformar a proposição em uma expressão baseada na linguagem que seja capaz, ou pelo menos espera-se que seja compreendida pelo receptor;
- Síntese – ato de transformar o plano em ação, através de um meio: fala, escrita, etc.
- Percepção – é a decodificação da comunicação física realizada pelo ouvinte;
- Análise – é subdividida em três partes: a análise sintática, a interpretação semântica e a interpretação pragmática;
- Eliminação da ambiguidade – dentre as possíveis interpretações o receptor escolhe aquela que mais provavelmente o transmissor queria expressar;
- Incorporação – tomada de decisão entre acreditar ou não no conteúdo recebido.

### 2.1.3 Pragmática

Segundo Rodrigues & Caricatti (2009, p. 124), existem basicamente duas formas de abordar o fenômeno comunicativo entre as pessoas: o que foca os aspectos vistos nas seções anteriores relacionados à estrutura da língua, e outra que considera a língua como um fenômeno social, fruto de uma relação dialética entre a linguagem e a sociedade com seus valores, suas crenças no seio de lutas de poder.

A pragmática é uma subárea da linguística que contextualiza a linguagem no âmbito social e cultural. Isto a torna ainda mais distante da possibilidade de ser tratada pelas máquinas baseadas nas tecnologias computacionais atuais. Portanto, extrapola o escopo

deste trabalho, que é delimitado para uma abordagem do processo de extração de sentido de um texto escrito através da identificação de *n*-gramas de forma interdependente.

Desse modo, neste trabalho busca-se tratar o texto escrito utilizando algumas das várias técnicas do PLN a fim de facilitar os processos de recuperação dos documentos a partir de aspectos semânticos embarcados pelas EM. A seguir, mostram-se alguns desses conceitos.

## 2.2 Processamento de Linguagem Natural

O PLN é uma subárea da Inteligência Artificial (IA) da qual herda muito de seus métodos e princípios. O PLN surgiu a partir da demanda em estabelecer a comunicação entre o homem e o computador através da compreensão e produção de linguagem natural, como por exemplo: o português e o inglês. Os primeiros trabalhos de tratamento do PLN com implementações informatizadas surgiram no início da década de 50 do século passado. A partir da década de 60, surgiram várias aplicações voltadas para a compreensão da linguagem natural, capazes de aceitar e de responder a questões em inglês sobre diversos assuntos tais como: álgebra, medicina, relações de parentesco, etc.

A área de PLN, também denominada Linguística Computacional, é o conjunto de métodos formais para analisar textos e gerar frases escritas em um idioma humano. Seu objetivo final é fornecer aos computadores a capacidade de entender e compor textos. Sendo que "entender" um texto significa reconhecer o seu contexto, fazer análise morfológica, sintática, semântica, pragmática, criar resumos, extrair informação, interpretar os sentidos e até aprender conceitos a partir dos textos processados. Uma segmentação mais abrangente dessas etapas é apresentada por Chowdhury (2003, citado por Ladeira, 2010 p.67), segundo o qual a análise da linguagem natural pode ser realizada a partir de sete níveis interdependentes: fonológico, morfológico, léxico, sintático, semântico, discursivo e pragmático. Sendo que, para realizar o PLN, é necessário distinguir todos ou alguns desses sete níveis.

Segundo Nunes, Vieira & Lima (2007), o PLN lida com problemas relacionados à automação da interpretação e da geração da linguagem humana em aplicações como Tradução Automática, Sumarização Automática de Textos, Ferramentas de Auxílio à Escrita, Sistemas de Perguntas e Respostas, Categorização Textual, Recuperação e Extração de Informação, entre muitas outras. Além das tarefas relacionadas à criação e disponibilização de dicionários/léxicos e *corpus* eletrônicos, desenvolvimento de taxonomias e ontologias, investigações em linguística de *corpus*, desenvolvimento de esquemas de marcação e

anotação de conhecimento linguístico-computacional, resolução anafórica [5] , análise morfossintática automática, análise semântico-discursiva automática, etc.

Conforme explica Ramish (2009 p. 64), uma parte importante do conhecimento humano é expresso através da linguagem natural. A língua funciona como meio de registro do conhecimento científico. Consequentemente, sistemas de PLN têm grande interesse nesse contexto pelas características do seu léxico, rico em estruturas terminológicas. Acredita-se que a terminologia possa ser adquirida automaticamente através de *corpora* de domínio específico e técnicas de aprendizado dirigido pelos dados.

Entretanto, os computadores normalmente estão aptos, apenas para compreender de forma determinística, instruções escritas em linguagens formais, com vocabulário controlado. Como é o caso das instruções de um programa fonte escrito em uma linguagem de programação, tal como C++, Java, dentre outras; ou ainda, linguagens relacionadas à interface do sistema operacional com o usuário, tanto através das linhas de comando no ambiente caractere, quanto através da metáfora do *desktop* no ambiente gráfico. Afinal as formas de linguagens tratadas pelos computadores se restringem a um contexto específico com um conjunto controlado de termos tratados de forma precisa, contendo regras fixas e estruturas lógicas bem definidas. Ou seja, a interação com o computador é feita através de uma linguagem própria, limitada em termos lexicais e gramaticais, isenta de ambiguidades, o que permite ao computador saber exatamente como deve proceder a cada comando. Portanto, ainda existe muita dificuldade em processar instruções escritas ou faladas na linguagem humana de forma livre, pois, em um idioma humano, uma simples frase normalmente contém ambiguidades, nuances e interpretações que dependem do contexto, do conhecimento do mundo, de regras gramaticais, culturais e de conceitos abstratos. Isso torna a modelagem computacional da linguagem uma tarefa bastante complexa.

As aplicações que tratam a línguagem natural podem ser divididas em duas classes: aplicações baseadas em texto e aplicações baseadas em diálogos. As aplicações baseadas em texto são sistemas que procuram documentos específicos em uma base de dados (exemplo: encontrar livros relevantes em uma biblioteca), tradutores de documentos, e sistemas que resumem textos (exemplo: produzir três páginas resumidas de um livro de cem páginas). Com relação às aplicações baseadas em diálogos, pode-se citar as interfaces de linguagem natural para bancos de dados, os sistemas tutores e os sistemas que interpretam

[5] Referência anafórica é definida por Oliveira e Wazlawick (1998) como sendo um fenômeno linguístico que ocorre quando um pronome ou sintagma nominal em uma frase faz referência a alguém ou a um objeto previamente mencionado no texto.

e respondem comandos expressos em linguagem escrita ou falada. Para um sistema ser considerado um tratador da língua natural, duas condições devem ser satisfeitas:

- Um subconjunto de entrada e/ou saída do sistema é codificado em uma linguagem natural;
- O processamento da entrada e/ou a geração da saída é baseada no conhecimento sobre aspectos sintáticos, semânticos e/ou pragmáticos de uma linguagem natural.

Pode-se perceber, principalmente analisando a segunda condição, que a exigência da interpretação do conteúdo de uma sentença não é satisfeita nos sistemas que processam a linguagem natural puramente como *strings,* como, por exemplo, nos editores de texto e nos pacotes estatísticos. Dessa forma, o PLN em seus processos e no desenvolvimento de recursos, ferramentas e aplicações tem uma forte interação interdisciplinar, principalmente com as áreas de Linguística, CI, CC e IA.

> Os problemas básicos de se compreender a informação e a comunicação, suas manifestações, o comportamento informativo humano e os problemas aplicados ligados ao "tornar mais acessível um acervo crescente de conhecimento", incluindo as tentativas de ajustes tecnológicos, não podem ser resolvidos no âmbito de uma única disciplina. Este fato ficou claro, a partir da afirmação de BUSH, para todos que refletiram acerca das complexidades envolvidas. Problemas complexos demandam enfoques interdisciplinares e soluções multidisciplinares. (SARACEVIC, 1996, p. 48)

Geralmente para que um sistema computacional interprete uma sentença em linguagem natural, é necessário manter informações morfológicas (léxicas), sintáticas (regras gramaticais) e semânticas (significados) armazenadas em um dicionário, juntamente com as palavras que o sistema compreende. O processo segue as seguintes etapas: primeiro, o analisador morfológico identifica palavras ou expressões isoladas em uma sentença, sendo esse processo auxiliado por delimitadores de pontuação e espaços em branco. As palavras identificadas são classificadas de acordo com seu tipo de uso ou categoria gramatical. Em seguida, o analisador sintático reagrupa as estruturas das palavras e o analisador semântico analisa o sentido das mesmas. Os significados são compostos pelas estruturas criadas pelo analisador sintático.

O emprego do analisador morfológico é fundamental para a compreensão de uma frase, pois, para formar uma estrutura coerente de uma sentença, é necessário compreender o significado de cada uma das palavras componentes. Segundo Manning & Schütze (2003), as partes do discurso (POS[6]) são os grupos linguísticos de palavras de uma

[6] Tradução literal do termo original em inglês *parts of speech* (POS)

linguagem que tem um comportamento sintático similar, e frequentemente um típico estilo semântico. Essas classes de palavras são chamadas de categorias sintáticas ou gramaticais. Três importantes partes de um discurso são substantivo, verbo e adjetivo. Substantivos tipicamente referem-se a pessoas, animais, conceitos e coisas. O verbo é utilizado para expressar ações e estados nas sentenças. Adjetivos descrevem propriedades dos substantivos.

As categorias de palavras são sistematicamente relacionadas por processo morfológico, tal como a formação do plural. A morfologia é importante no PLN, porque a linguagem é produtiva: em qualquer texto dado, encontrar-se-ão palavras já catalogadas e outras formadas que não foram vistas antes, portanto, não existe em dicionário pré-compilado. Muitas dessas palavras são morfologicamente relacionadas com palavras conhecidas. Ao entender-se o processo morfológico, pode-se inferir muito sobre as propriedades sintáticas e semânticas dessas novas palavras.

É importante estar apto para manipular morfologia em inglês, mas é absolutamente essencial, quando a linguagem é altamente flexionada como é o caso do finlandês. Enquanto no inglês um verbo regular tem somente quatro formas distintas, e verbos irregulares têm no máximo oito formas, os verbos finlandeses têm mais de 10.000 formas de conjugação. Seria tedioso, ou até mesmo não factível, manter todas as formas dos verbos em uma enorme lista.

O analisador sintático procura construir árvores de derivação para cada sentença, mostrando como as palavras estão relacionadas entre si. Esse processo se dá através das regras gramaticais da linguagem a ser analisada e das informações do analisador morfológico. Durante a construção da árvore de derivação é verificada a adequação das sequências de palavras às regras de construção impostas pela linguagem, na composição de frases, períodos ou orações. Dentre essas regras, pode-se citar a concordância e a regência nominal e/ou verbal, bem como o posicionamento dos termos em uma frase. Um termo corresponde a um elemento de informação (palavra ou expressão) e é tratado como unidade funcional da oração, participando da estrutura como um de seus constituintes, denominados sintagmas.

Para Oliveira (2011), nos sistemas de PLN, o maior problema é a transformação de uma frase potencialmente ambígua em uma não ambígua. Essa transformação é conhecida como desambiguação. As abordagens de linguagens formais são utilizadas com muito sucesso no estudo da análise sintática. Dentre elas:

- **Gramáticas Regulares:** para o processamento sintático da linguagem natural, elas são bastante simples e facilmente reconhecidas, porém, apresentam um

poder de expressão limitado (equivalente ao poder de expressão de um autômato finito).

- **Gramáticas Livres de Contexto:** são muito úteis no que tange à descrição de gramáticas em linguagem natural. Em geral, são mais poderosas que as regulares, permitindo a representação de linguagens com certo grau de complexidade. No entanto, a dificuldade em expressar dependências simples (exemplo: concordância entre verbo e sintagma nominal) constitui um dos maiores problemas para sua utilização no tratamento da língua natural. Abordagens puramente livres de contexto não são suficientemente poderosas para captar a descrição adequada desse gênero de linguagem. Ainda assim, é utilizada uma notação denominada como Definite Clause Gramma (DCG), disponível em Prolog, para definir gramáticas livres de contexto e analisar sentenças através do processamento do *parsing*.
- **Gramáticas Sensíveis ao Contexto:** os problemas de dependência expressos anteriormente são resolvidos nessa classe de gramática. Ainda assim, as gramáticas sensíveis ao contexto não abordam satisfatoriamente o tratamento de restrições gramaticais. O impedimento para sua utilização, contudo, reside na questão do reconhecimento. Ou seja, o problema de decidir se uma sentença pertence a uma gramática sensível ao contexto é uma função exponencial sobre o tamanho da sentença, o que torna a implementação do procedimento de verificação uma questão complexa, do ponto de vista computacional.

A maioria das pesquisas propõem trabalhar em modelos que se situam em um nível intermediário entre as gramáticas livres de contexto e as sensíveis ao contexto, aliando boa capacidade de representação, incluindo construções que permitam modelar as dependências, e um modelo computacional viável. (Oliveira, 2009). Desse modo, são várias as direções das linhas de pesquisa, dentre os quais destacam-se aqueles que utilizam métodos linguísticos e os que utilizam métodos estatísticos.

Este trabalho delimita-se por um recorte do tema, o qual considera as características estruturais do texto no processo de extração de sentido através da identificação de *n*-gramas, ou mais especificamente bigramas, que também são conhecidos como Expressões Multipalavras (EM), as quais serão descritas na próxima seção.

## 2.3 Expressões multipalavras

Inicialmente, faz-se necessário definir três conceitos fundamentais para este trabalho: Expressão Multipalavras (EM), Termo Técnico-Científico (TCC) e Termo Multipalavras (TM).

A definição de EM é ampla, pois engloba diversos fenômenos distintos como compostos nominais, expressões idiomáticas e termos compostos. As EM são necessariamente compostas por mais de uma palavra.

Os TCC e os TM são fenômenos linguísticos ligados ao texto técnico-científico definidos como locuções que possuem estatuto terminológico. Sendo que os TCC podem ser unidades lexicais únicas, aceitam pouca variabilidade (morfológica, raramente sintática) e representam um único conceito. Enquanto os TM não correspondem ao conceito de fraseologia do domínio, são altamente flexíveis e normalmente possuem uma estrutura complexa que associa mais de um conceito.

A seguir são apresentadas uma definição para cada um desses termos citados no trabalho de Ramisch (2009, p. 65):

- EM é um conjunto de duas ou mais palavras com semântica não-composicional, ou seja, o sentido do sintagma não pode ser compreendido totalmente através do sentido de suas componentes (Sag et al. 2002).
- TCC é uma unidade lexical ou multilexical com significado não ambíguo quando empregada em textos especializados, ou seja, a terminologia de um domínio é a representação linguística dos seus conceitos (Krieger and Finatto 2004)[7].
- TM é um termo composto por mais de uma palavra. (SanJuan et al. 2005[8], Frantzi et al. 2000[9]).

Feitas essas considerações iniciais, destaca-se que o foco desta tese está nas EM. A seguir são apresentadas definições encontradas na revisão da literatura sobre o tema, sendo que, ao fim dessa seção, como conclusão apresenta-se a definição de EM adotada neste trabalho.

---

[7] Artigo intitulado “Introdução à Terminologia: teoria & Prática” publicado pela editora Contexto em 2004, citado por Ramisch (2009).

[8] Artigo intitulado “A symbolic approach to automatic multiword term structuring” publicado em 2005 no 19° volume, páginas 524 a 542, citado por Ramisch (2009).

[9] Artigo intitulado “Automatic recognition of multi-word terms: the C-value/NC-value method. Publicado em 2000 no International Journal on Digital Libraries, páginas 115 à 130, citado por Ramisch (2009).

Na realidade, não existe uma definição formal consensual na literatura sobre EM. Em linhas gerais, considera-se que as EM são formações compostas de duas ou mais palavras que, quando associadas, possuem uma expressividade semântica mais forte do que quando cada um de seus termos são postos separadamente. Para Sag et al. (2002 p. 2) EM são: “interpretações idiossincráticas que cruzam os limites (ou espaços) entre as palavras”. Outra descrição para o uso do termo EM é:

> [...] expressão multipalavra vem sendo utilizado para descrever um grande número de construções distintas, mas fortemente relacionadas, tais como verbos de suporte (fazer uma demonstração, dar uma palestra), compostos nominais (quartel general), frases institucionalizadas (pão e manteiga), e muitos outros. [...] EM engloba um grande número de construções, tais como: expressões fixas, compostos nominais e construções verbo-partícula. (VILLAVICENCIO et al. 2010 p. 16)

Segundo Ranchhod (2003, p. 2), as expressões fixas são objetos linguísticos que apresentam divergências terminológicas e a ausência de critérios de análise que os levaram ser consideradas como objetos linguísticos excepcionais, não integráveis na gramática das línguas. Entretanto, tem ocorrido um crescente interesse, sobretudo na área de PLN, afinal essas formas fixas são tão numerosas em qualquer tipo de texto, que não podem ser ignoradas. Portanto, essas características das EM as tornam relevantes no tratamento dos recursos lexicais, os quais são importantes insumos informacionais para muitas aplicações relacionadas ao PLN, tais como: tradução automática, sumarização de texto, etc.

Para Sarmento (2006), o texto não é um simples amontoado aleatório de palavras. A ordem da colocação das palavras no texto é que produz o sentido. Portanto, o estudo da co-ocorrência das palavras traz consigo uma informação importante. Isso pode indicar que as palavras estão relacionadas, diretamente por composicionalidade ou afinidade, ou indiretamente por semelhança. Portanto, a base da linguística empírica consiste em encontrar a partir da frequência de co-ocorrências observada, as dependências significativas entre os termos. Evert (2005 citado por Sarmento) aponta como sendo quatro esses grupos de medidas:

– testes de significância estatística;
– coeficientes de associação;
– baseadas em conceitos da teoria da informação;
– baseadas em heurísticas diversas.

Conforme expresso por Zhang et al. (2009), a capacidade de expressar sentido de uma palavra depende das demais palavras que a acompanham. Quando uma palavra aparece acompanhada por um conjunto de termos, maiores são as chances desse conjunto possuir um significado relevante. Isso significa que não apenas a palavra, mas também a informação contextual é útil para o processamento de informações. É a partir dessa ideia

simples e direta que pesquisas sobre EM são motivadas. Espera-se capturar conceitos semânticos relevantes do texto expressos pelas EM. Nesse sentido, Villavicencio et al. (2010) destaca que muitas pesquisas têm buscado formas de automatização na aquisição lexical. Esses trabalhos buscam entender a formação dos recursos lexicais, uma área ainda carente de pesquisas.

Sag (2002, p. 4) apresenta a seguinte classificação das EM:

- Expressões Fixas – são aquelas que não apresentam flexões morfossintáticas e não permitem modificações internas. Elas desafiam as convenções da gramática e interpretação composicional, pois ao tratá-las na forma de palavra por palavra não teríamos a representação da expressão composta, que tem um sentido próprio dado pela composição.
- Expressões Semi-Fixas – são aquelas que possuem restrições na ordem das palavras e composição, mas admitem eventuais variações léxicas na flexão, na forma reflexiva e na escolha de determinantes. Esse tipo de EM é categorizada em três subgrupos: as expressões não-decomponíveis; os compostos nominais; e os nomes próprios. A primeira categoria, termo em inglês *non-decomposable idioms,* ocorre quando se juntam duas ou mais palavras para formar uma expressão que possui um novo significado, distinto daquele obtido pelas palavras de forma isolada. Exemplo “chutar o balde”, que tem como significado composto a ideia de “desistir”. Nesse caso há variabilidade da expressão idiomática. A segunda categoria os compostos nominais, do inglês *compound nominals,* são similares às expressões não-decomponíveis sendo unidades sintaticamente inalteráveis que na maioria dos casos podem ser flexionadas em número. Vejamos como exemplo as expressões “presidente da república” e “deputado federal”, na primeira expressão somente presidente pode ser flexionado, enquanto que, na segunda, ambas as palavras são passíveis de flexão. A terceira categoria os nomes próprios, do inglês *proper names,* são sintaticamente altamente idiossincráticos. Vejamos por exemplo o composto “Espírito Santo”, pode estar relacionado ao estado federativo do Brasil, pode ser um sobrenome, etc.
- Expressões Sintaticamente Flexíveis – são expressões que admitem variações sintáticas na posição de seus componentes. Os tipos de variação possíveis são: construções verbo-partícula que consistem de construções de um verbo e uma ou mais partículas que podem ser semanticamente idiossincráticos ou composicional; expressões idiomáticas decomponíveis. Um exemplo é “tirar o

cavalinho da chuva". O termo decomponível é utilizado por que, nesse caso, o significado "desistir da ideia" pode ser decomposto em "tirar" (desistir de), "o cavalinho da chuva" (a ideia); construções verbo-leve, do inglês *light-verbs*, é um verbo considerado semanticamente fraco estando sujeito à variabilidade sintática completa, incluindo a passivação. Eles são altamente idiossincráticos, pois existe uma notória dificuldade em predizer qual verbo-leve combina com qual substantivo.

- Expressões Institucionalizadas – são expressões composicionais, do inglês *collocation*, que podem variar morfológica ou sintaticamente e que normalmente possuem alta ocorrência estatística.

Calzolari et al. (2002, p. 1934) corroboram com a classificação apresentada por Sag et al. (2002) e ainda incluem um "etc" no final. Ou seja, como os próprios autores definem EM é utilizada para descrever diferentes, mas relacionados fenômenos, que podem ser descritos como uma sequência de palavras que agem como uma unidade em algum nível de análise linguístico e que apresentam alguns ou todos dos seguintes comportamentos: reduzida transparência sintática e semântica; redução ou ausência de composicionalidade; mais ou menos estável; passível de violação de alguma regra geral sintática; elevado grau de lexicalização (dependendo de fatores pragmáticos); alto grau de convencionalidade. Ainda segundo esses mesmos autores, as EM estão situadas na interface entre a gramática e o léxico. Eles apresentam também algumas das causas das dificuldades ocorridas no âmbito teórico e computacional para o tratamento das EM, como sendo: a dificuldade de estabelecer limites claros para o domínio das EM; a falta de léxicos computacionais de tamanho razoável para auxiliar no PLN; perante a perspectiva multilingue, muitas vezes não é possível encontrar uma correspondência direta lexical equivalente; dificuldade generalização dos léxicos (geral e terminológico) para um contexto específico.

Segundo Moon (1998 citada por VILLAVICENCIO et al.), as EM são unidades léxicas formadas por um amplo contínuo entre os grupos composicionais e os não-composicionais ou idiomáticos. Nesse contexto, entende-se por expressão composicional aquelas que, a partir das características de seus componentes, determinam as características do todo. E não-composicional ou expressões idiomáticas aquelas cujo significado do conjunto de palavras nada tem a ver com o significado de cada uma das partes. Dadas essas características, ao tratar as EM como palavras separadas por espaço, certamente trará anomalias para o processo de RI.

A ocorrência das EM nas línguas, de maneira geral, são muito frequentes conforme é apontado por Biber et al. (1999, citado por Wang e Liu 2011). Segundo esses autores, na

língua inglesa, as EM representam de 30% a 45% do idioma falado e cerca de 21% da escrita acadêmica. Entretanto, esses números podem estar ainda subestimados, se se considerar que o surgimento de novas EM ocorrem com frequência, como por exemplo: computação em nuvens, energia limpa, etc. Wang e Liu (2011) reafirmam ainda que as EM são uma questão ainda a ser melhor resolvida pelas aplicações que lidam com PLN.

Após revisar a literatura na busca de encontrar uma definição consensual para EM, percebe-se que o termo tem um uso genérico o qual engloba vários conceitos ou subtipos conforme descritos anteriormente. Desse modo, empregam-se diferentes métodos ora estatísticos, ora linguísticos, ora uma combinação de ambos para identificar as EM de forma mais estrita. Portanto, faz-se necessário apresentar a definição de EM a qual será utilizada neste trabalho. Tomando-se como base que o objetivo é apresentar um método que possa ser utilizado, independente do contexto e do idioma para identificar descritores (*n*-gramas) em um documento de referência fornecido pelo usuário e utilizá-los no processo de busca de documentos similares apresentar-se-á a seguir a definição que melhor cabe a este trabalho. EM são expressões fixas que co-ocorrem em um documento com uma frequência acima de um limite pré-definido, considerando-se as características da estrutura do documento.

## 2.4 Medidas de associação estatística

As medidas de associação estatística são um instrumental muito utilizado nas pesquisas relacionadas com o PLN. Conforme descrito por Sarmento (2006), a ocorrência de um termo no documento sugeri a hipótese nula $p_{Ho}$ como sendo a probabilidade de ocorrência de um termo ser independente da ocorrência de um outro, expresso através da expressão apresentada em (2.1).

$$\boxed{p_{Ho}(t_1, t_2) = p(t_1)\,p(t_2)} \tag{2.1}$$

Desse modo, basicamente, o objetivo das técnicas de associação estatística é de rejeitar empiricamente a hipótese nula e quantificar o grau de dependência existente entre os termos que compõem o documento.

Neste trabalho utiliza-se o software NSP[10] proposto por Pedersen et al. (2000) para realizar a identificação automática das EM nos documentos do *corpus*. O NSP implementa treze técnicas distintas para identificação das EM. Sendo que cada uma dessas técnicas empregadas produzem como resultado uma lista de EM ordenadas por um critério de

[10] Ngram Statistic Package (NSP) - http://www.d.umn.edu/~tpederse/nsp.html

relevância próprio. Conforme detalhado por Banerjee e Petersen (2000), para cada documento processado pelo *software*, é realizado um processo de separação do texto em termos. Esses termos são agrupados par a par seguindo a mesma sequência na qual eles estão postos no texto original. Os pares repetidos são agrupados a fim de contabilizar a sua frequência de ocorrência. Em seguida, o *software* cria uma matriz de contingência para cada par de termos $T_1$ e $T_2$, que ocorrer com uma frequência igual ou acima de um limiar definido em parâmetro. A tabela de contingência é uma matriz que contém valores relativos à frequência das quatro combinações possíveis para as ocorrências de $T_1$ e $T_2$, ou seja, quando os termos ocorrem juntos; quando ocorrem de forma mutuamente exclusiva; e finalmente quando ambos não ocorrem.

Essa matriz de contingência serve de base para o cálculo de todas as medidas de associação estatística geradas pelo *software*. Um exemplo dessa matriz é mostrado na Figura 2.

| Termos | $T_2$ | $\sim T_2$ | Totais |
|---|---|---|---|
| $T_1$ | $n_{11}$ | $n_{12}$ | $n_{1p}$ |
| $\sim T_1$ | $n_{21}$ | $n_{22}$ | $n_{2p}$ |
| | $n_{p1}$ | $n_{p2}$ | $n_{pp}$ |

FIGURA 2 – Matriz de contigência 2 x 2.
Fonte: Pedersen (2000) Adaptada.

O conteúdo da matriz é obtido através da contabilização da frequência de ocorrência dos termos representados por $T_1$ e $T_2$ e pela não ocorrência representado por $\sim T_1$ e $\sim T_2$, sendo:

- $n_{11}$ é o número de vezes que os termos $T_1$ e $T_2$ ocorrem juntos formando um bigrama;
- $n_{12}$ é o número de vezes que o termo $T_1$ ocorre com alguma outra palavra diferente do termo $T_2$;
- $n_{21}$ é o número de vezes que o termo $T_2$ ocorre com alguma outra palavra diferente do termo $T_1$;
- $n_{22}$ é o número de vezes que os termos $T_1$ e $T_2$ não ocorrem nos bigramas;
- $n_{p1}$ é o número total de vezes que o termo $T_1$ ocorre como sendo a palavra mais à esquerda do bigrama.
- $n_{1p}$ é o número total de vezes que o termo $T_2$ ocorre como sendo a palavra mais à direita do bigrama.

Os valores das células internas são calculados através do produto de suas associações marginais dividido pelo tamanho da amostra, através das expressões mostradas em (2.2).

$$m_{11} = \frac{n_{p1} * n_{1p}}{n_{pp}} \qquad m_{21} = \frac{n_{p1} * n_{2p}}{n_{pp}}$$
$$m_{12} = \frac{n_{p2} * n_{1p}}{n_{pp}} \qquad m_{22} = \frac{n_{p2} * n_{2p}}{n_{pp}} \tag{2.2}$$

A seguir, far-se-á uma descrição dessas medidas implementadas pelo pacote NSP e descritas por Pedersen et al. (2000), as quais utilizam os valores da matriz de contingência apresentados anteriormente. Informações complementares sobre a utilização e o funcionamento do NSP estão descritos na seção 3.3 do capítulo de metodologia.

### 2.4.1 Log-likelihood Ratio

A medida Log-likelihood Ratio (ll) é uma medida de desvio entre os dados observados e o que deve ser esperado se os termos do bigrama forem independentes. Um *score* alto indica que existe uma menor evidência em favor de concluir que as palavras são independentes.

Para se chegar ao valor do log-likelihood, calcula-se a soma dos desvios observados entre os valores observados e esperados de cada célula da matriz, conforme apresentado na expressão mostrada em (2.3).

$$ll = 2 * \left[ n_{11} * Log\left(\frac{n_{11}}{m_{11}}\right) + n_{12} * Log\left(\frac{n_{12}}{m_{12}}\right) + n_{21} * Log\left(\frac{n_{21}}{m_{21}}\right) + n_{22} * Log\left(\frac{n_{22}}{m_{22}}\right) \right] \tag{2.3}$$

### 2.4.2 Pointwise Mutual Information

A medida Pointwise Mutual Information (pmi) é definida como sendo o logaritmo do desvio entre a frequência observada de um bigrama pela probabilidade dele ser independente. É calculada pela expressão mostrada em (2.4).

$$pmi = \log\left(\frac{n_{11}}{m_{11}}\right) \tag{2.4}$$

O pmi tende a sobre-estimar os bigramas com baixa frequência de observação. Para prevenir que isso ocorra, uma variação de pmi é usada, a qual incrementa a influência da frequência observada, através da introdução de um expoente (exp) no numerador da fórmula conforme mostrado na expressão (2.5).

$$pmi = \log\left(\frac{n_{11}^{\exp}}{m_{11}}\right) \tag{2.5}$$

### 2.4.3 Mutual Information

A medida True Mutual Information (tmi) é definida como sendo o peso médio da pmi para todos os pares de valor observado e esperado para cada célula da matriz de contingência, dado pela expressão mostrada em (2.6).

$$tmi = \left[\frac{n_{11}}{n_{pp}} * Log\left(\frac{n_{11}}{m_{11}}\right) + \frac{n_{12}}{n_{pp}} * Log\left(\frac{n_{12}}{m_{12}}\right) + \frac{n_{21}}{n_{pp}} * Log\left(\frac{n_{21}}{m_{21}}\right) + \frac{n_{22}}{n_{pp}} * Log\left(\frac{n_{22}}{m_{22}}\right)\right] \tag{2.6}$$

Onde pmi é calculado pela expressão mostrada em (2.7):

$$pmi = \log\left(\frac{n_{11}}{m_{11}}\right) \tag{2.7}$$

### 2.4.4 Poisson Stirling

A medida Poisson Stirling é um logaritmo negativo de aproximação da medida Poisson-likelihood. Ela é usada na fórmula de Stirling para uma aproximação fatorial da medida Poisson-likelihood, conforme mostrada na expressão (2.8).

$$Poisson\_Stirling = n_{11} * \left(\log\left(\frac{n_{11}}{m_{11}}\right) - 1\right) \tag{2.8}$$

### 2.4.5 Fisher exact test – Left Sided

O teste exato de Fisher é calculado através da fixação dos totais marginais e das probabilidades de cálculo hipergeométrico para todos os possíveis valores da tabela de contingência. Calculado pela expressão mostrada em (2.9).

$$prob\_hipergeométrica = \frac{(n_{1p})!(n_{2p})!(n_{p1})!(n_{p2})!}{(n_{11})!(n_{12})!(n_{21})!(n_{22})!(n_{pp})!} \quad (2.9)$$

O teste Left Fisher é calculado pela soma das probabilidades de todos as possíveis combinações par a par da tabela de contingência formada pela fixação dos totais marginais e alterando o valor do $n_{11}$ para menor do que um dado valor. O teste exato de Left Fisher diz quão provável é, aleatoriamente, uma amostra da tabela em que $n_{11}$ é menor do que o observado. Em outras palavras, ele nos diz quão provável é a amostra de uma observação, em que as duas palavras são menos dependentes do que o atualmente observado.

### 2.4.6 Fisher exact test – Right Sided

O teste Right Fisher é calculado pela soma das probabilidades de todos os possíveis combinações par a par da tabela de contingência formada pela fixação do totais marginais e alterando o valor do $n_{11}$ para maior ou igual do que um dado valor. O teste exato de Right Fisher diz quão provável é, aleatoriamente, uma amostra da tabela em que $n_{11}$ é maior do que o observado. Em outras palavras, ele diz quão provável é a amostra de uma observação, em que as duas palavras são mais dependentes do que o atualmente observado.

### 2.4.7 Two-tailled Fisher

O teste Two-tailled Fisher é um procedimento estatístico, usado para comparar a hipótese nula de que um parâmetro da população é igual a um valor particular, contra a hipótese alternativa que o parâmetro da população é diferente a partir deste valor. As evidências sobre a hipótese nula é obtido a partir de uma estatística de teste. Esse teste é dito “bicaudal” porque a sua hipótese alternativa não especifica se o parâmetro é maior (*right side*) ou menor (*left side*) que o valor especificado pela hipótese nula. Assim, ambos os valores grandes e pequenos, isto é, os valores em ambas as caudas da sua distribuição, fornecem provas contra a hipótese nula. Este tipo de teste é relevante para situações em que os investigadores desejam testar uma hipótese nula, mas não se sabe previamente sobre a direção da alternativa.

O teste Two-tailled Fisher é calculado pela soma das probabilidades de toda a tabela de contingência com probabilidades menor que a probabilidade da tabela observada.

O teste Two-tailled Fisher diz quão provável seria observar uma tabela de contingência, que é menos provável do que a tabela atual.

### 2.4.8 Phi Coeficcient

Essa medida pode ser obtida através do quadrado da formulação tradicional do coeficiente Phi, calculada pela expressão (2.10).

$$Phi^2 = \frac{\left((n_{11} * n_{22}) - (n_{21} * n_{21})\right)^2}{n_{1p} * n_{p1} * n_{p2} * n_{2p}} \tag{2.10}$$

Os autores utilizam o valor de $Phi^2$ como equivalente ao teste Person's Chi-Squared multiplicado pelo tamanho da amostra, por ser mais apropriado para a identificação de EM, sendo calculado pela expressão (2.11).

$$Chi\_squared = n_{pp} * Phi^2 \tag{2.11}$$

### 2.4.9 T-Score

A medida T-Score é definida como uma relação da diferença entre o valor observado e a média do valor esperado para a variância da amostra. Essa formulação é uma variante do t-teste padrão que foi proposto para uso na identificação de EM em grandes amostras de texto. É calculada pela expressão mostrada em (2.12).

$$Tscore = \frac{n_{11} - m_{11}}{\sqrt{n_{11}}} \tag{2.12}$$

### 2.4.10 Pearson Chi-Square Test

O teste de Pearson Chi-Squared mede o desvio entre o valor observado e o esperado entre dois termos, com objetivo de verificar se eles são independentes, ou não. Quanto maior a pontuação, menor evidência existe em favor da conclusão de que as palavras são independentes. É calculado pela expressão mostrada em (2.13).

$$Chi\_Squared = 2 * \left[ \left( \frac{n_{11} - m_{11}}{m_{11}} \right)^2 + \left( \frac{n_{12} - m_{12}}{m_{12}} \right)^2 + \left( \frac{n_{21} - m_{21}}{m_{21}} \right)^2 + \left( \frac{n_{22} - m_{22}}{m_{22}} \right)^2 \right] \quad (2.13)$$

### 2.4.11 Dice Coeficcient

O coeficiente Dice é uma medida que pode ser utilizada para avaliar a proporção em que os termos de um conjunto de palavras co-ocorrem. O tamanho da interseção entre os conjuntos é normalizada pela média do tamanho dos conjuntos envolvidos. O valor pode ser calculado pela expressão mostrada em (2.14).

$$Dice = \frac{2 * n_{11}}{n_{p1} + n_{1p}} \quad (2.14)$$

### 2.4.12 Jaccard Coeficcient

O coeficiente Jaccard pode ser calculado a partir de uma transformação do coeficiente Dice, conforme mostrado na expressão (2.15).

$$Jaccard = \frac{Dice}{2 - Dice} \quad (2.15)$$

Ou, pela quantidade dos termos que formam um bigrama (interseção), divididos pela soma da quantidade dos termos que aparecem juntos (união). Calculados através da expressão mostrada em (2.16).

$$Jaccard = \frac{n_{11}}{n_{11} + n_{12} + n_{21}} \quad (2.16)$$

### 2.4.13 Odds Ratio

O Odds Ratio calcula a relação entre o número de vezes que as palavras de um bigrama ocorrem em conjunto, pelo número de vezes que as palavras ocorrem separadamente. É um cruzamento do produto da diagonal e dos elementos que não pertencem à diagonal. Ou, como é comumente conhecida, a razão dos produtos cruzados. Se $n_{12}$ ou $n_{21}$ forem iguais a zero, eles serão “suavizado” para evitar denominador com valor zerado. É calculada pela expressão mostrada em (2.17).

$$OddsRatio = \frac{n_{11} * n_{22}}{n_{12} * n_{21}} \quad (2.17)$$

## 2.5 Sistemas de Recuperação da Informação

Os SRI são um importante fundamento deste trabalho. Entretanto, antes de apresentar algumas das definições encontradas na literatura sobre o tema, faz-se necessário apresentar uma breve discussão sobre o significado de dados e informação. Dado é considerado como sendo uma sequência de símbolos que podem ser codificados, interpretados e manipulados por programas de computadores, assim como transportados em redes e dispositivos de comunicação. Por outro lado, a informação carrega um grau maior de abstração, não dispensando o sujeito que se apropria dela a partir dos dados. Essa apropriação está relacionada com a interpretação. Embora haja distinção, na literatura entre os termos dados e informação, na prática, os SRI recuperam documentos, coleção de dados, nos quais os termos da busca foram parcial ou integralmente encontrados, dependendo do método empregado no processo de recuperação.

A representação lógica dos documentos em um SRI pode ser feita a partir de todos os termos do documento, ou considerando-se apenas termos selecionados por especialistas humanos chamados de vocabulário controlado. Nas primeiras implementações dos SRI, por limitações computacionais, utilizava-se um conjunto menor de palavras selecionadas por especialistas humanos. Nesse caso, produzia-se uma visão lógica mais concisa dos documentos, essa forma pode levar a uma recuperação de informação de baixa qualidade. Diversas outras formas intermediárias foram sendo adotadas na tentativa de melhorar a precisão das buscas. Com o aumento da capacidade de processamento e memória dos computadores atuais é possível a representação de um documento pelo seu conjunto completo de palavras, é a chamada representação total do texto.

A seguir são apresentados alguns dos fundamentos conceituais empregados nas implementações dos SRI, considerando-se desde os processos manuais até os automatizados.

### 2.5.1 O processo de indexação manual

As primeiras formas utilizadas pelos bibliotecários para recuperar conteúdos foram proporcionadas pelas técnicas de classificação através da codificação apoiada em linguagens documentárias. Essas linguagens através de uma gramática com regras e

instruções bem definidas orientam o trabalho do indexador para extrair os descritores que melhor descrevem o conteúdo do documento. Nesse sentido, a linguagem documentária concretiza a capacidade simbólica do homem, através da organização de seus termos e regras em um sistema próprio. O processo de classificação dos documentos, realizado de forma manual por especialistas, é um processo intelectual exaustivo, pois demanda a leitura de cada texto pelo indexador a fim de selecionar as palavras-chave ou descritores, que representem o documento numa forma compatível com uma dada linguagem documentária.

Segundo Cintra (1983, p. 6), existem dois procedimentos básicos para a depreensão dos descritores: a apreensão instantânea das unidades de informação e a apreensão por análise. Nesse sentido, devem ser observadas pelo indexador as partes relevantes do texto tais como: título, resumo, introdução e conclusão, os termos que possuem maior frequência de ocorrência e que traduzem melhor a percepção do indexador no sentido do texto. A autora destaca ainda a dificuldade em expressar através da linguagem documentária as nuances decorrentes da linguagem natural tais como: a polissemia, a homonímia, a sinonímia e os modos e expressões de relações complexas. A definição desses conceitos é apresentada a seguir.

- Polissemia – Nome dado à pluralidade de sentidos de uma mesma forma. Ela pode se dar por: (1) extensão, como, por exemplo, em “estação” que pode significar: “parada”, “épocas do ano”; (2) metáfora que é um tipo de extensão, no qual atua um componente analógico. Em “serra”, por exemplo, o significado de “montes” decorreu da analogia com a ferramenta “serra”; (3) restrição, como, por exemplo, abrir (do latim *aperire*) nos deu a palavra aperitivo, que por sua vez corresponde a dois termos, restritos a dois significados: “purgativo” na linguagem médica e “beberete” na linguagem comum.
- Homonímia – Corresponde à igualdade entre significantes de significados diferentes. É, pois, o estudo das formas que apenas se diferenciam pela significação ou função, já que a estrutura fonológica é a mesma. Ela pode ser: (1) total, como em “fiar” que tanto significa “tecer”, quanto “confiar”; (2) parcial, como em coser e cozer.
- Sinonímia – Decorre de coincidência de significado entre diversas palavras. Exemplos: o significado de “mar” pode ser expresso através dos termos mar ou oceano. A utilização do sinônimo é, provavelmente, uma dificuldade mais séria para as linguagens de indexação. Basta observar que na língua portuguesa, os padrões estéticos não permitem, mesmo em linguagem científica, a constante

repetição de um mesmo termo. Isso gera no documento a presença da mesma ideia convertida em várias formas semelhantes de expressá-las.

Observados esses aspectos para escolha do descritor, deve-se ter especial atenção, nessa operação de delimitação dos significados, com a sua semântica contextual. Nesse sentido, a autora define **semântica** como a disciplina que se ocupa do sentido ou da significação dos elementos; e **sintaxe** como a disciplina que se ocupa das relações que se estabelecem a partir da organização sintagmática dos elementos; e **morfologia** como a disciplina que sintetiza parcialmente aspectos da semântica e da sintaxe, encarrega-se da identificação das partes da palavra e de suas condições de ocorrência.

Os descritores de um texto podem ser analisados em termos de: (1) os sintagmas de símbolos notacionais (números, letras, pontuação, marcas) isto é, unidades resultantes da combinação de formas menores em unidades de nível superior. (2) lexemas como combinação de fonemas, ou seja, como combinação de unidades capazes de promover a distinção entre os signos da língua. Nas linguagens de indexação, os signos, em geral, lidam com os termos de uma língua particular, fixando significados de forma a anular o sentido simbólico do signo linguístico. Dessa forma, os signos documentários exigem do indexador uma percepção do contexto para que a tradução tenha um forte poder de partilha na comunidade à qual o documento se destina.

### 2.5.2 Processo de indexação automatizado

Com a explosão informacional ocorrida nas últimas décadas, realizar o trabalho de indexação, utilizando especialistas, tornou-se inviável do ponto de vista do volume do conteúdo informacional. Para atender a essa crescente demanda, pesquisadores de diferentes linhas de pesquisa, tomando como base os aportes teóricos da linguística, biblioteconomia e da computação dentre outras veem paralelamente, ao longo das últimas décadas, realizando pesquisas que buscam automatizar esses processos de indexação e recuperação da informação. Nesse contexto, uma forma de busca muito comum na Web é fazer a consulta através de um conjunto de palavras fornecidas de forma livre através da interface de um sistema de busca. O princípio de funcionamento do mecanismo de busca é calcular a soma do valor apurado, através do casamento cada um dos termos da consulta com os termos da coleção de documentos.

Para Lancaster & Warner (1993), os SRI consistem em uma interface entre uma coleção de informações, em meio impresso ou não, e uma população de usuários. Cabendo aos SRI as seguintes tarefas: aquisição e armazenamento; organização e controle; e

distribuição e disseminação aos usuários. Baeza_Yates e Ribeiro-Neto (1999, p.2-3) apontam os bibliotecários e os especialistas em lidar com informações, como sendo os primeiros interessados na utilização dos SRI em domínios específicos, tais como as bibliotecas digitais e os centros de documentação. Atualmente podemos considerar que os SRI mais conhecidos são os mecanismos de busca na *World Wide Web*.

Manning e Schütze (2003, p. 529-531) consideram que a pesquisa em Recuperação da Informação (RI) lida com o desenvolvimento de algoritmos e modelos para recuperar informações a partir de um repositório de documentos. Os autores enquadram a RI como um subcampo do PLN, destacando, como sendo o seu problema clássico, a recuperação *ad hoc,* em que o usuário entra com uma consulta que descreve a informação desejada e o SRI retorna uma lista de documentos obtidos como resposta. Ainda, segundo esses mesmos autores, existem dois principais modelos de SRI: os sistemas de correspondência exata, aqueles os quais retornam documentos que satisfaçam precisamente uma expressão de consulta; e os sistemas que recuperam os documentos de acordo com a sua estimativa de relevância para a consulta.

Russell & Norvig (2004, p. 813) definem o processo de RI como sendo a tarefa de encontrar documentos relevantes que atendam às necessidades de informação de um usuário. Esses autores afirmam que os SRI se caracterizam por:

- Definir o que é o documento a ser recuperado: um parágrafo, uma página, ou um texto completo;
- Definir a forma de consulta: uma lista de palavras, uma sequência de palavras adjacentes em forma de uma frase ou parte de uma frase, se pode conter operadores booleanos e não-booleanos;
- Definir o subconjunto de respostas: as relevantes e as não-relevantes;
- Definir a forma de apresentação do resultado: uma lista ordenada de documentos, um mapa giratório com os resultados apresentados num espaço tridimensional.

Para ter-se uma visão geral dos diferentes tipos de SRI recorre-se à taxonomia apresentada por Baeza-Yates & Ribeiro-Neto (1999, p.21) que pode ser visualizada na Figura 3.

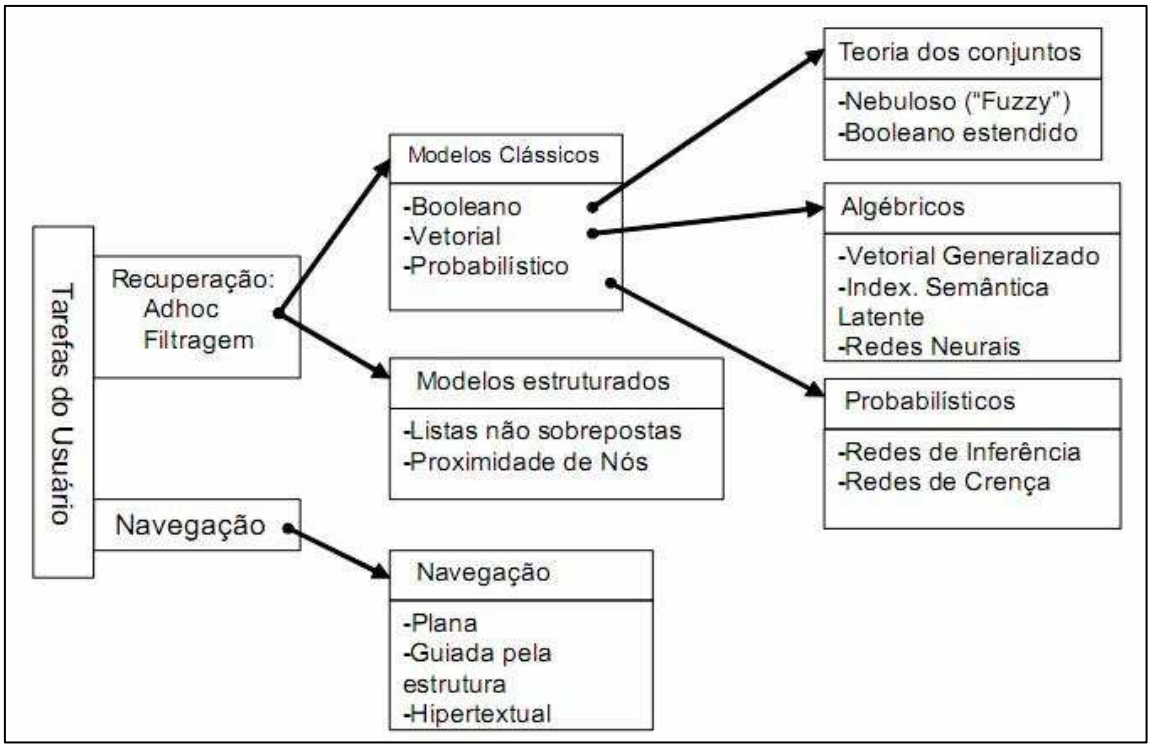



FIGURA 3 – Taxonomia dos SRI.
Fonte: Souza (2007) adaptado de Baeza-Yates & Ribeiro-Neto, 1999.

Segundo esses autores, a demanda dos usuários por informações pode ser através de:

- Recuperação a*d hoc* – quando o acervo de documentos sofre poucas alterações enquanto novas consultas são submetidas aos sistemas;
- Recuperação por filtragem – nesse caso, existe uma consulta estabelecida que monitora a adição de novos documentos. Em intervalos de tempo critérios de filtragem são aplicados à coleção de documentos disparando alertas para o usuário proponente da consulta;
- Navegação – nesse caso, ocorre a busca hipertextual, em que o usuário usa um sistema de pesquisa sem preconceber uma consulta. Normalmente, não existe necessariamente uma indexação prévia.

Como a proposta desta tese é realizar buscas em uma coleção de documentos estável, a base técnica adotada pela ferramenta de SRI elaborada é a recuperação *ad hoc*. Desse modo, esse é um recorte teórico que se adotará. Entretanto, antes de detalhar essa forma de recuperar informações, é necessário entender como funciona o processo de preparação dos documentos para que a RI ocorra. Essa preparação pode sofrer algumas adaptações dependendo do modelo a ser implementado, mas, em geral, todas passam pelas etapas descritas por Baeza_Yates e Ribeiro-Neto (1999, p.163-175). Dentre essas

etapas, algumas são obrigatórias e outras opcionais no processo de indexação dos termos para permitir a busca. A Figura 4 apresenta um detalhamento desse processo.

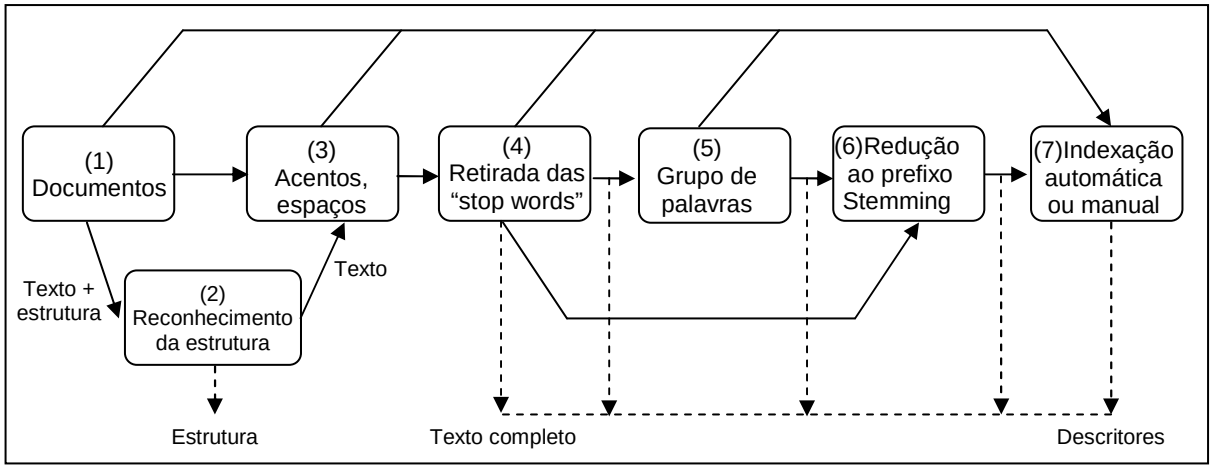



FIGURA 4 – Fases do pré-processamento dos documentos.
Fonte: Adaptada de Baeza-Yates & Ribeiro-Neto (1999, p. 166).

O objetivo desse processo é preparar previamente a coleção de documentos de tal forma a viabilizar a consulta. Considerando que, em cada retângulo, é realizada uma etapa do processo, a seguir apresenta-se uma descrição de cada uma delas. Na etapa (1), cabe definir qual será a representação lógica dos documentos contidos no *corpus*. Ou seja, quais informações serão utilizadas para representar um documento podendo ser: o subconjunto de seus termos, aqueles que melhor representam o seu conteúdo geralmente, substantivos, verbos ou o conjunto completo dos termos que aparecem no documento. Adicionalmente, na etapa (2), pode-se ainda agregar informações sobre a estrutura do documento, tais como: separação dos parágrafos e das sentenças, títulos, caracteres em maiúsculas ou minúsculas, fonte (negrito, itálico, etc). Na etapa (3), é realizada uma padronização dos termos transformando toda a cadeia de caracteres do texto, normalmente em letras minúsculas, e retirando todos os ruídos tais como: caracteres de acentuação, caracteres especiais de controle e de formatação. Na etapa (4), opcionalmente, pode ser utilizada para retirar as palavras que ocorrem com muita frequência em todos os documentos as "*stop words*", tais como: preposições, artigos, etc. Na etapa (5), opcionalmente, pode ser utilizada para fazer o agrupamento de palavras e estruturas sintáticas e gramaticais através do uso de sintagmas nominais ou Lematização[11] . Na etapa (6), opcionalmente, pode ser utilizada para fazer a Radicalização[12] dos termos. E finalmente, na etapa (7), os termos são organizados em uma estrutura de lista invertida, em que cada termo distinto encontrado no *corpus* faz referência aos documentos nos quais ele foi encontrado.

[11] O termo lematização é originalmente conhecido em inglês por *lemmatizer.*
[12] O termo radicalização é originalmente conhecido em inglês por *stemmer.*

Cabe ainda enfatizar que as técnicas de Lematização e a Radicalização, citadas anteriormente, visam reduzir a demanda necessária de recursos computacionais para representação do léxico da coleção dos documentos e que também possibilitam expandir a busca. Na medida em que um termo da consulta pode ser considerado similar a não apenas um, mas a vários termos que possuem o mesmo radical, cabe ressaltar que essas técnicas são dependentes do idioma. Silva (2007) faz uma clara distinção entre esses termos e apresenta os seus conceitos. Ambas as técnicas têm como meta reduzir as palavras ao seu radical. Entretanto, a forma de como a redução da palavra se processa é totalmente diferente. Na lematização, as palavras flexionadas são reduzidas ao seu lema. O processo se dá pela análise morfológica da palavra. Para o processamento desse algoritmo existe a necessidade de se fazer uma etiquetação prévia das palavras, pois a lematização depende do conhecimento prévio da classe gramatical da palavra. A técnica consiste em, dependendo da classe gramatical, aplicar regras para reduzir as palavras em: gênero e número (ex: belo, belos, bela, belas para belo); verbo reduz todas as conjugações para o infinitivo; retirada dos diminutivos e superlativos (ex: pobrezinho, homenzinho para pobre, homem), etc. Por outro lado, na radicalização, em vez de se ter no índice todas as formas da palavra flexionadas, faz-se a redução ao seu radical indexando apenas este. Desse modo, o processo de radicalização trabalha com heurísticas que visam encontrar a divisão silábica das palavras oxítonas, paroxítonas e proparoxítonas e aplicar a elas um conjunto de regras e exceções a fim de promover cortes nos sufixos das palavras. No entanto, existe uma diferença importante que deve ser ressaltada entre essas duas técnicas. Enquanto que na lematização as palavras geradas existem no idioma, na radicalização a palavra resultante não necessariamente existe no idioma. Nessa pesquisa, opta-se pela não utilização dessas técnicas de compressão.

Entretanto, a indexação completa incorpora nos índices muitos termos irrelevantes para busca. São as chamados *stop words* definida por Souza (2006 p. 50) como sendo “[...] palavras que para um dado idioma, apresentam baixo conteúdo informacional, sendo irrelevantes como descritores, e usualmente eliminados dos índices”. A remoção das *stop words* tem como objetivo principal eliminar termos que não são representativos dos documentos. Nesse contexto, “não representativos” significa dizer que os termos têm baixa capacidade de discriminação entre os documentos e baixo conteúdo informacional. Essa operação também pode ser considerada como uma técnica de compressão de textos, pois a eliminação das *stop words* reduz o número de palavras a serem analisadas no documento e também o número de palavras a serem armazenadas no dicionário de palavras.

Todo esse pré-processamento de identificação dos termos que representam a coleção de documentos culmina com a criação de uma lista invertida. Desse modo, cada termo selecionado para compor o vocabulário que representa a coleção de documentos possui uma referência, em forma de lista, com todos os documentos que o contem. Somente após, concluído esse pré-processamento é que pode ocorrer o casamento do conjunto de palavras-chave fornecido pelo usuário com os termos existentes na lista invertida. O processamento realizado pelo casamento dos termos pode ser total (booleano) ou parcial (ponderado pela relevância). Portanto, retoma-se o tema da recuperação *ad hoc* apresentada por Baeza-Yates & Ribeiro-Neto (1999, p.21), a fim de apresentar os modelos descritos pelos autores dentro desse contexto:

- Clássico – cada documento é descrito por um conjunto de palavras-chave representativas;
- Estruturado – aquele em que, além das palavras-chave, permite ao usuário especificar a busca em partes específicas da estrutura do documento (por exemplo, buscar uma palavra quando essa aparece no subtítulo de uma figura).

Os modelos clássicos são subdivididos em:

- Booleano – para cada consulta são recuperados todos os documentos que possuem os termos nas condições especificadas pelo usuário;
- Vetorial – utiliza uma representação *n*-dimensional em que todos os termos presentes na consulta são comparados ao conjunto de todos os documentos representados no espaço vetorial constituído pelos termos da coleção.
- Probabilísticos – supõe-se que exista um conjunto ideal de documentos para uma consulta e a recuperação desse conjunto ocorre através de interações sucessivas.

Os modelos estruturados são subdivididos em:

- Listas não sobrepostas: a ideia desse modelo é construir diversas listas de indexação de modo a refletir, por exemplo, os níveis hierárquicos do texto. Cada elemento de uma lista representa uma área do texto;
- Proximidade de nós: esse modelo trabalha com um índice para mapear a ocorrência das palavras e um outro para identificar as características estruturais. Essa estrutura hierárquica possibilita verificar se todos os termos procurados estão ligados hierarquicamente a um mesmo nó. Desse modo, todo conteúdo

hierarquicamente ligado ao nó comum dos termos da busca é considerado como resposta válida para a consulta.

Uma outra abordagem que pode ser empregada na definição de critérios de relevância de um documento é ordenálos de forma antecedente à consulta. Isto é, atribuir um fator de qualidade ao documento, o que representa a chance de ele ser relevante a qualquer consulta. Dentre os elementos que poderão compor esse fator, podem-se citar: o número de referências que apontam para o documento conhecido como Page Rank[13], o tempo de existência, a fonte de origem, dentre outros.

Tomando como base o amplo traballho de pesquisa realizado por Zhai (2008), no qual ele apresenta uma revisão crítica dos diversos métodos adotados para o processo de RI, acrescenta-se à categorização apresentada por Baeza-Yates & Ribeiro-Neto (1999); Manning e Schütze (2003), um outro modelo. Esse modelo se caracteriza por, em vez da função de pontuação buscar o casamento exato do descritor no documento, ela permite o casamento por aproximação semântica. Dessa forma, esse método permite a busca por sinônimos ou multilingue. Este modelo de tradução pode ser entendido, imaginando um usuário que deseja o documento iria formular uma consulta, mas internamente ela seria executada em duas etapas. Na primeira, o sistema buscaria a palavra exata no documento, na segunda, a palavra seria traduzida em uma outra diferente, mas semanticamente relacionada. Um grande desafio aqui é como obter o modelo de tradução. Para isso duas abordagens foram propostas: treinar os sistema com dados, o que não é uma tarefa fácil; ou utilizar um método heurístico para gerar alguns pares sintéticos para a consulta de documentos para treinar o modelo de tradução. Esses métodos têm mostrado melhora no desempenho de recuperação em relação à consulta exata. Mas ainda existem desafios na utilização deles na prática, por exemplo, em como melhorar a eficiência do cálculo do *score* que tem de considerar muitas outras palavras, para casamentos possíveis com cada palavra da consulta.

Ainda, segundo Zhai (2008), é evidente que a acurácia de um SRI está diretamente relacionada com a função de cálculo de relevância adotada, e esse tem sido o maior desafio de pesquisa da área de RI. Nesse sentido, destaca-se que uma extensão natural do método básico de consulta está em ir além dos modelos tradicionais, que fazem o casamento de palavras-chave de forma independente. Ou seja, modelos que verificam as ocorrências das palavras (unigramas) considerando-as de forma completamente independente uma das outras. Como alternativa  surgem os modelos que podem capturar alguma dependência

[13] Page Rank é um algoritmo de análise de link utilizado pelo Google, que atribui um peso para uma página de acordo com o número de referência que aponta para ela.

entre as palavras da busca, ao considerá-las como sendo *n*-gramas. Um modelo de busca por *n*-gramas visa identificar a relação de dependência entre as palavras que o compõem. Isso pode, potencialmente, capturar a dependência de palavras adjacentes, ou mesmo palavras com uma distância de até um limiar definido previamente. Tais modelos de *n*-gramas capturam a dependência com base nas posições das palavras nas sentenças que formam o texto. Adicionalmente, esse mesmo autor apresenta uma outra linha de abordagem que busca capturar a dependência entre as palavras com base na estrutura da gramática. Em todas essas abordagens, a fórmula de recuperação se resume a uma combinação de atribuição de notas maiores para as unidades correspondentes (*n*-gramas), do que para as palavras isoladas. Embora essas abordagens tenham benefícios, principalmente relacionados à captura da dependência entre as palavras, essa melhoria tende a ser pequena, e uma das razões para esses resultados pode estar ligada ao fato de que à medida que se avança para modelos mais complexos, os dados se tornam ainda mais escassos, o que dificulta a obtenção de uma estimativa exata do modelo. Apesar de a melhora dos resultados obtidos pelas busca que utilizam *n*-gramas ter sido modesta, em trabalhos precedentes a este, esse é o foco deste trabalho. Afinal, acredita-se que para o objetivo específico proposto e através da utilização de uma combinação de técnicas será possível obter ganhos no processo de RI, através do uso de *n*-gramas dependentes, as EM.

#### 2.5.2.1 Modelo booleano

O modelo booleano é baseado na lógica de conjuntos. Ele foi largamente utilizado nos primeiros SRI, dado a sua simplicidade de implementação em sistemas informatizados. Mas mostrou-se ineficiente para as buscas em grandes e heterogêneas coleções de documentos. Isto se deu em função de usualmente produzir respostas ora vazias, ora muito extensas. Isto ocorre, pois essa técnica se baseia no resultado de uma expressão booleana binária aplicada a cada documento da coleção, que pode obter como resposta um resultado verdadeiro ou falso, representando respectivamente a existência ou não existência do termo. Como no subconjunto de respostas verdadeiras, todos os documentos têm a mesma relevância não é possível ordenar o resultado por esse critério, afinal não existe o casamento parcial entre a consulta e o documento. Desse modo, todos os documentos que têm os mesmos termos são igualmente relevantes para a consulta. Além disso, existe a necessidade de a consulta do usuário ser expressa em uma notação boolena, a qual pode gerar dificuldades para usuários leigos. Visando contornar esse problema, outras técnicas de casamento parcial foram criadas a fim de produzir respostas ordenadas por relevância. E são essas que interessa enfatizar, por ser essa a técnica empregada na ferramenta elaborada para realização da parte empírica desta tese.

### 2.5.2.2 Modelo probabilístico

Muitos SRI utilizam modelos estatísticos, em que dada uma consulta, o objetivo é encontrar documentos que terão a maior probabilidade de serem relevantes. Para avaliar a probabilidade, primeiro é necessário definir uma forma de representar os documentos em um modelo de palavras (unigramas), também conhecido como "saco de palavras" do inglês *bag of words*. Nesse modelo, conforme descrevem Manning e Schütze (2003, p.142), o que importa na consulta é a frequência das palavras no documento e não se elas se encontram na mesma ordem da requisição. Ou seja, as expressões "Maria é mais rápida do que João" e "João é mais rápido do que Maria", embora tenham significados distintos para efeito da busca têm o mesmo efeito. Esse modelo se caracteriza por não considerar a relação de dependência dos termos utilizados na busca. Ele é conhecido como sendo o modelo bayesiano ingênuo, em que o resultado final da probabilidade de consulta de cada documento é obtido multiplicando-se a probabilidade de cada palavra encontrada, dado pela expressão mostrada em (2.18):

$$\boxed{P(Q \mid D, r) = \prod_j P(Q_j \mid D, r)} \qquad (2.18)$$

Onde: *P* é a probabilidade, *Q* é a consulta, *D* é o documento, *r* denota resultado verdadeiro e $Q_j$ indica a *j*-ésima palavra na consulta.

### 2.5.2.3 Modelo Vetorial

Uma outra abordagem utilizada para representar os termos como informação local (no nível do documento) e global (no nível da coleção de documentos) é o modelo de espaço vetorial. Manning e Schütze (2003, p.539) definem o modelo de Espaço Vetorial como sendo um dos modelos mais amplamente utilizados devido à sua simplicidade conceitual e ao uso da metáfora que relaciona proximidade espacial com a proximidade semântica. Essa abordagem permite um casamento parcial delineado através dos pesos que expressam o grau de similaridade entre a consulta e os documentos. Cada documento é representado como um vetor de termos, sendo que cada um deles possui um peso associado que indica o seu grau de importância no documento. Os pesos podem ser calculados de diversas formas e servem para especificar a magnitude do vetor. Segundo Singhal et al. (1996), o cálculo do peso de relevância do documento é uma das partes mais importantes de um SRI. Esses métodos de cálculo do peso geralmente se baseiam em:

- Term Frequency (*tf*) – frequência do Termo correspondendo ao número de ocorrências do termo no documento;

- Document Frequency (*df*) – frequência do Documento correspondendo ao número de documentos da coleção em que o termo ocorre;
- Collection Frequency (*cf*) – frequência da Coleção correspondendo ao número total de ocorrências do termo na coleção.

Entretanto, para um melhor entendimento desse modelo, faz-se necessário apresentar alguns conceitos básicos que o sustentam.

O plano cartesiano é formado por duas retas perpendiculares, que representam um eixo horizontal *x* ou eixo das abscissas e um eixo vertical *y* ou eixo das ordenadas. A representação de um ponto é feita por pares ordenados com valores de *x* e de *y*. A Figura 5 mostra dois pontos $P_1$ e $P_2$ definidos pelas coordenadas $P_1(x_1, y_1)$ e $P_2(x_2, y_2)$.

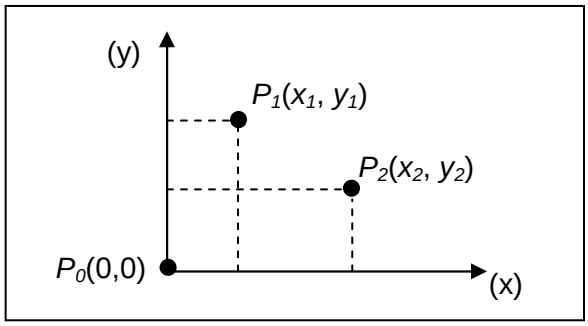



FIGURA 5 – Pontos $P_0$ , $P_1$ e $P_2$ no plano cartesiano.
Fonte: Elaborada pelo autor

O produto escalar $P_1 \bullet P_2$ pode ser obtido pelo somatório da multiplicação das coordenadas dos pontos em cada eixo, dado pela expressão (2.19).

$$\boxed{P_1 \bullet P_2 = x_1 * x_2 + y_1 * y_2} \qquad (2.19)$$

Se os pontos $P_1$ e $P_2$ estiverem definidos em três dimensões com as coordenadas $P_1(x_1, y_1, z_1)$ e $P_2(x_2, y_2, z_2)$, então o produto escalar é dado pela expressão (2.20). Sendo que se tiver um número *n* de dimensões pode-se generalizar acrescentando os produtos das coordenadas em cada nova dimensão.

$$\boxed{P_1 \bullet P_2 = x_1 * x_2 + y_1 * y_2 + z_1 * z_2} \qquad (2.20)$$

Um outro conceito importante a ser apresentado é como calcular a distância entre dois pontos num plano e em seguida projetar esses conceitos para a distância entre dois pontos num espaço *n*-dimensional. Nesse sentido, considerando $P_0$ como sendo a posição de coordenada (0, 0), tem-se $d_1$ como sendo a distância entre os pontos $P_0$ e $P_1$ e $d_2$ como sendo a distância entre $P_0$ e $P_2$, conforme mostrado na figura 6.

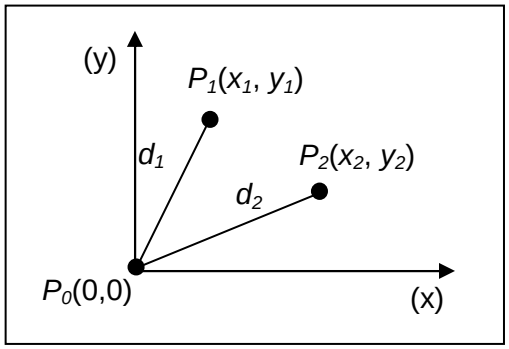



FIGURA 6 – Distâncias $d_1$ e $d_2$
Fonte: Elaborada pelo autor.

Para obter-se a magnitude da distância $d_1$ entre os pontos $P_0$ e $P_1$ e a distância $d_2$ entre os pontos $P_0$ e $P_2$, também chamada de distância Euclidiana, pode-se utilizar a expressão (2.21).

$$d_1 = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \text{ e } d_2 = \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} \quad (2.21)$$

Mas, como na coordenada $P_0$ os valores de $x$=0 e de $y$=0, obtém-se a expressão simplificada (2.22).

$$d_1 = \sqrt{(x_1)^2 + (y_1)^2} \text{ e } d_2 = \sqrt{(x_2)^2 + (y_2)^2} \quad (2.22)$$

Entretanto, como o objetivo é utilizar o modelo de Espaço Vetorial para expressar os dados da coleção, deve-se adaptar a representação das retas e substituí-las por vetores, conforme mostra a Figura 7.

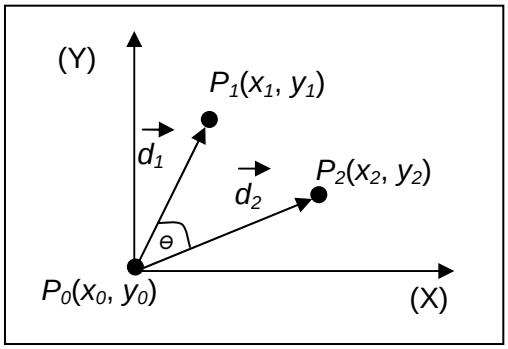



FIGURA 7 – Representação dos vetores no plano
Fonte: Elaborada pelo autor.

Como um vetor representa duas grandezas: o valor/modulo e a sua direção/sentido, o modelo considera que o peso $w$ está representado pelo seu módulo e a direção representada no eixo. Sendo que cada eixo representa um termo do espaço $n$-dimensional. Desse modo, um documento é representado por um vetor resultante da projeção de todos os seus termos no espaço $n$-dimensional. Considerando que uma consulta de usuário $q$

também pode ser representada da mesma forma, pela projeção de seus termos produzindo um vetor $\vec{q}$, quanto maior for o grau de similaridade entre os vetores resultantes da consulta com o documento, maior será a proximidade dos vetores projetados no espaço *n*-dimensional que representam essas duas grandezas.

Conforme apresentado por Manning e Schütze (2003, p.146), esse modelo quantifica a similaridade entre dois vetores resultantes pela projeção de seus termos no espaço *n*-dimensional considerando os pesos. Sendo que o peso representa a importância que cada termo tem para o documento. Entretanto, essa medida apresenta uma desvantagem por representar dois documentos muito similares através de vetores resultantes que podem ser significativamente bem diferentes, simplesmente porque um documento é muito maior do que o outro. Isso ocorre, pois mesmo com uma idêntica distribuição em ambos os documentos, os valores absolutos do peso podem ser distintos. Portanto, para compensar esses efeitos do tamanho do documento é necessário fazer uma ponderação pelo tamanho do documento. Para padronizar o valor do coeficiente calculado é utilizada a técnica Vector Space Model também conhecida como Cosine Similarity Vector (CSV). Ou seja, para normalizar o valor calculado da similaridade entre os dois vetores utiliza-se o resultado obtido pelo coseno do ângulo Θ formado pela interseção dos vetores resultantes das projeções, da consulta e do documento, no espaço *n*-dimensional. Esses valores calculados variarão entre 0 e 1. Desse modo, quando o ângulo entre os vetores for se aproximando de zero, indica que o resultado do coseno aproxima-se do valor um, representando máxima similaridade. Por outro lado, se o ângulo entre os dois vetores aproxima-se de 90 graus, o coseno tende ao valor zero, desse modo não há similaridade. A Figura 8 apresenta um esboço da representação do modelo de espaço vetorial considerando apenas $n = 3$. Ou seja, é uma simplificação, em três dimensões, para que possa ser representado através de um desenho. Os termos que compõem o dicionário estão representados nos eixos $T_1$, $T_2$ e $T_3$. Os pontos $w_1$, $w_2$ e $w_3$ mostrados nos eixos $T_1$, $T_2$ e $T_3$ correspondem repectivamente à magnitude dos pesos de cada um desses termos para o documento $d_1$. Os pontos $w_4$, $w_5$ e $w_6$ mostrados nos eixos $T_1$, $T_2$ e $T_3$ correspondem respectivamente à magnitude dos pesos de cada um desses termos para o documento $d_2$. Os vetores $d_1$ e $d_2$ representam a projeção resultante dos termos dos documentos. O *q* representa o vetor resultante da projeção dos termos da consulta. Desse modo, os cosenos dos ângulos $\Theta_1$ e $\Theta_2$ representam a similaridade entre o documento $d_1$ e a consulta *q* e entre o documento $d_2$ e a consulta *q* respectivamente.

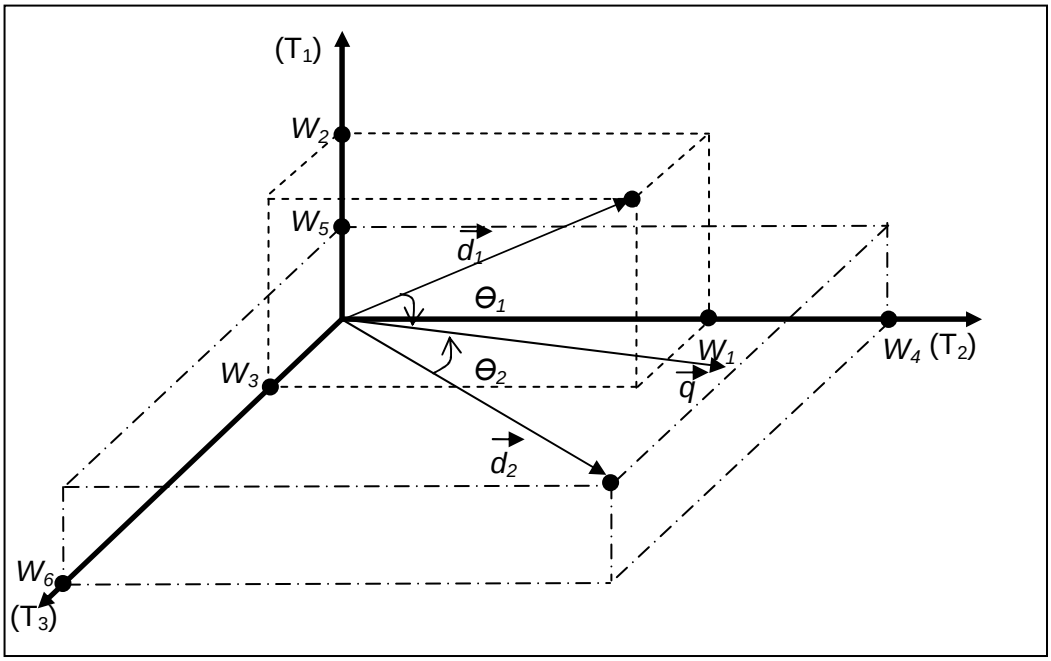



FIGURA 8 – Representação do *cosine similarity vector*
Fonte: Elaborada pelo autor.

Se se considerar que o ângulo $\Theta_1$ é menor que $\Theta_2$ significa dizer que o documento $d_1$ é mais relevante para a consulta $q$ do que o documento $d_2$. A expressão para calcular essa medida de similaridade é mostrada em (2.23).

$$Sim(\vec{d}_1, \vec{d}_2) = \cos eno\Phi = \frac{P_1 \bullet P_2}{|P_1||P_2|} = \frac{x_1 * x_2 + y_1 * y_2}{\sqrt{(x_1)^2 + (y_1)^2} * \sqrt{(x_2)^2 + (y_2)^2}} \qquad (2.23)$$

Ou seja, a medida de similaridade mostra o quanto os vetores que representam os documentos se aproximam do vetor que representa a consulta. Conforme pode ser visto na expressão (2.23), o valor da similaridade pode ser obtido pela razão entre o produto escalar dos vetores pelo produto da raiz quadrada das coordenadas em cada dimensão ao quadrado. Para exemplificar, atribuir-se-ão valores ao exemplo mostrado na Figura 8, cuja representação lógica dos documentos é composta por dois documentos, sendo que cada um deles contém três termos. Considerando o peso como sendo a frequência do termo no documento e a consulta $q$ como sendo a busca pelos termos $t_1$ = ciencia *e* $t_2$ = informacao. Como o termo $t_3$ = computacao, não está sendo consultado ao peso da consulta nesse eixo foi atribuído o valor zero. A Figura 9 apresenta esse conjunto de valores.

| Frequência dos termos nos documentos | | | | |
|---|---|---|---|---|
| Termos/ | T1 | T2 | T3 | Cálculo |
| Documentos | ciencia | informacao | computacao | Coseno |
| D1 | 4 | 2 | 3 | |
| *(q, d1)* | 1 | 1 | 0 | 0,787839 |
| D2 | 6 | 1 | 8 | |
| *(q, d2 )* | 1 | 1 | 0 | 0,492518 |

FIGURA 9 – Cálculo CSV da consulta *q* nos documentos $d_1$ e $d_2$.
Fonte: Elaborada pelo autor.

Portanto, a representação lógica desses documentos mostra que o léxico é composto por três termos em cada documento. Desse modo, utiliza-se o peso *w* como sendo a frequência do termo e no cálculo do CSV conforme mostrado na expressão (2.24) para o documento $d_1$ e na (2.25) para o documento $d_2$.

$$Sim(\vec{q},\vec{d}_1) = \frac{4*1+2*1+3*0}{\sqrt{4^2+2^2+3^2}*\sqrt{1^2+1^2+0^2}} = 0.787839 \quad (2.24)$$

$$Sim(\vec{q},\vec{d}_2) = \frac{6*1+1*1+8*0}{\sqrt{6^2+1^2+8^2}*\sqrt{1^2+1^2+0^2}} = 0,492518 \quad (2.25)$$

Para apurar a similaridade deve-se calcular o coseno do ângulo formado entre a consulta com cada um dos vetores resultantes, que representam cada um dos documentos da coleção e classificar o resultado em ordem decrescente. Para fazer isso, precisa-se construir um espaço *n* dimensional de termos, onde cada eixo representa um único termo do dicionário e o valor do peso é atribuído para cada documento que possuir o termo. Desse modo, o espaco *n*-dimensional terá o valor de *n* correspondente à quantidade de termos distintos extraidos do *corpus.*

O cálculo da similaridade também pode ser realizado através da expressão (2.26).

$$Similaridade(Q,D) = \frac{\sum_{k=1}^{n} w_{qd} * w_{dk}}{\sqrt{\sum_{k=1}^{n} (w_{qk})^2 * \sum_{k=1}^{n} (w_{dk})^2}} \quad (2.26)$$

Onde:

- *Q* representa o vetor de bigramas da consulta;
- *D representa o* vetor de bigramas de cada documento;
- $W_{qk}$ peso do bigrama *k* para a consulta;
- $W_{dk}$ peso do bigrama *k* para o documento.

Entretanto, no caso particular da busca realizada por documentos similares através de uma consulta por palavras-chave, os pesos dos termos da consulta são considerados com o seu valor igual a um. Além disso, todos os demais termos do dicionário que não fazem parte da consulta têm peso igual a zero. Desse modo, a expressão (2.26) pode ser simplificada para a expressão (2.27), a qual considera para o cálculo da relevância o somatório de peso em que houver o casamento entre o documento e a consulta. Embora, esse caso não se aplica no protótipo de *software* elaborado nesta tese, afinal a frequência de co-ocorrência do bigrama no documento de referência também pode ser considerado no cálculo da relevância.

$$Similaridade(Q,D) = \cos\Phi = \frac{1}{\sqrt{(w_1)^2 + (w_2)^2 + ... + (w_n)^2}} \tag{2.27}$$

Existem várias formas de se cálcular o peso (*w*). Em outras palavras, a meta é calcular o coeficiente de relevância (peso) a partir da consulta de um termo *t* em um documento *d*. A abordagem mais simples de apuração do peso é conhecida como Term Frequency (TF), a qual considera o peso como sendo número de ocorrências do termo *t* no documento *d.* Desse modo, quanto maior for a ocorrência de um termo da consulta em um documento, mais relevante o documento se torna. A expressão para apurar a TF, que foi a adotada nos exemplos apresentados até agora, é mostrada na expressão 2.28 e

$$TF = freq(k,S) \tag{2.28}$$

Onde: *TF é igual* frequência do termo *k* no documento para consulta *S.*

Um dos problemas dessa abordagem é que todos os termos são considerados como tendo igual importância. Enquanto na prática, a capacidade discriminação para determinar a relevância é distinta. Por exemplo, na coleção de documentos sobre uma área específica de conhecimento existem termos que, apesar de raros em documentos de conteúdo geral, são comuns a todos os documentos dessa mesma área. O que leva a concluir-se que produzir uma medida de relevância que considera apenas a frequência do termo no documento não

é uma boa estratégia. Nesse sentido, outras formas de apurar a relevância foram propostas, como se verá a seguir.

Salton e McGill (1983, p. 201, 211) relatam, a partir de seus experimentos, que uma das formas de melhorar a precisão das respostas obtidas nas consultas é adotar técnicas que consideram as características em comum dos documentos (*intra-document*) e as características para fazer a distinção entre os documentos (*inter-document*). Dessa forma, os pesos são usados para computar as similaridades entre os documentos são relativizados pela dissimilaridade, ou capacidade discriminatória do termo. Em outras palavras, quanto maior a frequência do termo em documentos distintos, menos importante ele é para discriminar um documento dentro da coleção. Essa estratégia é conhecida como ponderação Term Frequency Inverse Document Frequency (TF-IDF), na qual existem muitas variantes para cácular o peso *w,* conforme mostrado na expressão 2.29.

$$\begin{aligned} w_1 &= TF = \frac{TF_{doc}}{TF_{\max}} \\ w_2 &= IDF = \log\left(\frac{N}{n}\right) \\ w_3 &= TF * IDF \\ w_4 &= TF * \log\left(\frac{N-n}{n}\right) \end{aligned} \qquad (2.29)$$

Onde se tem: em $w_1$, $TF_{doc}$ como sendo a frequência em que o termo foi encontrado no documento e $TF_{\max}$ como sendo a frequência do termo mais frequente do documento; em $w_2$: o cálculo do inverso da frequência do documento *IDF*, em que *N* é o número total de documentos da coleção e *n* é o número de documento que contém o termo; em $w_3$ o peso é calculado pelo produto de $w_1$ e $w_2$; em $w_4$ em que no denominador da expressão é obtido pela diferença entre a quantidade de documentos da coleção pela quantidade de documentos em que o termo foi encontrado. Desse modo, tem-se que *TF* corresponde às características intradocumentos do termo e *IDF* dá uma medida de distinções interdocumento.

Outras técnicas também podem ser empregadas para calcular o peso *w*, dentre elas destacam-se: coeficiente Jaccard e o coeficiente de Pearson. Adicionalmente, ao se considerar que verbalizar uma consulta não é uma tarefa trivial, outras formas de melhorar os resultados das buscas vêm sendo testadas. Afinal, muitas das vezes, os usuários não conseguem expressar bem a sua real necessidade de informação, não só pela incapacidade individual, mas por ser uma tarefa ambígua e imprecisa. Neste sentido, a meta é melhorar o

resultado das respostas produzido pelos SRI, em que se destacam as técnicas de Relevance Feedback. Ou seja, sistemas que mostram o que o usuário deseja, ou sistemas que expandem a consulta modificando ou adicionando algo o seu conteúdo. Robertson, Spark, Rijsbergen (1976) foram os precurssores da ideia de calcular a relevância através da probabilidade. A ideia geral é calcular o *score* da relevância como sendo a probabilidade do documento *d* como sendo relevante para a consulta *q,* sendo P(*rel*|*d*). Em outras palavras, a ideia é estimar a probabilidade que certos termos têm, ou não, em aparecer em documentos que são e nos que não são relevantes. Isso se dá ao destinguir os termos que são "bons" e os que são "ruins". Onde "bons": significa aparecer em muitos documentos que o usuário deseja e não aparecer em documentos que o usuário não deseja. E, "ruins" significa: aparecer em muitos documentos que o usuário não deseja e não aparecer em documentos que o usuário deseja. Essas probabilidades são estimadas baseadas na contagem da ocorrência dos termos e pode ser obtida pela expressão 2.30.

$$F4(reweighting) = \frac{\frac{r+0.5}{R-r+0.5}}{\frac{n-r+0.5}{N-n-R-r+0.5}} \qquad (2.30)$$

Onde:

- *r* representa o número de documentos relevantes que contém o termo;
- *R* representa o número de documentos marcados como sendo relevantes pelo usuário;
- *n* representa o número de documentos que contém o termo;
- *N* representa número de documentos da coleção.

Sendo que 0.5 é utilizado apenas para evitar a divisão por zero. Um outro aspecto que deve ser considerado, em especial para o foco deste trabalho, é que em uma base de documentos que contém teses, dissertações e artigos se caracteriza por possuir documentos de tamanhos diversos. Tipicamente com documentos que poderão variar desde cinco até pouco mais de três centenas de páginas. Essa variação de tamanho faz com que documentos maiores possam ser sobre estimados no cálculo do peso. Singhal et al. (1996) afirma que durante anos pesquisadores têm trabalhado com a suposição de que a relevância do documento é independente do seu tamanho. Entretanto, seus estudos mostraram que os documentos longos têm maiores chances de serem julgados como relevantes em uma consulta de usuário do que os documentos menores. Nesse sentido,

esses mesmos autores propõem uma variação da função de relevância TF * IDF incorporando no cálculo um fator para normalizar o tamanho do documento. Essa normalização do peso é uma forma de penalizar o peso dos termos identificados em documentos longos, a fim de reduzir ou de até mesmo remover completamente a vantagem que documentos longos teriam em relação aos menores.

No trabalho apresentado por Fang and Zhai (2005), os autores formalizam derivações da função TF * IDF visando atender a quatro axiomas da RI, descritos a seguir:

- A TF frequência do termo é um importante indicador no cálculo da relevância;
- Deve ser considerado o nível de saturação da TF. Ou seja, o crescimento da frequência não pode ser tratado de forma linear no cálculo da relevância. Desse modo, deve haver um ponto de saturação em que o aumento da frequência do termo não produz mais efeito no cálculo da relevância. Isso deve relativisar o valor calculado de tal forma que a ocorrência de vinte vezes não produza tratamento muito diferente do que a ocorrência de dez vezes;
- Palavras mais raras devem ser relativizadas no cálculo. O que é produzido pela aplicação da IDF;
- Documentos longos devem ser normalizados em relação aos documentos menores.

Desse modo, visando normalizar documentos de tamanhos diferentes no processo de busca, Robertson e Spark (1976) propuseram o BM25 como precursor de uma série de modelos derivados desse trabalho. Foi a partir desse trabalho que considera o peso local de um termo que foi desenvolvido um importante modelo de RI, conhecido como Okapi BM25, após ter sido implementado pelo SRI Okapi nos anos 1980 e 1990 pela London's City University. A fórmula utilizada para o cálculo é apressentado em 2.31.

$$score(d,q) = \sum_{i-1}^{n} \frac{TF_{di} * IDF_{di}}{TF_{di} + length_i} \tag{2.31}$$

Sendo que *length* corresponde ao cálculo da razão do tamanho do documento, em palavras, pelo tamanho médio dos documentos da coleção. Ou seja, documentos grandes acima do valor médio do *corpus* terão o valor de tamanho (*length*) maior do que um. Esse valor, somado ao denominador da expressão produz uma redução no valor da relevância. Desse modo, quanto maior for a diferença de tamanho do documento em relação ao valor da média de tamanho, maior será essa redução.

Conforme apresentado por Manning, Raghavan & Schütze (2009, p. 232-243), novos parâmetros foram introduzidos no esquema de ponderação BM25 como forma de construir um modelo probabilístico sensível às variações de tamanho do documento. Algumas dessas variações são apresentadas a seguir em (2.32) e (2.33).

Onde:

- $df_t$ representa o número de documentos que contem o termo $t$;
- $df_{td}$ representa a frequência do termo $t$ no documento $d$;
- $L_d$ representa o tamanho do documento, em palavras;
- $L_{avg}$ representa o valor do tamanho médio dos documentos da coleção;
- $k_1$ é um parâmetro positivo de ajuste para calibrar a escala da frequência do termo no documento;
- $b$ é um parâmetro de ajuste que determina a escala pelo tamanho do documento.

$$RSV_d = \sum_{t \in q} \log\left[\frac{N}{df_t}\right] * \frac{(k_1+1)tf_{td}}{k_1\left[(1-b)+b*\left(L_d / L_{avg}\right)\right]+tf_{td}} \tag{3.32}$$

Se o valor de $k_1$ for igual a zero, fará com que o valor do termo que multiplica a expressão no numerador seja igual a um e no denominador irá zerar a expressão entre colchetes. fará com que o valor do termo que multiplica a expressão entre colchetes seja igual a um. Desse modo, a expressão da frequência do termo no documento será desconsiderada. Por outro lado, aumentar o valor de $k_1$ trará uma maior participação da frequência do termo no resultado do cálculo.

Se $b$ for igual a um, indica o uso total do valor de ponderação de tamanho calculado para cada documento, correspondente ao peso do termo pelo tamanho do documento. Enquanto que, se $b$ for igual a zero a não haverá normalização de tamanho.

$$RSV_d = \sum_{t \in q} \log\left[\frac{N}{df_t}\right] * \frac{(k_1+1)tf_{td}}{k_1\left[(1-b)+b*\left(L_d / L_{avg}\right)\right]+tf_{td}} * \frac{(k_3+1)}{k_3+tf_{tq}} \tag{3.33}$$

Onde:

- $df_{tq}$ representa o número de documentos que contêm o termo $t$ na consulta $q$;
- $k_3$ é um outro parâmetro positivo de ajuste que calibra a escala da frequência do termo na consulta.

Nessa expressão, não é necessário incluir a normalização de tamanho da consulta, pois no processo de RI sempre a consulta tem tamanho fixo.

O ajuste desses parâmetros em ambas as expressões possibilita otimizar o desempenho das respostas para cada *corpus* específico. Segundo esses mesmos autores, testes empíricos mostram que valores dos parâmetros que dão um bom desempenho são para $k_1$ e $k_3$ na faixa entre 1,2 e 2, e do parâmetro $b$, um valor igual a 0,75. Eles afirmam ainda que BM25 tem sido usada com muito sucesso em uma diversidade de tarefas de busca e em diferentes coleções de documentos.

No contexto desta tese, três diferentes formas de calcular o coeficiente de relevância foram implementadas: TF-IDF, CSV e BM25. Entretanto, todos esses modelos foram adaptados para ponderar o peso da frequência dos termos de um bigrama a um coeficiente estrutural (Ce). Dessa forma, o termo que compõe o bigrama tem seu peso relativizado de acordo com a sua estrutura encontrada no documento original. Ou seja, a frequência dos termos será majorada para capturar essas características estruturais do termo no documento. Adicionalmente, uma outra adaptação das técnicas do cálculo de relevância aplicadas neste trabalho se fez necessária. Afinal, em vez dos descritores serem representados por termos isolados, eles representados por bigramas constituídos por termos dependentes. Portanto, para atender aos objetivos específicos desta tese elaborou-se um SRI para atuar em um *corpus* específico realizando buscas a partir das EM extraídas pelo uso dessas três técnicas distintas, a fim de comparar os resultados obtidos. Uma descrição complementar dessas adequações das técnicas empregadas será descrita no próximo capítulo de metodologia, qual apresenta os métodos empíricos utilizados.

A seguir apresentam-se alguns fundamentos das técnicas avaliação das respostas produzidas por um SRI.

### 2.5.3 Avaliando as respostas de um SRI

Para avaliar o desempenho de um SRI, duas métricas são normalmente utilizadas, a Precisão e a Revocação. A Figura 10 ajuda a explicar esses conceitos. Considerando:

- *C* – A coleção de documentos;
- *R* – O conjunto real dos documentos relevantes;
- *A* – O conjunto de respostas obtidas;
- *RA* – Os documentos que são relevantes do conjunto de respostas obtidas.

Portanto, tem-se em (2.34):

$$\boxed{\begin{aligned} \text{Precisão} &= \frac{RA}{A} \\ \text{Revocação} &= \frac{RA}{R} \end{aligned}} \qquad (2.34)$$

A precisão é a razão entre o número de documentos relevantes recuperados sobre o total de documentos da resposta. A precisão mede a proporção dos documentos que são relevantes no conjunto de respostas. E revocação é a razão entre o número de documentos relevantes recuperados sobre o total de documentos relevantes existentes na coleção de documentos. A revocação mede a proporção dos documentos da resposta que são relevantes pelo número total dos documentos relevantes da coleção.

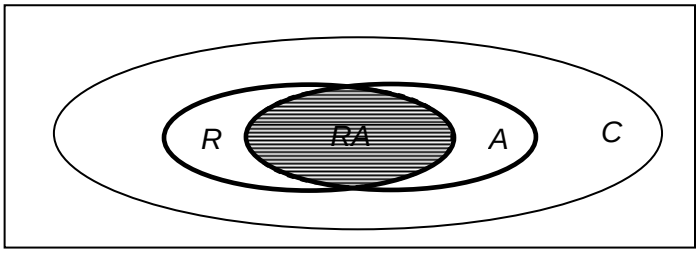



FIGURA 10 – Precisão e Revocação para uma dada requisição
Fonte: Extraída de Baeza-Yates & Ribeiro-Neto (1999, p.75).

Avaliando os extremos, como pode ser observado na Figura 11, tem-se que um sistema que retorna todos os documentos da coleção como resposta indica máxima revocação e mínima precisão. Por outro lado, um sistema que retorna apenas um documento relevante teria máxima precisão e mínima revocação. Se se considerar *RA* = *R*, ou seja, todos os documentos relevantes estão na resposta, a parte hachurada da Figura 11 à esquerda corresponde aos "falsos positivos[14]". Analogamente, a parte hachurada da Figura 11 à direita corresponde aos "falsos negativos[15]".

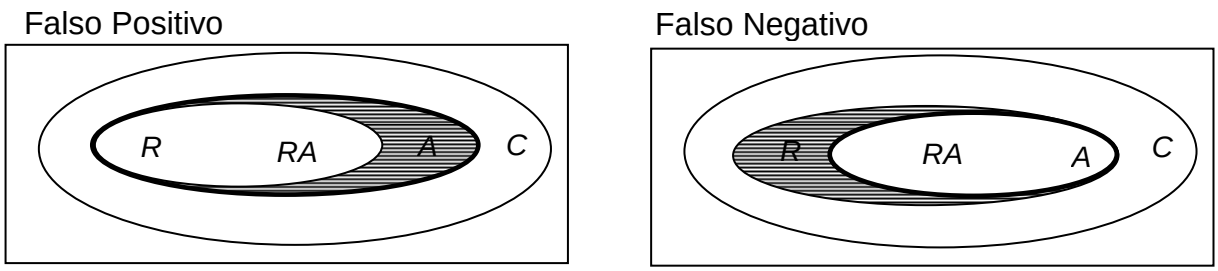



FIGURA 11 – Representa à esquerda máxima Revocação, e à direita máxima precisão.

[14] Corresponde aos documentos retornados que não são relevantes.
[15] Corresponde aos documentos não retornados que são relevantes.

Fonte: Elaborada pelo autor.

Baeza_Yates e Ribeiro-Neto (1999, p.75-83) afirmam que, embora as medidas de precisão e revocação sejam amplamente usadas para avaliar e comparar resultados de algoritmos empregados nos processos de RI, elas apresentam alguns problemas de exatidão. Afinal, para determiná-las é necessário estabelecer através de um grupo de especialistas qual é o correto conjunto de respostas para uma determinada consulta e compará-la com as respostas produzidas por cada algoritmo. Dessa forma, o cálculo das medidas de precisão e revocação baseiam-se na suposição de que o conjunto de documentos relevantes para uma consulta será o mesmo independente de quais foram os especialistas que o determinaram. Entretanto, a interpretação semântica de um conteúdo de texto realizado por uma pessoa é um processo interpretativo com algum grau de subjetividade. Portanto, pessoas diferentes podem ter interpretações diferentes sobre quais documentos são relevantes e quais não são.

## 2.6 O estado da arte

No estudo apresentado por Ladeira (2010), o qual analisa os últimos 40 anos da produção científica nacional da área de PLN, realiza-se a avaliação de 621 publicações de forma horizontal e de 68 publicações através da análise de conteúdo traçando um abrangente panorama evolutivo dessa linha de pesquisa. Nesse cenário, foram constatadas as seguintes evidências: (1) a mudança do enfoque das aplicações: inicialmente, era dada maior ênfase às ferramentas linguísticas de processamento sintático e semântico, e recentemente uma nítida exploração das aplicações práticas; (2) ocorreu um expressivo crescimento de publicações após o ano 2000; (3) as áreas de CC e linguística atingem mais de 80% das publicações, sendo a participação da CI pouco representativa nesse contexto; (4) apenas doze pesquisadores foram os responsáveis por mais de 20% de toda a pesquisa nacional, sendo que desses nenhum se declara como pertencente à área da CI; (5) observa-se que a RI foi a problemática que teve o maior destaque e com uma forte concentração dos trabalhos com publicação recente. Além disso, a maioria dos trabalhos analisados sobre RI estão voltados para técnicas de pré-processamento de documentos, o que segunda a autora sugere que esse ainda seja um tema em aberto. Ainda ela destaca que os resultados obtidos em alguns trabalhos sobre RI têm sido muito ruins, não apresentando melhorias significativas aos trabalhos anteriores.

Não obstante ao panorama nacional e geral do tema PLN, analisado por Ladeira (2010), ao focar a busca mais diretamente com o tema desta tese verifica-se que as técnicas automatizadas de identificação de EM, que servem de base para encontrar os

descritores nos documentos de referência são temas de diversos trabalhos. Entretanto, não foram encontrados trabalhos que utilizam as EM extraídas como descritores no processo de busca para identificação de documentos similares. Dentre os trabalhos que visam identificar as EM publicados nas últimas décadas destacam-se, a seguir, alguns dos mais relevantes.

Mais recentemente, conforme apontado por Wang e Liu (2010), muitos trabalhos têm como foco dominante a identificação e extração de EM. Devido a complexidade desses processos, diferentes abordagens têm sido empregadas. De maneira geral, essas abordagens de extração envolvem: (1) métodos estatísticos; (2) informações linguísticas; (3) métodos híbridos, os quais combinam essas abordagens.

Dentre esses trabalhos que visam identificar as EM destacam-se dentre outros: Dias, Lopes e Guilloré (1999); Evert e Krenn (2005); Chen, Yeh, Chau (2006); Pedersen et al. (2011) que trabalham num contexto independente de linguagem e baseados em métodos estatísticos; Silva e Lopes (1999) que visam extrair *n*-gramas a partir da análise do texto em um contexto local denominado LocalMaxs; Cazolari et al. (2002); Pecina (2010); Portela, Mamede e Baptista (2011) o qual leva em consideração as características morfo-sintáticas do texto, por isso demandam intensivo uso de recursos computacionais; Ramisch (2009) e Villavicencio et al. (2010) que utilizam método híbrido para identificação de EM para o processo de tradução automática.

Outros trabalhos que merecem destaque são os apresentados por Pearce (2002) e por Ramisch, Araújo e Villavicêncio (2012) os quais apresentam uma avaliação comparativa das principais técnicas e abordagens que vêm sendo adotadas por diversos pesquisadores sobre o tema de extração de EM em diversos *corpora* e idiomas.

Cada uma dessas abordagens citadas busca interpretar os conteúdos textuais escritos em linguagem natural, mas seguem caminhos diversos obtendo resultados de custo computacional[16] e de conteúdos distintos. Dessa forma, as vantagens e desvantagens de cada um delas depende do contexto para o qual estão sendo utilizadas.

A abordagem estatística para extração de EM através da co-ocorrência de palavras em textos utiliza várias técnicas que buscam identificar as EM como sendo um conjunto de palavras adjacentes que co-ocorrem com uma frequência acima da esperada para uma sequência aleatória de palavras em um *corpus.* Dessa forma, a abordagem associativa nada mais é que a utilização de um conjunto de medidas de associação que visam identificar as expressões candidatas a EM. Dentre as técnicas empregadas destaca-se: coeficiente de Chi

[16] Custo computacional, neste contexto, relaciona-se ao consumo de recursos computacionais de processamento demandados numa relação direta com o tempo de resposta.

Quadrado de Pearson; coeficiente de Dice; Informação Mútua Pontual (PMI, do inglês Pointwise Mutual Information); medida de Poisson Stirling dentre outras.

No trabalho de Dias, Lopes e Guilloré (1999), os autores questionam que muitos estudos se restrigem a dar apenas um tratamento lexicográfico ao processo de extração de informação de textos. Portanto, sugerem o uso das EM como forma de obter melhor teor informacional do texto que possam ser utilizadas pelas aplicações de RI e tradução automática. Desse modo, eles propõem em seu estudo a implementação de um sistema baseado exclusivamente em técnicas estatísticas para extrair EM, que ocorrem no texto de forma contígua e não-contígua[17]. Eles utilizaram um *corpus* paralelo com o debate político do parlamento europeu com cerca de trezentas mil palavras em cada um dos quatro diferentes idiomas francês, inglês, italiano e português. O sistema proposto reuni os conceitos de Espectativa Mútua (do inglês Mutual Expectation) proposto por Dias (1999 citado por Dias, Lopes e Guilloré) e o processo de aquisição de EM baseado no algoritmo LocalMax proposto por Da Silva (1999 citado por Dias, Lopes e Guilloré). Esse sistema está estruturado nas seguintes etapas: A primeira transforma o conteúdo textual do *corpus* em tabelas de contingência contabilizando os *n*-gramas contíguos e não-contíguos. A segunda mede a coesão de todos os *n*-gramas através do cálculo da Expectativa Mútua para todos eles. A terceira elege as EM comparando todo o conjunto de *n*-gramas pelo valor de coesão utilizando o algoritmo LocalMax. Finalmente a qualidade das EM extraídas é testada e comparada com quatro outras medidas de associação calculadas para cada um dos idiomas existentes no *corpus.* Como resultado, os autores apontam que a técnica de Espectativa Mútua apresentando maior precisão na extração, além de superar o problema da palavra muito frequente que ocorre nas demais técnicas de medida de associação empregadas.

Dias, Lopes e Guilloré (1999) destacam que a maioria dos trabalhos de PLN têm se concentrado no reconhecimento e extração de informações explícitas no texto negligenciando os contextos implícitos compostos por unidades léxicas que devem ser consideradas como indivisíveis por terem um significado ou função que não é necessariamente o mesmo que analisar cada uma das palavras separadamente. Dessa forma, esses autores trabalham utilizando métodos estatísticos de expectativa mútua conjugando o processo de aquisição lexical com o algoritmo de máximo local para identificação dos léxicos compostos.

[17] Segmentos de texto não contíguos são aqueles que as EM aparecem com quebra da sequência das palavras pela presença um ou mais palavras intercambiáveis dentro do segmento.

A abordagem utilizada por Evert e Krenn (2005) é baseada no cálculo estatístico das medidas de associação das palavras contidas no texto. Nos testes empíricos esses autores utilizaram um subconjunto de oito milhões de palavras extraídas de um *corpus* constituído de um jornal escrito no idioma alemão. A abordagem proposta foi dividida em três passos.

No primeiro extraem-se as tuplas léxicas do *corpus* fonte contendo pronomes (*P*), substantivos (*S*) e verbos (*V*). Em seguida, esses dados são agrupados em pares (*P*+*S, V*) e colocados em uma tabela de contingência, representada por uma estrutura tridimensional, em que cada par está colocado num plano *P*+*S* por *V*. Finalmente atribui-se no terceiro eixo do plano a informação da frequência representada por quatro células. Dessa forma, realiza-se uma comparação entre todos os pares léxicos extraídos do *corpus* com as suas sentenças, contabilizando a cada sentença, uma das quatro possibilidade: existe *PS* e existe *V*; existe *PS* e não existe *V*; não existe *PS* e existe *V*; não existe *PS* e não existe *V*. Ou seja, é acrescida uma unidade para cada vez que uma das possibilidades ocorrerem.

No segundo passo, as medidas de associação são aplicadas às frequências coletadas no passo anterior. Desse processamento resulta uma lista de pares de EM candidatas com seus respectivos *scores* de associação calculados ordenados do mais fortemente associado para o menos fortemente associado. Os "*n*" primeiros candidatos da lista são selecionados para serem utilizados no próximo passo.

O terceiro passo constitui da avaliação da lista EM gerada por um especialista humano que retira manualmente os falsos positivo identificados pelo processo automatizado.

Dessa forma, a abordagem proposta por esses autores se caracteriza por ser uma extração de EM semi-automática. Esses mesmos autores propõem o uso de uma técnica de extração de uma amostra aleatória e representativa do *corpus* em vez do conjunto completo dos documentos que visa minimizar o trabalho intelectual de um especialista.

Yagonova e Pivovarova (2010) trabalham para identificar a natureza das *collocations* no idioma russo. Esses autores utilizam medidas estatísticas que permitem identificar automaticamente as *collocations* no texto e ranqueá-las de acordo com o seu grau de estabilidade ou correspondência com o valor da medida escolhida. A lista das *collocations* identificadas é um reflexo das características linguísticas e extralinguísticas encontradas nos textos analisados, sendo que a ordenação pela relevância depende da técnica de medida de associação estatística empregada no processo de ranqueamento. Os autores utilizaram uma coleção de textos totalizando mais de 60 milhões de *tokens* extraídos de um site de notícias do período de abril a dezembro de 2009. O método automatizado envolveu inicialmente uma marcação morfológica da coleção, seguida por uma análise sintática a fim de remover parcialmente os homônimos. O conteúdo resultante desse processamento foi separado em

fragmentos de texto tomando como base os marcadores de pontuação. O próximo passo foi identificar as cem *collocations* melhor ranqueadas obtidas através das medidas estatísticas Mutual Information (MI) e T-Score que ocorram numa frequência acima de quarenta vezes. Finalmente, os resultados obtidos foram manualmente analisados comprovando as hipóteses propostas: o método MI permite destinguir nome de objetos, termos e combinações complexas refletindo a área de conhecimento ou assunto do texto; enquanto que T-Score, trabalha melhor para distinguir as propriedades estilísticas do texto, ou seja, “combinações linguísticas gerais” (derivadas de palavras funcionais e palavras discursivas) e "construções agrupadas". Um dos problemas que esses autores identificaram no uso da técnica MI é que a medida depende do tamanho do *corpus* analisado e ela tende a sobre estimar ruídos, tais como palavras estrangeiras.

Uma outra forma de abordar o problema é através da abordagem simbólica, que busca encontrar o sentido sintático, morfológico e pragmático do texto baseando-se em um dicionário controlado de palavras e em um conjunto de regras visando à interpretação do mesmo. Nesse caso, o processamento é fortemente dependente do idioma e do domínio do *corpus*, enquanto a abordagem estatística procura dar um tratamento ao texto através do reconhecimento de padrões de comportamento baseados na frequência de co-ocorrência das palavras. Ou seja, as EM são o conjunto de palavras que co-ocorrem numa frequência acima do acaso. É possível identificar também abordagens que visam extrair EM de forma automatizada ou semi-automatizada, na qual ocorre a supervisão de um especialista.

Cazolari et al. (2002) utilizam uma abordagem focada nas EM que são produtivas por um lado e, que, por outro, demonstram regularidades que possam ser generalizadas para as classes de palavras com propriedades semelhantes. Particularmente, eles buscam encontrar dispositivos gramaticais que permitam a identificação de novas EM motivados pelo desejo do reconhecimento o mais automatizado possível na aquisição das EM. Nesse sentido, a pesquisa desses autores estudou em profundidade dois tipos de EM: os verbos de suporte e os substantivos compostos (ou complexos nominais). Segundo eles, esses dois tipos de EM estão no centro do espectro de variação em composicionalidade que pode ser observado pela coesão interna juntamente com um elevado grau de variabilidade em lexicalização e variação dependente do idioma.

A pesquisa conduzida por Villavicencio et al. (2010) busca extrair as EM combinando duas abordagens distintas: a abordagem associativa e a abordagem baseada em alinhamento lexical[18]. Na primeira, as medidas de associação são aplicadas para todos os

[18] Dois textos escritos em idiomas distintos são considerados como alinhados quando eles possuírem marcas que identifiquem os pontos de correspondência entre o texto original e a sua tradução.

bigramas e trigramas gerados a partir do *corpus* e o resultado dessas medidas são utilizados para avaliação. A segunda abordagem extrai de forma automatizada as EM tomando como base os alinhamentos lexicais das versões de um mesmo conteúdo escrito nos idiomas português e inglês. O alinhamento final é gerado a partir da interseção dos alinhamentos em ambos os sentidos. Antes de usar a técnica de alinhamento, o *corpus* é etiquetado morfossintaticamente a fim de aplicar filtros de categorias gramaticais na lista inicial de EM extraídas. Para combinar os resultados obtidos pelas duas abordagens os autores utilizaram as redes bayesianas.

A abordagem de alinhamento lexical verifica se a EM encontrada em um documento escrito em um idioma também ocorre na versão correspondente escrita em outro idioma. Para ser possível essa análise, os documentos necessitam estar alinhados através da correspondência das palavras entre as diferentes versões expressas em idiomas distintos. Entretanto, para ser possível o alinhamento é necessário que os documentos sejam analisados a partir de seus aspectos morfológicos tratados através de um pré-processamento de etiquetação[19]. Dessa forma, as classes gramaticais são utilizadas como informação adicional no processo de identificação das EM.

Na pesquisa desenvolvida por Zhang et al. (2009), é proposto um método denominado Enhanced Mutual Information and Collocation Optimization (EMICO) para extrair EM com foco nas entidades nomeadas. Estas se caracterizam por serem compostos contíguos com duas a seis palavras que descrevem conceitos com padrões sintáticos mais estáveis. Esses autores empregam essa técnica em processamento de mineração de textos e a comparam com as técnicas de indexação tradicional do modelo de espaço vetorial conjeturando que o uso da EM para interpretação semântica do texto produz melhores resultados do que os modelos estatísticos e semânticos que lidam com palavras individuais.

No trabalho de Chen, Yeh, Chau (2006), apresenta-se um sistema alternativo para extrair EM, por considerar que os métodos estatísticos tradicionais lidam com uma grande quantidade de dados ruidosos e que consumem muito tempo de processamento. Desse modo, eles elaboraram um experimento baseado em um *corpus* constituído de 308 documentos escritos em chinês tradicional, que considera cada ideograma como sendo uma palavra e aplicaram uma metodologia dividida em quatro passos:

- Gerar segmentos – nesse passo, os autores utilizaram uma pequena e pré-definida lista de *stop words* como entrada inicial. O objetivo é utilizar as *stop words* como *tokens* para o processo de separar o documento em segmentos

[19] Programa de computador conhecido genericamente como etiquetador de categorias gramaticais. Gera uma saída, normalmente em XML, associando cada palavra à sua classe gramatical: substantivo, verbo, artigo, etc.

de texto. Durante esse estágio, a frequência do segmento de texto é calculada para o respectivo documento e também para o conjunto dos documentos.

- Calcular o peso dos segmentos de texto – no cálculo do peso são consideradas a quantidade de palavras contidas no segmento de texto, e as respectivas frequência do segmento no documento e no *corpus*.
- Fragmentar os segmentos de texto – nesse passo, é aplicada uma regra que considera que um segmento de texto só pode ser separado se, e somente se, ele estiver contido dentro do outro e ao mesmo tempo o seu peso for maior que do outro segmento. Esse procedimento faz com que segmentos de maior peso tenham menor possibilidade de serem segmentados, pois já representam uma EM.
- Selecionar as EM – por fim aplica-se um filtro ao valor calculado do peso do segmento. Esse processo limita as EM extraídas como sendo somente aquelas que tiverem os maiores pesos tomando como base o valor do limite informado.

Um trabalho bastante afim encontrado na literatura, não pelo método empregado, mas pelo uso intuitivo do mesmo conceito é o modelo de verificação de aspectos combinados apresentado por Roussinov (2012). Diferentemente das técnicas de consulta convencionais, de expansão ou de tradução que estão limitadas a apenas buscar nos documentos os unigramas de forma independente, esse trabalho se caracteriza por propor uma função de similaridade para ranquear as respostas obtidas através da apropriação de dois aspectos: os aspectos presentes $A_p$ e os aspectos faltantes $A_f$, sendo esses condicionados à presença dos aspectos presentes. Ou seja, considera a consulta do usuário como uma sequência de palavras (*n*-gramas) de tal forma que os termos de busca são avaliados também como termos dependentes. Desse modo, o método empregado considera os dois aspectos que são automaticamente identificados e tratados. O primeiro $A_p$, a presença do termo no documento, é verificada pelo casamento exato. O segundo $A_f$ considera uma estimativa dos aspectos que não se manifestam explicitamente nos documentos da coleção. Desse modo, a presença implícita de um aspecto é obtida pela sua presença explícita, estatisticamente prevista. A predição é baseada nos indicadores contidos no texto. Esses indicadores são sequências (*n*-gramas) de palavras no documento composto de até três termos. Para cada indicador *i* oriundo do documento, o algoritmo estima $P(A_f \mid i, A_p)$, que é a probabilidade de ocorrência dos aspectos faltantes condicionada

à ocorrência conjunta de $A_p$ e *i*. Em geral, cada probabilidade condicional é estimada como mostrada na expressão (2.35).

$$\boxed{P\left(A_f \mid i, A_p\right) = \left(\frac{NUM}{DEN}\right)} \qquad (2.35)$$

Sendo NUM (numerador) o tamanho da amostra dos documentos em que o aspecto faltante $A_f$ ocorre com $A_p$ e *i,* e DEN (denominador), o tamanho da amostra de documentos em que o indicador *i* ocorre juntamente com $A_p$. Para estimar o tamanho da amostra, foi utilizada a um subconjunto World Wide Web, (a Wikipedia) extraída através da API do motor de busca Bing da Microsoft. O modelo empregado se baseia apenas nas características booleanas da linguagem de consulta, especificamente sobre os operadores AND (conjunção) e NEAR (proximidade). O objetivo é captar a presença conjunta de *i* e $A_p$ através do operador NEAR, por exemplo, "Antartica NEAR station", já para captar a presença de $A_f$, $A_p$ e *i* várias combinações diferentes dos operadores NEAR e AND são utilizadas. Desse modo, para cada consulta do usuário, o algoritmo analisa milhares de tais possíveis indicadores.

Um trabalho recente apresentado por Rayson et al. (2009), faz um retrospecto histórico da pesquisa sobre EM e destaca alguns dos principais grupos de pesquisa que atuam no mundo. A meta é facilitar o trabalho daqueles que estão desenvolvendo pesquisas nessa área. Esses autores relatam que, no início dos anos 1990, as EM passaram a receber maior atenção dos pesquisadores de PLN, nesse sentido eles citam a influência dos trabalhos de Smadja (1993)[20], Dagan and Church (1994)[21], Wu (1997)[22]; Daille (1995)[23], dentre outros. Eles destacam que um importante marco ocorreu a partir de 2001 com o interesse despertado pelo Centre for the Study of Language and Information (CSLI), da universidade de Stanford, o qual visa investigar um meio para codificar a variedade de EM em gramáticas de precisão. Outro importante trabalho tem sido conduzido pela universidade de Lancaster, o qual resultou em uma grande coleção de termos semanticamente anotados. Com esses recentes desenvolvimentos de corpus linguísticos, pesquisadores de diferentes áreas têm se juntado, possibilitando desenvolver trabalhos que abordam as EM a partir de diferentes perspectivas. Nesse sentido, desde 2003, essa comunidade de pesquisadores

[20] Smadja, F. (1993). Retrieving collocations from text: Xtract. Computational Linguistics, 19(1), 143–177.

[21] Dagan, I., & Church, K. (1994). Termight: Identifying and translating technical terminology. In Proceedings of the 4th conference on applied natural language processing (pp. 34–40). Stuttgart, German.

[22] Wu, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Lingusitics, 23(3), 377-401.

[23] Daille, B. (1995). Combined approach for terminology extraction: Lexical statistics and linguistic filtering. filtering. Technical paper, UCREL, Lancaster University.

vêm organizando workshops dentro de importantes congressos da área da CC, tais como ACL e LREC. Esse interesse reflete a importância desse tema dentro do campo de pesquisa da PLN. Sendo que, nos primeiros workshops o tópico mais recorrente estava relacionado ao processo de identificação e extração automática de EM, surgindo propostas a partir de diferentes perspectivas, baseadas em análise linguística e em medidas estatísticas. Com o decorrer das pesquisas e objetivando desenvolver algoritmos mais eficientes, as pesquisas se voltaram para a busca do entendimento mais profundo das propriedades estruturais e semânticas das EM, tais como padrões morfo-sintáticos, composionalidade semântica, comportamento semântico em diferentes contextos, propriedades de transformação de EM entre idiomas, etc. Após duas décadas de esforços, a comunidade de linguística computacional vem construindo valiosos recursos e ferramentas para aplicação no mundo real do PLN, tais como mapeamento de termos de consultas para sinônimos tanto para o uso em SRI, quanto para sistemas de tradução automática e mineração de dados em textos que demandam a identificação de conceitos multipalavras.

Ainda segundo esses mesmos autores, apesar do considerável esforço que tem sido depreendido nas pesquisas sobre EM, ainda existe um longo caminho a ser seguido.

Na busca de extrair sentido de um texto a partir de suas partes mais relevantes, outras estratégias têm sido adotadas. Nesse sentido destaca-se o uso dos sintagmas nominais utilizados como descritores de busca, abordados pelos trabalhos de Kuramoto (1995) e Souza (2005) e da pesquisa de Maia & Souza (2010) que buscam utilizar os sintagmas para agrupar documentos correlatos. O método de identificação dos sintagmas nominais utiliza uma abordagem baseada na linguística, em que as palavras do texto são previamente etiquetadas a fim de identificá-las em classes gramaticais que servirão de base para a extração dos sintagmas. Entretanto, a identificação dos sintagmas exige um processamento analítico em profundidade das sentenças que demanda um exaustivo processamento computacional baseado em regras dependentes do idioma.

No contexto desta tese, como se busca testar um processo de RI, através da utilização de partes semanticamente relevantes do texto como descritores para o processo de Busca Comparada, a um custo computacional que viabilize o tempo de resposta para o processamento *online* e independente de idioma*,* opta-se pelo uso das EM que são mais fáceis de serem obtidas. Esses aspectos levam a supor que o uso das EM é mais apropriado para o contexto da aplicação da metodologia proposta de recuperar documentos similares em um *corpus* a partir das EM extraídas de um documento utilizado como referência para a busca.  O ineditismo desta tese está em obter a semântica do texto representado pelas EM, obtidas a partir de um algoritmo determinístico que utiliza aspecto

da estrutura física dos documentos, a serem utilizadas como descritores do processo de busca. Para verificar e eficiência do método extrair-se-ão as EM comparando os resultados obtidos pela técnica determinística proposta em relação a treze diferentes técnicas de medidas de associação estatística obtidas através do pacote NSP.

Esta tese tem como proposta combinar métodos que visam extrair a semântica do texto, considerando os aspectos da estrutura física dos documentos e representada pelas EM. Dessa forma, o fio condutor dessa proposta é a busca por reduzir o conteúdo de um documento a um conjunto de descritores multipalavras (*n*-gramas) que possam expressar o seu sentido. A partir dessas EM identificadas, utilizá-las como descritores do processo de busca de documentos cujo assunto esteja corelacionado ao documento utilizado como referência.

# 3 METODOLOGIA

A metodologia utilizada nesta tese emprega uma abordagem dividida em quatro fases que ocorrem em momentos distintos, a saber:

A primeira fase, descrita na seção 3.1, consiste em selecionar os documentos de um domínio específico para criar um *corpus*; elaborar um componente de *software,* denominado “Server.exe”, capaz de processar esses documentos a fim de filtrar, separar os seus conteúdos em sentenças e palavras com o objetivo de disponibilizar duas funcionalidades: (1) indexar as palavras em uma estrutura de lista invertida em memória a fim de disponibilizar um serviço de rede via Hipertext Transfer Protocol (HTTP) para possibilitar a consulta dos documentos da coleção; (2) converter os documentos processados para o formato texto puro e gravar arquivos com o mesmo nome dos documentos originais e com a extensão renomeada para (.txt). Resumidamente o objetivo é disponibilizar um serviço de RI da coleção de documentos e convertê-los em formato de texto para servir de base para o processamento da terceira fase, descrita na seção 3.3.

A segunda fase, descrita na seção 3.2, consiste em elaborar um outro componente de *software*, denominado “Client.exe”, capaz de processar o documento de referência fornecido pelo usuário da busca. O objetivo é filtrar e separar o texto em sentenças e palavras a fim de disponibilizar duas funcionalidades: (1) ordenar as palavras em uma estrutura de dados em memória que sirva de base para extração das EM através de uma heurística determinística denominada Heudet, proposta pelo autor. As EM identificadas servem de descritores no processo de Busca Comparada através de requisições de consulta ao serviço Server disponibilizado via rede; (2) gravar as EM identificadas em arquivos de mesmo nome dos documentos originais e com a extensão renomeada para (.heudet), em formato texto puro, a serem utilizados na quarta fase, descrita na seção 3.4.

A terceira fase, descrita na seção 3.3, consiste em processar os arquivos (.txt) produzidos pela primeira fase, utilizando o *software* Ngram Statistics Package (NSP) a fim de identificar as EM através de treze diferentes técnicas probabilísticas de medidas de associação estatísticas. Nesse sentido foi elaborado um *script*, mostrado no apêndice A, para disparar a execução dos componentes do *software* NSP, possibilitando automatizar o processo. Para cada documento do *corpus* é executado um processamento para cada uma das técnicas estatísticas disponibilizadas no pacote NSP. Dessa forma são, produzidos treze arquivos de mesmo nome dos documentos originais e com uma extensão própria renomeada correspondente para cada uma dessas técnicas. Esses arquivos são gerados para serem utilizados na quarta fase, descrita na seção 3.4.

A quarta fase, descrita na seção 3.4, consiste em comparar os resultados apresentados pelas técnicas determinísticas e estatísticas utilizadas para realizar o processo de extração de EM, ocorridos na segunda e terceira fases. Finalmente, os resultados obtidos são analisados em termos de tempo de resposta, precisão.

Essas quatro fases descritas operacionalizam todo o processo experimental desta pesquisa. A execução de cada fase ocorre através da realização das etapas contidas em cada fase. Para um melhor entendimento algumas dessas etapas foram divididas em subetapas que contêm os passos seguidos para realizar a etapa em um nível mais detalhado. Desse modo, a metodologia será apresentada em dois níveis de detalhamento:

- o primeiro nível, apresenta uma visão geral das etapas contidas nas seções, o qual é mostrado na Figura 12;
- o segundo nível, uma visão que detalha algumas dessas etapas em subetapas, as quais são apresentadas nas figuras descritas a seguir..

A etapa 3.1.2 compartilhada com a etapa 3.2.2 denominada “Converter documentos PDF em termos normalizados” é detalhada na Figura 13. A etapa 3.1.3 denominada “Processar termos” é detalhada na Figura 14. E, finalmente a etapa 3.2.2 denominada “Gravar as EM entraídas em arquivos (.heudet)” é detalhada na Figura 15.

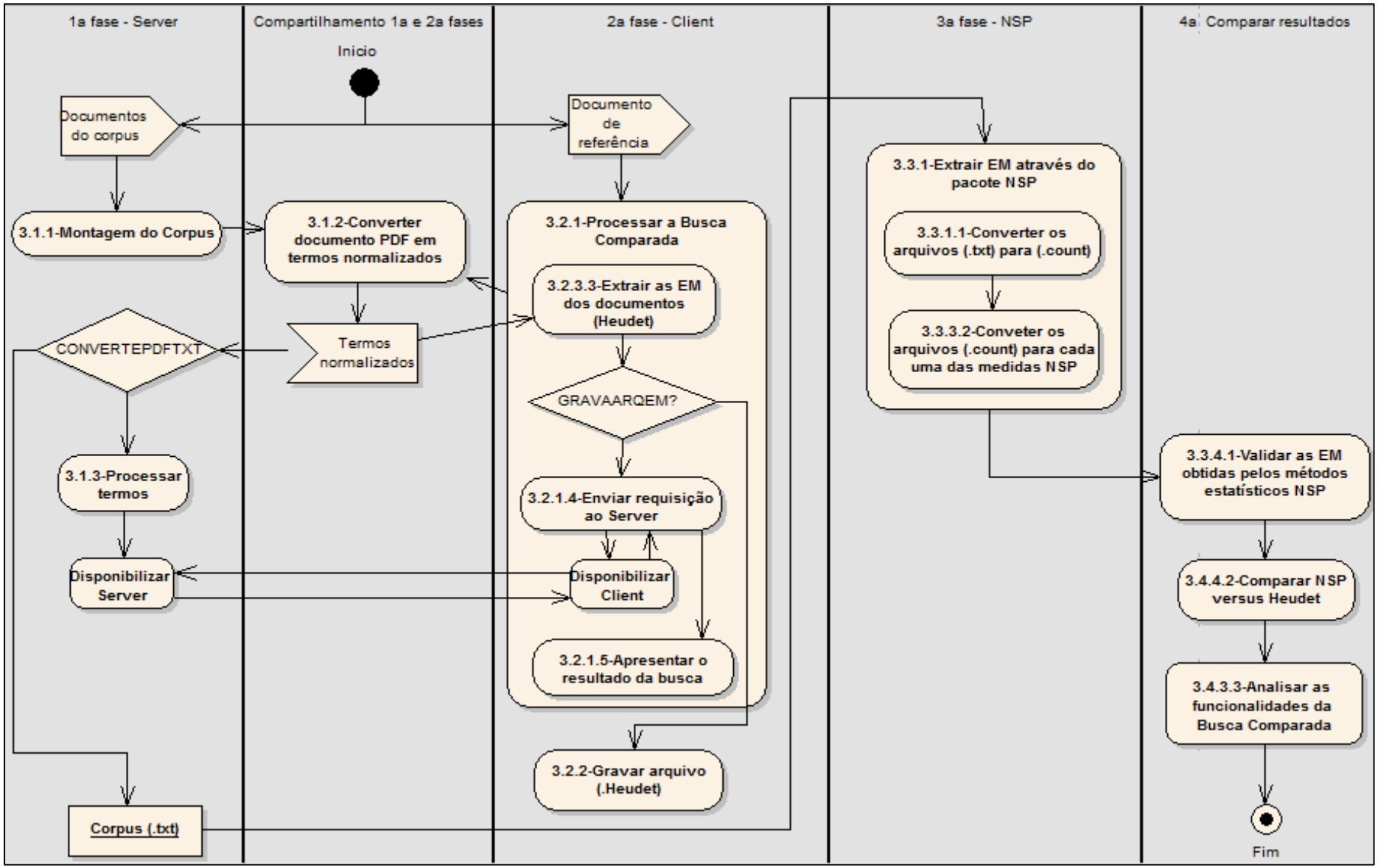



FIGURA 12 – Visão geral das fases operacionais da metodologia utilizada.
Fonte: Elaborada pelo autor.

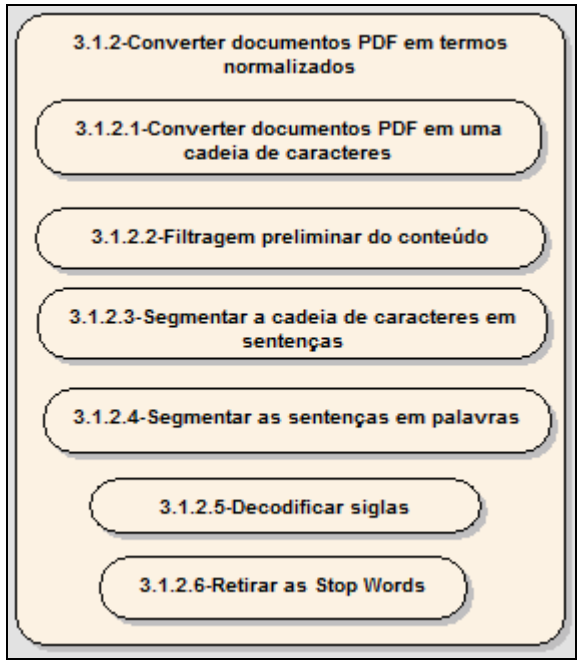



FIGURA 13 – Detalhamento da etapa 3.1.2.
Fonte: Elaborada pelo autor

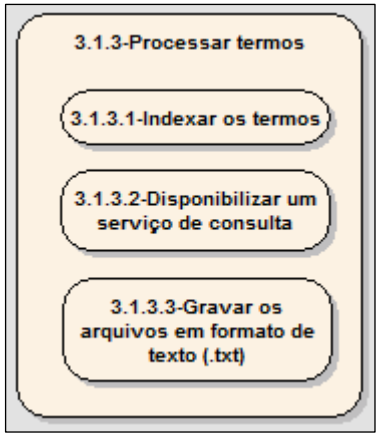



FIGURA 14 – Detalhamento da etapa 3.1.3
Fonte: Elaborada pelo autor

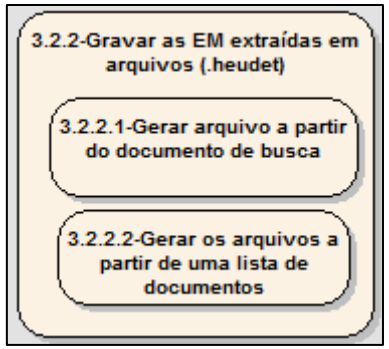



FIGURA 15 – Detalhamento da etapa 3.2.2.
Fonte: Elaborada pelo autor

## 3.1 Descrição da primeira fase

Para um melhor entendimento dos processos desenvolvidos, essa fase foi subdividida nas seguintes etapas.

- Etapa 1 – **Montagem do *corpus*** – Nessa etapa, foi realizada uma coleta dos artigos publicados no principal congresso da área da Ciência da Informação (Enancib-2010) em formato PDF. O objetivo é definir uma estrutura hierárquica no sistema de arquivos computadorizado para ordenar e organizar esses arquivos e suas extensões geradas a partir dos testes empíricos realizados.
- Etapa 2 – **Converter documento PDF em termos normalizados** – Nessa etapa, foi elaborada uma biblioteca de funções de *software* para ser utilizada de forma compartilhada pelo *software* desenvolvido, na primeira e segunda fases.
- Etapa 3 – **Processar termos** – Nessa etapa, o objetivo é processar os termos produzidos pela etapa anterior a fim de atender a duas funcionalidades principais: (1) disponibilizar um serviço de busca por palavras-chave compostas por bigramas; (2) gerar um arquivo, com formato de texto puro, com o mesmo nome do arquivo original, mas com a extensão (.txt).

A seguir cada uma dessas etapas será descrita em detalhes.

### 3.1.1 Montagem do *corpus*

O objetivo dessa etapa é coletar os documentos de um domínio específico para montagem de um *corpus*. Sendo assim, selecionamos todos os artigos completos publicados no ano de 2010 do principal encontro científico da área da Ciência de Informação (ENANCIB). A Tabela 1 mostra a quantidade de artigos publicados pelos seus respectivos Grupos Temáticos (GT) participantes desse encontro nacional. Todos os arquivos foram obtidos em formato Portable Document Format (PDF) e armazenados em um sistema de arquivos informatizado organizado em pastas e subpastas de forma hierárquica por GT, totalizando 195 artigos.

TABELA 1 – Composição das quantidades de artigos do *corpus* utilizado.

| **Ano** | **2010** | **Ano** | **2010** |
|---|---|---|---|
| GT 1 | 19 | GT 6 | 19 |
| GT 2 | 21 | GT 7 | 20 |
| GT 3 | 14 | GT 8 | 18 |
| GT 4 | 34 | GT 9 | 10 |
| GT 5 | 16 | GT 10 | 24 |
| TOTAIS | | | 195 |

Fonte: Elaborada pelo autor.

Cabe ressaltar que um documento foi descartado por ser apenas um anexo composto por duas páginas. Desse modo o total de documentos contidos no *corpus* utilizado totalizou 194. Foi definido utilizar esse *corpus* reduzido de documentos devido à limitação imposta pelo *software* Text Extraction Toolkit[24] v. 4.0 (PDF-TET), em sua versão de avaliação, o qual suporta todas as funcionalidades da versão licenciada, entretanto, com limitação de ler documentos PDF que contenham até dez páginas e com no máximo um Mbyte de tamanho. Os artigos científicos em geral, incluindo os utilizados neste trabalho, normalmente possuem mais de dez páginas, tipicamente até trinta páginas. Sendo assim, foi necessário contornar essa restrição imposta pelo *software* para realização do trabalho experimental. Portanto, utilizou-se um outro *software* Adolix Split and Merge PDF[25] v.2.1.29, também com limitações de funcionalidades por ser uma versão *freeware.* Desse modo, foi necessário dividir previamente cada um dos documentos do *corpus* em pedaços de uma página. Esse processo teve de ser realizado de forma manual através de uma interface Windows limitada a cinco páginas por vez, por causa da limitação da versão utilizada. Desse modo, foi necessário executar o programa de fragmentação dos documentos em médias cinco vezes para cada um dos 194 documentos do *corpus.* Como resultado final, cada documento PDF original foi separado em vários arquivos. Um arquivo para cada página adicionando ao nome do arquivo original o número da página. A Figura 16 apresenta à esquerda a tela de interface do *software* Adolix, utilizado para fragmentar um documento PDF e, à direita a estrutura de diretórios onde o *corpus* foi armazenado, identificado por o caminho relativo a partir da pasta Enancib2010. Finalmente, Logo abaixo, também no lado

[24] PDF-TET é um acrônimo para *Portable Description Format – Text Extraction Toolkit*. Uma *Application Program Interface* (API) para extração de texto imagens e metadados de documentos em formato PDF. *Copyright* © 1997-2010 PDFLib GmbH.

[25] Adolix Split and Merge pfd – é um aplicativo que permite dividir um documento pdf em vários arquivos, um por página, ou agrupar vários arquivos pdf em um único arquivo.

direito, um exemplo de fragmentação do documento (31.pdf). Nesse exemplo, o documento se mostra dividido em 17 arquivos nomeados como de 3101.pdf até 3117.pdf.

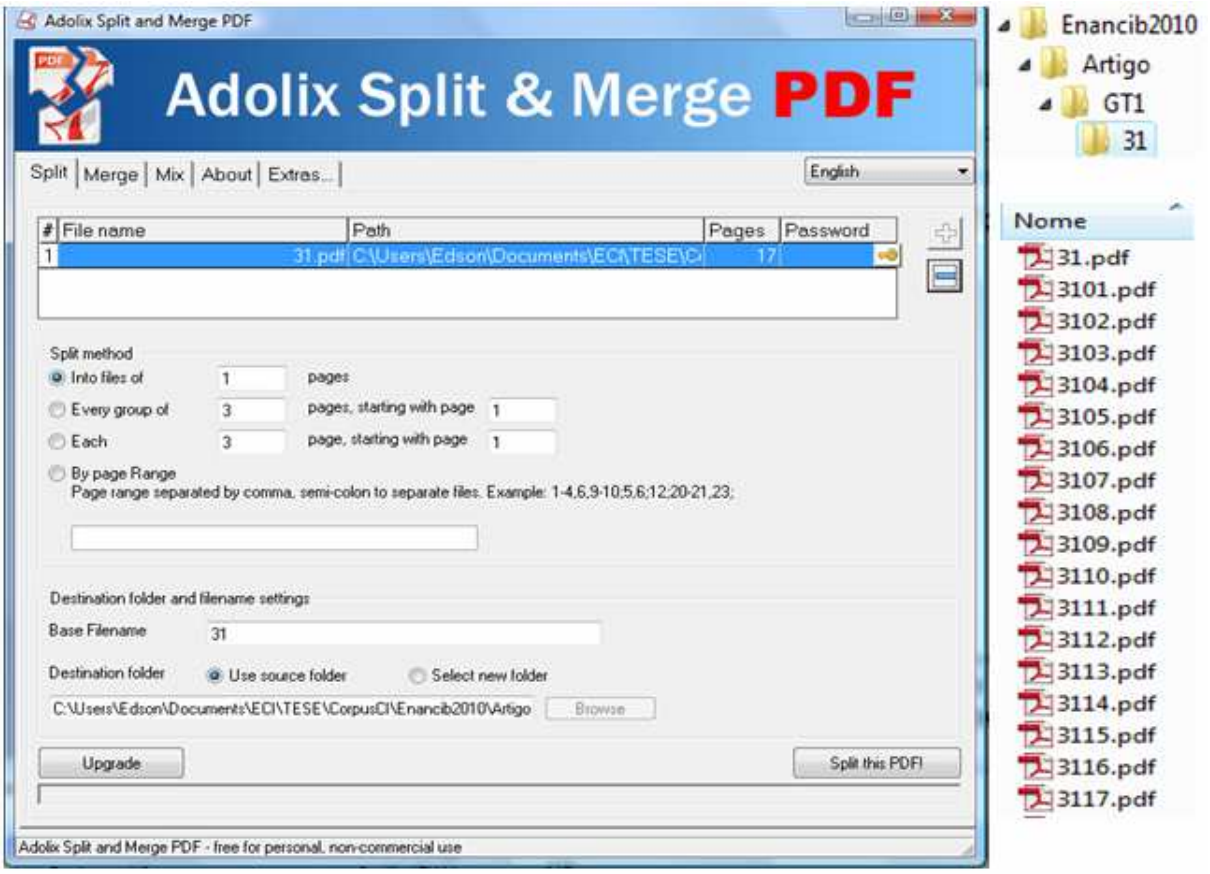



FIGURA 16 – Esboço da fragmentação do documento 31.PDF utilizando o *software* Adolix.
Fonte: Elaborada pelo autor.

Essa estrutura pré-definida do nome do arquivo permitiu ao *software* elaborado pelo autor processar o documento em páginas e agrupá-lo internamente a fim de tratá-lo como um conteúdo único.

Cabe ressaltar que, embora tenham sido criados todos esses artifícios para contornar a necessidade de aquisição de uma licença comercial, isso não altera a avaliação dos resultados obtidos neste trabalho. Uma eventual utilização do protótipo elaborado como uma aplicação real e com a devida aquisição da licença do pacote Text Extraction Toolkit serão necessárias pouquíssimas adaptações no *software* elaborado. Sendo que, nesse caso, o uso do *software* Adolix fica dispensado e o tempo de execução da ferramenta como um todo ficará otimizado. Afinal, em vez de converter vários arquivos a cada consulta do usuário e em seguida juntá-los, o sistema terá apenas de lidar com um único arquivo simplificando sobremaneira o processamento como um todo. A Figura 17 apresenta um esboço parcial do sistema de arquivos criado.

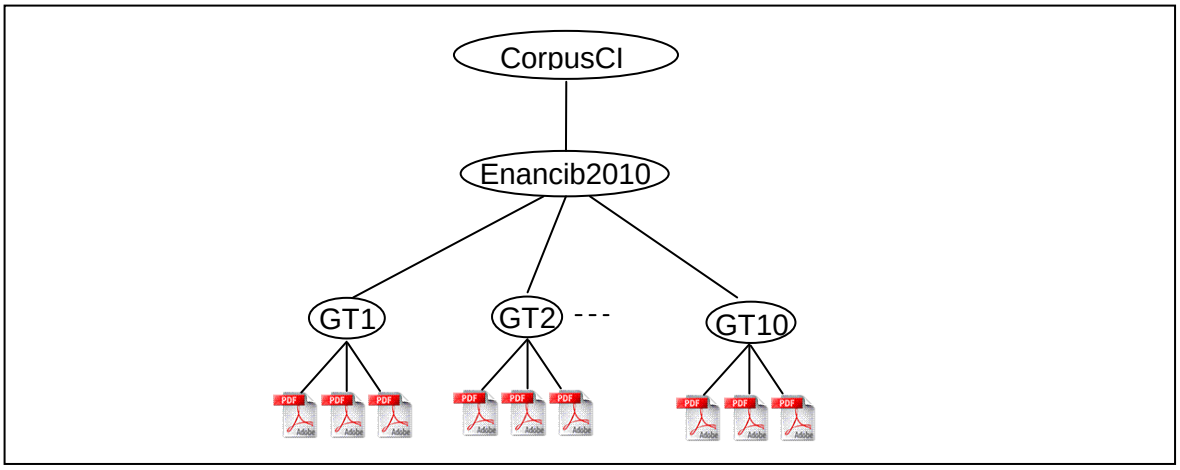



FIGURA 17 – Estrutura hierárquica do *corpus.*
Fonte: Elaborada pelo autor.

### 3.1.2 Converter documento PDF em termos normalizados

Essa etapa é apresentada dividida em seis subetapas distintas com objetivo de dar maior clareza ao processo de conversão dos documentos. Na prática, esse bloco de tarefas será executado sempre em conjunto, tanto na primeira quanto na segunda fase, para garantir que o processo de identificação e normalização dos termos seja sempre o mesmo, tanto na indexação do *corpus,* quanto na segmentação em termos do documento usado como referência da Busca Comparada. Essas subetapas são mostradas a seguir:

- Subetapa 1 – Converter documentos PDF em cadeia de caracteres;
- Subetapa 2 – Filtragem preliminar do conteúdo;
- Subetapa 3 – Segmentar a cadeia de caracteres em sentenças;
- Subetapa 4 – Segmentar as sentenças em palavras;
- Subetapa 5 – Decodificar siglas;
- Subetapa 6 – Retirar as *stop words*.

#### 3.1.2.1 Converter documentos PDF em uma cadeia de caracteres

O propósito desta subetapa é processar cada documento de entrada em formato PDF, a fim de transformá-lo em uma cadeia de caracteres, sentenças e palavras, em formato de texto puro. Afinal, para que documentos digitais possam ser processados pelo aplicativo de indexação e extração de EM é necessário que estejam no formato texto, codificados no padrão ASCII. Ou seja, é necessário converter o arquivo do formato binário, típico do *software* no qual ele foi editado, para um formato de texto puro. Esse processo,

usualmente, pode ser feito utilizando um *software* de conversão ou uma biblioteca que manipula documentos em diferentes formatos e padrões de codificação. Em geral, essa não é uma tarefa trivial. Na prática, os documentos podem estar em diversos formatos binários proprietários e codificados em diversos padrões de codificação. Isso inclui caracteres em qualquer idioma, tais como: chinês, árabe, caracteres acentuados do português, dentre outros que são representados em múltiplos *bytes*, além de ainda poder conter imagens e metadados.

Esta tese trabalhará somente com documentos de entrada no formato PDF protegidos ou não. Para realizar a conversão para o formato de texto serão utilizados os dois *softwares* o Adolix e PDF-TET citados anteriormente. O primeiro permitiu segmentar os documentos do *corpus* página a página gerando um arquivo por página. E o segundo *software* é composto de uma Dynamic Link Library (DLL) que foi acoplada aos componentes de *software* elaborados com o objetivo de converter cada página dos documentos em texto puro.

#### 3.1.2.2 Filtragem preliminar do conteúdo

O propósito desta subetapa é de realizar uma filtragem preliminar do conteúdo após ter sido convertido em texto sem formatação. O processo de conversão foi realizado separando o conteúdo página por página o que se mostrou o mais adequado na identificação dos elementos textuais tais como os cabeçalhos e rodapés das páginas. Desse modo a heurística adotada avalia o conteúdo que ocorre de forma repetida em todas as páginas a partir do início de cada página. Esse excerto, denominado de cabeçalho, é filtrado, portanto, eliminado no texto convertido. Um procedimento inverso, analisando a página a partir do seu final foi realizado para eliminar o rodapé comum em todas as páginas do documento.

Um outro processo de filtragem foi realizado, nessa subetapa, para eliminar as referências que contêm termos tais como: nome de autores e de obras; que muitas das vezes estão fora do tema central do documento. A heurística adotada considera que as referências estão postas na parte final dos documentos. Dessa maneira, busca-se identificar o excerto de texto partindo do fim em direção ao início limitando-se a percorrer até 30% do documento. O objetivo é de identificar a ocorrência do termo “referência”, “referências” ou “referenciais” de forma não sensível a maiúsculas ou minúsculas a fim de descartar conteúdo até esse ponto.

Ainda durante esse processo de filtragem preliminar foram identificados e convertidos todos os caracteres de acentuação, usuais na língua portuguesa, que são

representados através de caracteres de múltiplos *bytes*, transformando-os em caracteres simples e não acentuados, mas preservando a grafia original do texto das letras maiúsculas e minúsculas. Portanto, todos esses conteúdos considerados como ruidosos são filtrados do texto a ser transferido para a próxima subetapa.

#### 3.1.2.3 Segmentar a cadeia de caracteres em sentenças

Esta subetapa é realizada após os documentos estarem convertidos para o formato de uma cadeia de caracteres e filtrados preliminarmente. O objetivo é realizar o *parsing,* ou seja, processar a cadeia de caracteres extraída do documento a fim de separar as sentenças que o compõem.

A precisão desse processo é de fundamental importância, para que um erro nesse ponto não se propague nos processamentos posteriores. Conforme descrito por Mikheev (2002 p. 290), processar a divisão de um texto em sentenças, na maioria dos casos é uma tarefa simples, basta considerar como separador os caracteres: ponto final, ponto de exclamação, ponto de interrogação. Entretanto, existem algumas exceções como, por exemplo: quando o ponto final é utilizado entre números, em abreviações, ou ainda, quando forem ambos os casos ao mesmo tempo. Portanto, alguns cuidados devem ser tomados, pois um erro na separação das sentenças pode gerar falhas na identificação das EM.

Veja-se um exemplo em que um erro no processo de separação de sentenças leva à identificação incorreta de EM. Ao considerar as sentenças mostradas a seguir como sendo uma única sentença, a expressão “ciência da informação” poderia ser interpretada como uma EM. Enquanto que, na verdade, não existe esse sentido expresso no texto, pois as palavras ciências e da informação não estão conectadas semanticamente por estarem em sentenças distintas. O algoritmo proposto nesta tese considera a estrutura do texto, portanto, ao considerar o fato dos termos estarem colocados em sentenças distintas, não os considera como termos dependentes, ou seja, como uma EM. Veja-se:

> Melhorar o bem-estar da humanidade é uma tarefa das ciências.
> Da informação surge o insumo para a tomada de decisão.

Para tratar essas exceções utilizou-se uma estratégia parecida com a adotada por Mikheev (2002), a qual considera os contextos locais do documento e aplica um pequeno conjunto de regras para fazer a desambiguação. Entretanto, para o contexto deste trabalho, essas regras puderam ser relaxadas sem prejuízo para o resultado final. Portanto, durante esse processo de conversão do texto foi realizado ainda um tratamento *byte* a *byte* dos caracteres excluindo aqueles segundo um conjunto preestabelecido de regras para executar essas tarefas:

- São retirados o ponto final (.) e a vírgula (,) de valores numéricos. A heurística utilizada nesse caso busca identificar a presença desses caracteres colocados entre valores numéricos;
- São retirados o ponto final (.) utilizado para abreviação de palavras. A heurística utilizada nesse caso busca identificar qual é o caractere anterior e o posterior ao ponto final avaliando caso a caso se o ponto final é um delimitador de uma sentença ou apenas usado para abreviar uma palavra;
- São retirados o ponto final (.) que estiver dentro de parênteses;
- São retirados expressões tais como “[...]” ou “(...)”.

Todos esses conteúdos foram retirados a fim de evitar que o *parser* de separação de sentenças seja induzido a erros. Além disso, a cadeia de caracteres resultante dessa subetapa receberá durante o tratamento *byte* a *byte*, a substituição pelo caracter de espaço em branco para todos os *bytes* que estão fora da faixa de valores legíveis da tabela ASCII, tais como caracteres de controle para tabulação, quebra de linha, etc. Desse modo, são considerados como delimitadores das sentenças os seguintes caracteres: ponto final, ponto de interrogação e o ponto de exclamação. Por fim, esses mesmos caracteres considerados como separadores na segmentação das sentenças, além do hífen, são eliminados das sentenças.

### 3.1.2.4 Segmentar as sentenças em palavras

Nesta subetapa, o objetivo é separar as sentenças em palavras a fim de se criar o vocabulário de palavras ou de termos normalizados do léxico. *Tokenization* é o nome em inglês utilizado para essa tarefa. Essa parte do processo é fundamental para possibilitar a obtenção de boas respostas na hora de pesquisar pelos termos de busca. Ao realizar a quebra da sentença em palavras, existem vários pormenores que devem ser observados. O mais importante deles é que o algoritmo de normalização dos termos utilizado durante a indexação e a criação do vocabulário, seja idêntico ao aplicado no processo de identificação dos descritores do documento de referência utilizados pelo processo automatizado de busca.

Manning, Raghavan & Schütze (2009, p. 22-26) definem *tokenization* como sendo a tarefa de receber como entrada uma dada sequência de caracteres de um documento e separá-lo em partes chamada de *tokens*, e, ao mesmo tempo, descartar aqueles caracteres que indicam os pontos de separação. Os *tokens*, ou seja, os pedaços que foram segmentados, normalmente passam por um processo de normalização antes de se tornarem um termo do vocabulário. A normalização tem como objetivo reduzir o número de

entradas do dicionário e facilitar o processo de busca por palavras-chave, ao unificar diferentes formas de grafar um *token*. Esses mesmos autores descrevem várias situações em que é necessário dar um tratamento especial de normalização aos termos tais como lematização e radicalização. Como agravante essas ações de normalização são dependentes do idioma o que prejudica a capacidade de generalização do método. Portanto, nesta tese elas não serão utilizadas.

Os caracteres normalmente usados para indicar a separação das palavras, a vírgula, o hífen e o espaço em branco não podem ser considerados como separadores de forma irrestrita. Por exemplo: a vírgula pode ser utilizada para separar os números inteiros dos decimais no modelo europeu de representação numérica, ou os milhares no modelo saxão; o hífen pode estar sendo usado para dividir sílabas de uma palavra escrita no fim de uma linha, ou em palavras compostas que podem ser encontradas em diferentes grafias; no caso do espaço em branco, o problema ocorre quando ele é utilizado separando nomes próprios, pois nesse caso os termos não deveriam ser separados por terem um significado composto.

Para mitigar esses problemas utilizaram-se algumas estratégias descritas a seguir. No caso da vírgula ela será descartada, dessa forma as representações numéricas serão expressas somente por números sem os separadores. No caso do hífen, vejamos um exemplo de conteúdos tais como: infraestrutura[26], infra-estrutura[27] ou infra estrutura[28]. Ao fazer uma busca no Google pelos três termos encontram-se dois resultados diferentes. Ao pesquisar por infra-estrutura ou infra estrutura foram encontrados aproximadamente 3.960.000 *links,* enquanto que, ao buscar por infraestrutura foram retornados aproximadamente 3.420.000 respostas. Ou seja, essa ainda é uma questão em aberto. Neste trabalho iremos desprezar o hífen, dessa forma, palavras grafadas com hífen serão tratadas como um único termo, e o hifén das quebra silábicas, ao ser retirado, irá reagrupar a palavra. No caso do espaço em branco o problema ocorre nos conteúdos que são nomes próprios, tal como: Belo Horizonte, pois o sentido nesse caso deve ser dado pelas duas palavras juntas, e não como duas entradas distintas do vocabulário. Nesta tese esse problema é atenuado, pois se essas palavras forem relevantes no contexto do documento, se tornarão um bigrama e serão encontradas na consulta apenas se estiverem no documento em sequência. Portanto, serão considerados como separadores de palavras os seguintes caracteres: espaço em branco, dois pontos, ponto e vírgula, barra vertical, barra invertida, sinal de adição, multiplicação, divisão, igual, abre e fecha chaves, abre e fecha colchete, abre e fecha parenteses, aspas duplas, aspas e circunflexo.

[26] Conforme o novo acordo ortográfico da língua portuguesa válido a partir de 2009. Acesso em agosto 2011.
[27] Grafada de forma anterior ao acordo. Acesso em agosto 2011.
[28] Grafada de forma incorreta, porém passível de ser encontrada. Acesso em agosto 2011.

Conforme citado por Cintra (1985, p. 5) uma interpretação proficiente de um texto se dá em nível de blocos ou segmentos maiores de informação. Desse modo, as partes que compõem um texto têm cargas semânticas diferenciadas de acordo com a estrutura do texto. Portanto, separadas às palavras cabe agora identificar em qual categoria ela se enquadra. As categorias que elencamos foram: palavras com todas as letras em maiúsculas, palavras com a primeira letra em maiúsculas e palavras com todas as letras em minúsculas. Essa categorização foi adotada a fim de atribuir um peso diferente ao valor semântico das palavras contidas na sentença dependendo da sua forma de apresentação. Esse peso será definido através de um coeficiente estrutural (Ce) parametrizado fazendo com que a relevância de uma EM seja relativizada.

Em seguida, é realizado o processo de normalização básico, em que todos os termos são transformados em minúsculas. Pois dessa forma reduz-se a ambiguidade na busca.

### 3.1.2.5 Decodificar siglas

Uma prática muito comum da escrita, principalmente na científica, é a utilização de abreviações. Normalmente, na primeira aparição, os termos são mostrados por extenso com as letras que compõem a sigla em cada termo apresentadas em maiúsculas, seguidos pela sigla propriamente dita entre parênteses e com letras maiúsculas separadas ou não por ponto final. Partindo dessa premissa, nessa subetapa a meta é identificar as siglas a fim de montar uma tabela de siglas utilizadas a cada documento e acrescer ao texto uma parte por extenso para todas as vezes que a sigla ocorrer ao longo de uma sentença. Essa estratégia é importante de ser adotada, pois o conteúdo expresso no texto apenas como sigla, não seria interpretado como sendo EM. Enquanto que, na verdade, esse tipo de conteúdo é normalmente de alto teor semântico para expressar o sentido do documento, e ao ser colocado por extenso, dependendo de sua frequência de ocorrência, fará com que esse conteúdo se torne uma EM.

### 3.1.2.6 Retirar as stop words

Nesta subetapa, após quebrar o documento em uma sequência de palavras, executa-se uma nova filtragem. O objetivo é de não inserir no vocabulário, as palavras que aparecem com muita frequência em todos os documentos, e que, portanto, têm pouco poder de discriminação. Manning, Raghavan & Schütze (2009, p. 27) definem *stop words* como sendo palavras muito comuns que parecem ter pouco valor para selecionar documentos correspondentes. Essas palavras normalmente pertencem à classe dos artigos, preposições e algumas conjunções. Esses mesmos autores explicam que uma estratégia que pode ser usada para determinar a lista de *stop words* é contabilizar o número de vezes que cada

termo aparece na coleção de documentos, e verificar, muitas das vezes manualmente, qual a relevância semântica do termo em relação ao domínio dos documentos que estão sendo indexados. Aqueles considerados irrelevantes são incluídos na lista de *stop word*s. Eles ressaltam também que vem ocorrendo uma tendência crescente de redução do número de termos que compõem a lista de *stop word*. Enquanto inicialmente os sistemas de indexação mantinham na lista cerca 200 a 300 palavras, os sistemas mais recentes vêm adotando listas cada vez menores entre 7 a 12 palavras no caso do idioma inglês, sendo que a maioria das máquinas de busca web não utiliza essas listas.

Neste trabalho usar-se-á uma lista de *stop words*, pois, para o propósito de Busca Comparada baseada em EM, isso contribuirá de forma positiva. Afinal a definição dos termos da busca é realizada de forma automatizada. Por exemplo, o conteúdo: “ciência da informação”, tratado sem filtrar as *stop words* seria um trigrama, após filtrado seria transformado em um bigrama. Tanto o processo de indexação do léxico, quanto o da busca adotam a mesma conduta, isso trará um ganho. Optou-se também por não incluir automaticamente novos termos na lista tomando como base a frequência dos termos ocorridos no *corpus* utilizado. O apêndice B apresenta a lista de *stop word* utilizada.

Adicionalmente, após a retirada das *stop words* é verificado ainda o tamanho de cada termo retornado após realizar a quebra da sentença em palavras. Aqueles com apenas um caracter são descartados. Os termos resultantes são então encaminhados para a próxima fase.

### 3.1.3 Processar Termos

Para proporcionar um melhor entendimento do processamento dos termos pelo Server, esta etapa foi dividida em três subetapas apresentadas a seguir:

- Subetapa 1 – Indexar os termos;
- Subetapa 2 – Disponibilizar serviço de consulta.
- Subetapa 3 – Gravar os arquivos em formato texto (.txt).

Essas subetapas são executadas somente após ter sido realizada a etapa de converter o documento PDF em termos normalizados, descrita na seção 3.1.2. Elas são melhor detalhadas a seguir e complementadas com as informações de configuração do *software* Server descritas no Apêndice C.

### 3.1.3.1 Indexar os termos

O objetivo desta subetapa é construir uma lista invertida dos termos do vocabulário, sendo que para cada termo haverá um apontamento para todos os documentos nos quais ele é referenciado. Adicionalmente, utilizou-se a técnica *positional index* descrita por Manning, Raghavan & Schütze (2009, p. 41-43). Essa técnica consiste em adicionar na estrutura da lista invertida a(s) posição(ões), controlada(s) a partir de uma sequência numérica, contendo a posição em que o termo foi encontrado no documento. Ou seja, qual o número da sentença e qual o número da palavra dentro da sentença. Isto permite realizar buscas em que se deseja encontrar a distância entre os termos de uma expressão em uma mesma sentença, tal qual é necessário para a identificação das EM. Cabe ressaltar que, em tempo de busca, é necessário realizar a busca em separado de cada um dos termos da expressão, e a partir do resultado retornado para cada termo é que se torna possível verificar se eles são consecutivos, através de um processamento de alinhamento do posicionamento dos termos, para somente, então, verificar se são adjacentes.

A Figura 18 apresenta um esboço da estrutura de dados utilizada por essa técnica. Onde: {$t_1$, $t_2$, $t_3$, ..., $t_n$} representam os termos do vocabulário; {$d_1$, $d_2$, $d_3$, ..., $d_n$} representam os documentos; {$p_1$, $p_2$, $p_3$, ..., $p_n$} representam a posição da sentença e da palavra dentro da sentença em que um determinado termo foi encontrado em um documento; e, {$r_1$, $r_2$, $r_3$, ..., $r_n$} representam uma referência para o local onde o documento está armazenado.

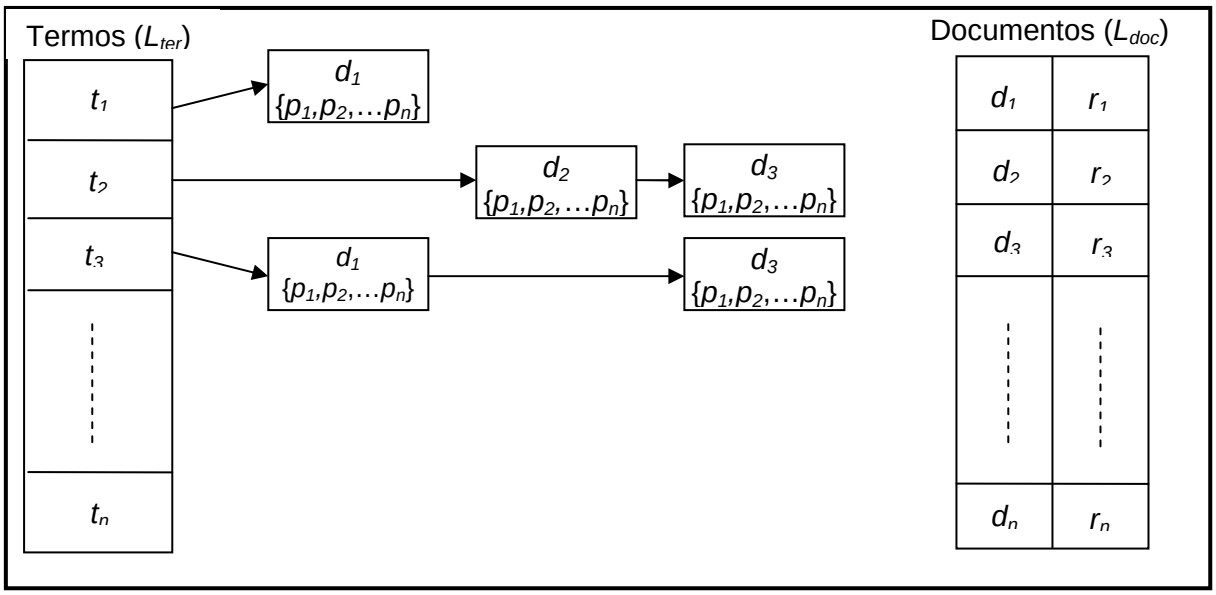



FIGURA 18 – Esboço da estrutura de dados utilizada na lista invertida com índice posicionado.
Fonte: Elaborada pelo autor.

Durante o processamento do Server para criação da estrutura apresentada na Figura 18, o algoritmo segue os passos descritos a seguir.

A cada documento do *corpus* existente na estrutura hierárquica de arquivos será incluído na lista de documentos $L_{doc}$ com uma referência do caminho no qual ele está fisicamente colocado. Isso ocorrerá logo após o documento ter sido processado pelas etapas definidas na seção 3.1.2, a qual tem como resultado uma coleção de termos numerados de forma sequencial pela sentença começando pelo número um até o número da última sentença e por ordem de termo dentro da sentença começando de um até o número do último termo de cada sentença. A meta seguinte é processar cada termo da coleção a fim de incluí-lo na lista de termos $L_{ter}$. Mesmo que o termo ocorra várias vezes no mesmo documento ou em documentos distintos, ele é incluído apenas uma vez nessa lista. Entretanto, para cada termo encontrado no documento serão criadas uma célula de memória $d_n$ e uma outra célula com a sua respectiva posição $p_n$, na qual ele ocorre na coleção de documentos. Cabe ressaltar que para os termos recorrentes em um mesmo documento é necessário criar apenas uma nova célula com o registro de cada posição $p_n$, considerando que $d_n$ é único para cada termo/documento.

Após todo o *corpus* ter sido processado, documento por documento, todos os termos estarão indexados na memória volátil do computador em uma estrutura de lista invertida e com registro das posições em que o termo foi encontrado no documento.

### 3.1.3.2 Disponibilizar um serviço de consulta

Essa subetapa é realizada ou não, de forma mutuamente exclusiva com a próxima etapa, dependendo de um parâmetro de configuração do Server. A função dessa etapa é disponibilizar um serviço de consulta através de um protocolo de comunicação entre os dois componentes de *software,* o Server e o Client. O protocolo de comunicação consiste no envio pelo Client de uma lista contendo todos os bigramas extraídos do documento de referência da busca e o do retorno da resposta dado pelo Server com uma referência do *link* para os documentos similares encontrados no *corpus*. Para cada bigrama será processada a busca de cada um de seus termos em separado. Os resultados obtidos serão analisados verificando se os termos de cada bigrama foram encontrados em uma mesma sentença de um mesmo documento e de forma adjacente. Nesse caso será computado o coeficiente de relevância, caso contrário esse item da resposta será descartado para que o próximo item possa ser analisado. Maiores detalhes serão descritos na seção 3.2.2 que apresenta as funcionalidades do componente de *software* Client.

### 3.1.3.3 Gravar os arquivos em formato de texto (.txt)

Conforme já descrito, essa subetapa é executada dependendo de um parâmetro de configuração do Server e de forma mutuamente exclusiva com a etapa anterior. Desse modo, em vez de disponibilizar um serviço de consulta, o Server pode ser configurado para executar apenas a funcionalidade de gerar um arquivo com o mesmo nome do documento original, mas com a extensão renomeada para (.txt). Portanto, nesse caso, a etapa de indexar termos, descrita na seção 3.1.3.1, não será executada. O arquivo convertido após passar pelas subetapas de normalização e filtragem dos termos consiste de um documento em formato texto puro, ou uma cadeia de caracteres não formatados. Ou seja, corresponde ao mesmo conjunto de termos a serem indexados em lista invertida. Esses arquivos gerados servirão de base para o processamento de extração das medidas de associação estatísticas realizadas pelo *software* NSP. Dessa maneira, a base textual utilizada para a extração dos bigramas é a mesma em todas as técnicas utilizadas nesta tese. A Figura 19 mostra um fragmento do arquivo “31.txt” após estar convertido para o formato (.txt). A Figura 20 mostra o mesmo documento no formato original em PDF.

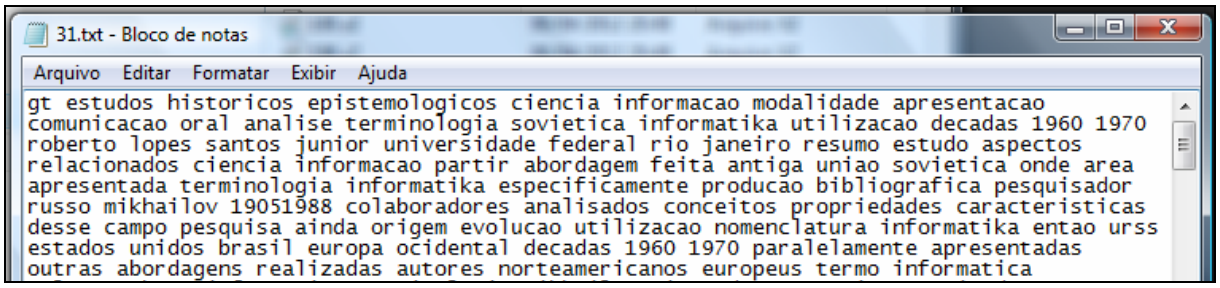
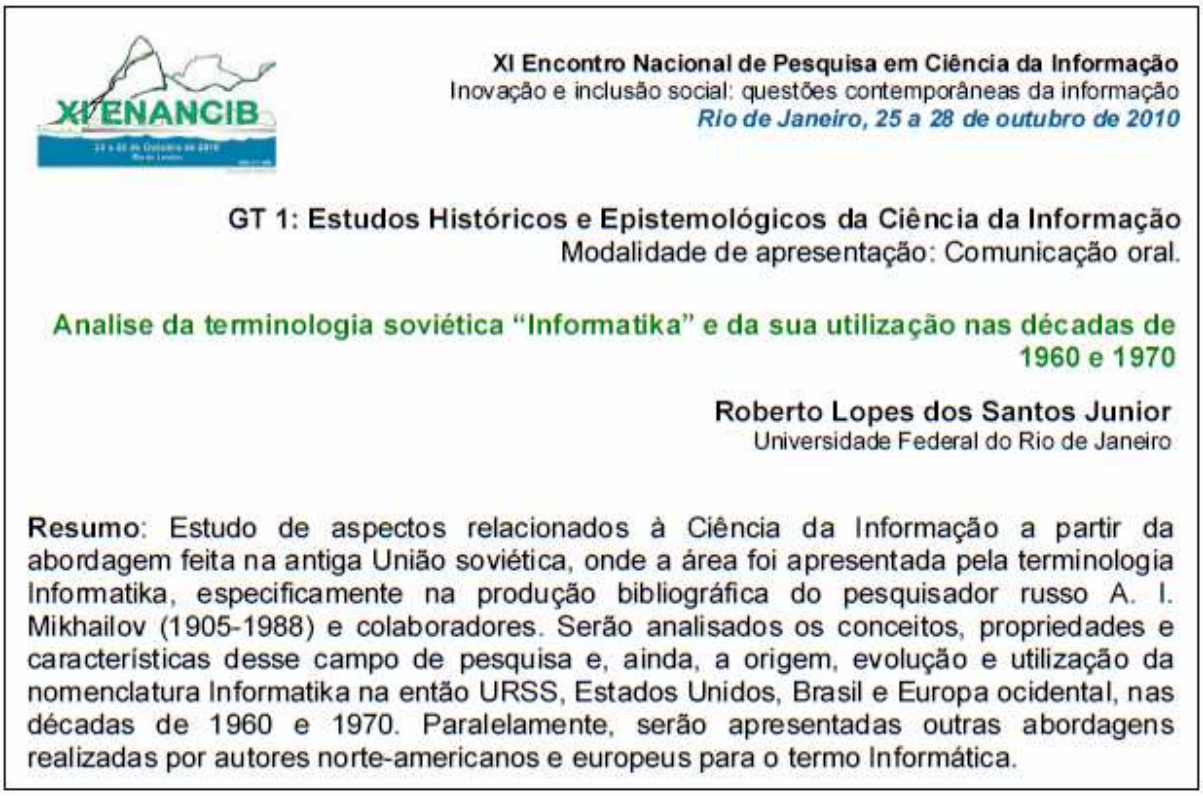



FIGURA 19 – Fragmento do arquivo em formato texto após a conversão.
Fonte: Elaborada pelo autor.


XI ENANCIB

XI Encontro Nacional de Pesquisa em Ciência da Informação
Inovação e inclusão social: questões contemporâneas da informação
*Rio de Janeiro, 25 a 28 de outubro de 2010*

GT 1: Estudos Históricos e Epistemológicos da Ciência da Informação
Modalidade de apresentação: Comunicação oral.

**Analise da terminologia soviética "Informatika" e da sua utilização nas décadas de 1960 e 1970**

**Roberto Lopes dos Santos Junior**
Universidade Federal do Rio de Janeiro

**Resumo**: Estudo de aspectos relacionados à Ciência da Informação a partir da abordagem feita na antiga União soviética, onde a área foi apresentada pela terminologia Informatika, especificamente na produção bibliográfica do pesquisador russo A. I. Mikhailov (1905-1988) e colaboradores. Serão analisados os conceitos, propriedades e características desse campo de pesquisa e, ainda, a origem, evolução e utilização da nomenclatura Informatika na então URSS, Estados Unidos, Brasil e Europa ocidental, nas décadas de 1960 e 1970. Paralelamente, serão apresentadas outras abordagens realizadas por autores norte-americanos e europeus para o termo Informática.

FIGURA 20 – Fragmento do documento "31.pdf" no formato original.
Fonte: Elaborada pelo autor.

## 3.2 Descrição da segunda fase

Esta fase tem como objetivo elaborar o *software* Client para converter os documentos utilizados como referência de busca. De forma semelhante ao processo de conversão realizado na seção 3.1, o documento é convertido do formato (.pdf) para texto puro (.txt), no qual os termos já estão normalizados. Esses termos são utilizados para atender a duas funcionalidades principais:

- Organizar os termos em uma estrutura de memória, que possibilite processar a extração das EM, para servirem de descritores da Busca Comparada através de requisições ao Server. Seu objetivo visa à identificação dos documentos similares existentes no *corpus*;
- Gerar um arquivo, no formato de texto, com o mesmo nome do arquivo original, mas com a sua extensão renomeada para (.heudet). Esse arquivo contém todos os bigramas extraídos dos documentos e as suas respectivas frequências observadas. O conteúdo desses arquivos servirá de base para o processamento da quarta fase, pois compara os resultados obtidos pelas diferentes técnicas

utilizadas no processo de extração das EM. Afinal, esses arquivos contêm as EM extraídas através da técnica Heudet proposta nesta tese.

Em ambas as funcionalidades dessa fase, os dados de entrada são o documento de referência em formato PDF. Portanto, esses documentos são submetidos ao mesmo processamento já descrito na primeira fase constituído pela etapa: “Converter documentos PDF em texto com termos normalizados”, já apresentado na seção 3.1.2.

### 3.2.1 Processar a Busca Comparada

Esta etapa tem como objetivo consultar, a partir de um documento de referência em formato PDF fornecido pelo usuário da busca, quais são os documentos correlatos existentes *corpus*. O processamento executado pelo *software* Client realiza uma busca comparando as EM encontradas no documento de referência, e expressas através de bigramas, com os documentos da coleção que estão organizados na estrutura de lista invertida. Desse modo, as requisições com os descritores (bigramas) são enviadas pelo Client através de um protocolo de comunicação Transmission Control Protocol / Internet Protocol (TCP/IP) estabelecido via rede com o serviço disponibilizado pelo Server. E através desse mesmo canal de comunicação, as respostas são retornadas ao Client. Para um melhor entendimento essa etapa foi dividida nas seguintes subetapas:

- Subetapa 1 – Receber o documento de referência da busca;
- Subetapa 2 – Converter os documentos PDF em termos normalizados;
- Subetapa 3 – Extrair as EM dos documentos (Heudet);
- Subetapa 4 – Enviar a requisição ao Server;
- Subetapa 5 – Apresentar o resultado da busca.

#### 3.2.1.1 Receber o documento de referência da busca

Nesta subetapa o objetivo é elaborar uma aplicação Web que sirva como interface do usuário final para processar a Busca Comparada. Para elaborar essa interface utilizou-se a linguagem PHP. O componente de *software* criado foi denominado “Buscomp”. Essa interface se encarrega e receber o caminho de acesso para o documento de referência seguido pela confirmação da requisicão realizada pelo usuário da busca. Confirmada a requisição, a aplicação se encarrega de fazer o *upload* do documento para o provedor de acesso da página e executar a chamada local da aplicação Client passando o documento como parâmetro de processamento.

Nesse experimento processa-se apenas um documento de referência. Entretanto, para adaptá-lo para lidar com mais de um documento de referência, basta concatenar os vários documentos na entrada. Desse modo, vários documentos podem ser processados como sendo um documento único. A Figura 21 apresenta um esboço da tela da interface utilizada.

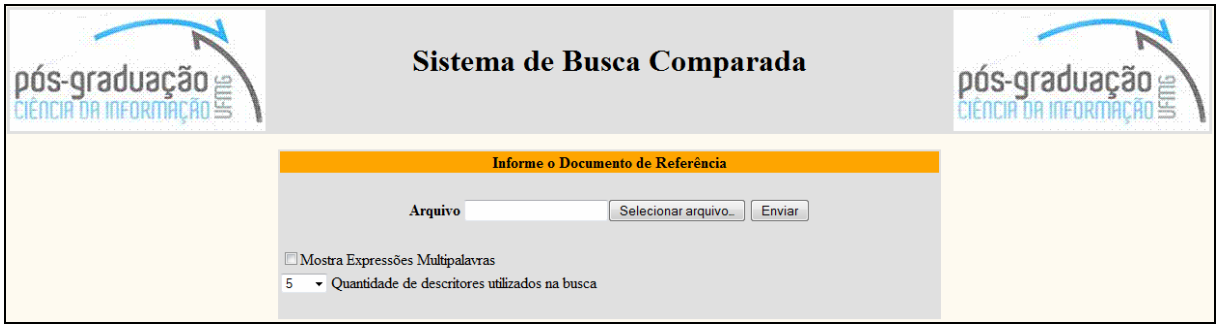



FIGURA 21 – Interface, onde é informado o documento de referência utilizado na Busca Comparada.
Fonte: Elaborada pelo autor.

O usuário deverá informar o caminho físico em que o documento está armazenado, ou clicar no botão “Selecionar arquivo”. Nesse caso, uma caixa de diálogo do Windows será aberta permitindo a escolha do arquivo que servirá de referência para a Busca Comparada. O usuário poderá ainda marcar as seguintes opções:

- o *checkbox* “Mostra Expressões Multipalavras”, o que permite visualizar na página de resposta quais foram os bigramas utilizados na busca;
- o *listbox* “Quantidade de descritores utilizados na busca”, que delimita a quantidade máxima de bigramas a ser considerada no processamento de busca.

Para realizar efetivamente a busca, basta clicar no botão “Enviar”, que o processamento de consulta será realizado retornando uma página com os documentos similares encontrados como resposta.

### 3.2.1.2 Converter os documentos PDF em termos normalizados

Em seguida cada documento a ser processado passará pela mesma subetapa comum já descrita na seção 3.1.2, portanto, não será descrita novamente.

### 3.2.1.3 Extrair as EM dos documentos (Heudet)

Nesta subetapa, o objetivo é receber os termos normalizados e identificados pelo número da sentença e pela posição do termo dentro da sentença e processá-los a fim de ordená-los em uma estrutura de dados em memória que permita a extração das EM. A

estrutura de dados em memória proposta pelo autor desta tese para viabilizar o processo de extração é mostrada na Figura 22.

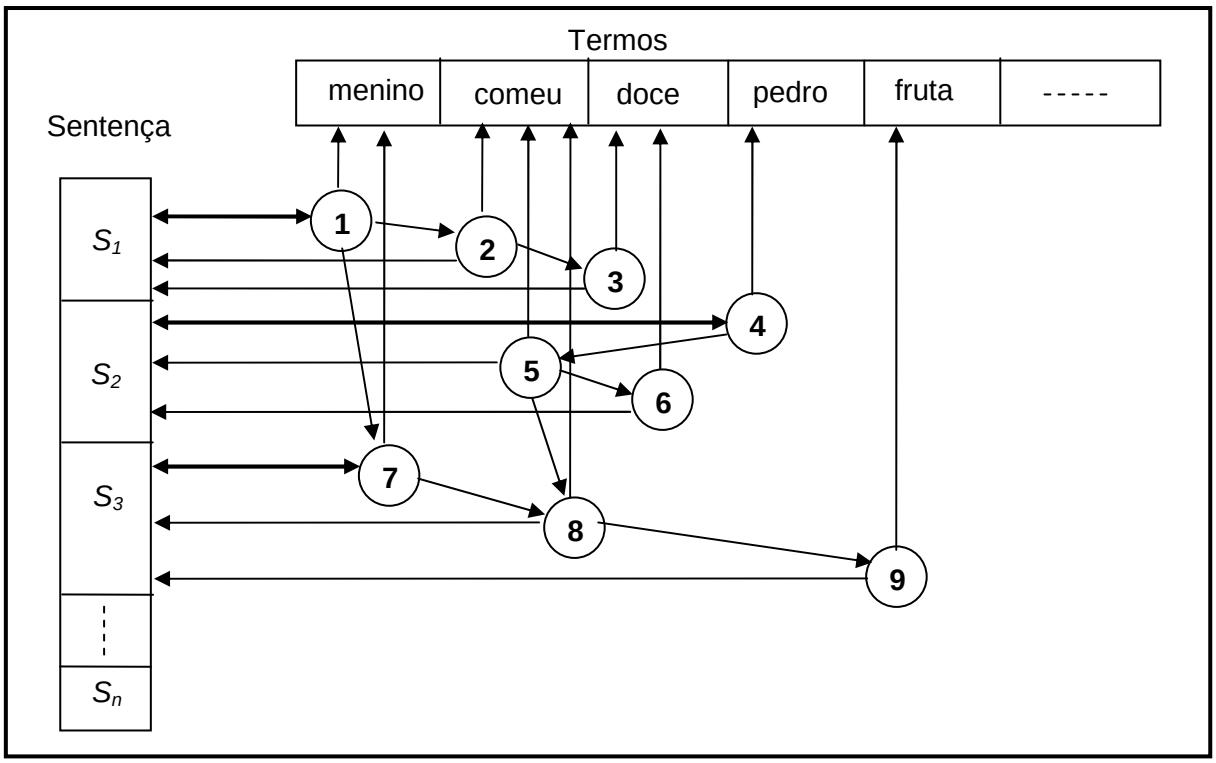



FIGURA 22 – Representação da estrutura de dados criada para extrair EM.
Fonte: Elaborada pelo autor.

Para entender como funciona essa estrutura, considera-se que o conteúdo do documento de referência $d_{ref}$ da busca é composto pelas sentenças $S_1$, $S_2$, $S_3$. Ou seja, $d_{ref}$ = { $S_1$, $S_2$, $S_3$ }, conforme mostrado em (3.1).

$$d_{ref} = \begin{matrix} S_1 \rightarrow \text{O menino comeu o doce.} \\ S_2 \rightarrow \text{Pedro comeu o doce.} \\ S_3 \rightarrow \text{O menino comeu a fruta.} \end{matrix} \quad (3.1)$$

Considera-se também, que, após executada a subetapa descrita na seção 3.1.2.2 no documento informado, tem-se como resultado o conjunto de termos normalizados $V$ = { $T_1$, $T_2$, $T_3$, $T_4$, $T_5$ }, conforme mostrado na Tabela 2.

TABELA 2 – Termos normalizados

| Identificação | Termos |
|---|---|
| $T_1$ | Menino |
| $T_2$ | Comeu |
| $T_3$ | Doce |
| $T_4$ | Pedro |
| $T_5$ | Fruta |

Fonte: Elaborada pelo autor.

E finalmente, considera-se que o conjunto de nós $N$ = { 1, 2, 3, 4, 5, 6, 7, 8, 9 } representa cada uma das nove palavras do texto a ser indexado de acordo com o exemplo proposto.

Como o processamento é realizado na ordem em que as sentenças são lidas, ao ler a sentença $S_1$ serão processados os termos $T_1$, $T_2$ e $T_3$ referenciados pelos três nós 1, 2 e 3 respectivamente. Ao ler a sentença $S_2$ serão processados os termos $T_4$, $T_2$ e $T_3$ referenciados pelos nós 4, 5, 6 e, assim, sucessivamente.

Após processadas todas as sentenças, a estrutura proposta permite identificar quais são as frases existentes na cadeia de caracteres e também em quais sentenças um determinado termo ocorre. Dessa forma, para extrair as EM, o algoritmo percorre as sentenças verificando para cada palavra, que ainda não foi processada, quais são as suas adjacentes. Em seguida, verifica a frequência com que a repetição dos termos adjacentes ocorre. Os termos com a frequência de repetição maior ou igual a uma quantidade informada em parâmetro (Qr) serão adicionados na lista dos bigramas a serem usados como descritores da Busca Comparada. Os demais são descartados. Um pseudo-código com os passos desse processo é mostrado em (3.2).

```
Enquanto (houver sentenças) faça
   Palavra = proxPalavra(Sentença)
   Enquanto (houver Adjacente) faça
      Adjacente = procAdjacente()
      Se nãoProcessada(Palavra)
         totAdjacentes = contaAdj(Palavra)
         Se totAdjacentes >= Qr
            Inclui(Palavra, Adjacente)
         Fimse
      Fimse
   FimEnquanto
FimEnquanto
```

(3.2)

Um ponto que deve ser destacado é que, apesar de esse processamento extrair apenas bigramas, não significa que expressões com *n*-gramas não sejam consideradas. Na prática, o processamento pode lidar com qualquer quantidade de termos consecutivos que tenham uma frequência observada igual ou acima da quantidade definida em parâmetro. Isso pode ser feito, pois qualquer conjunto de *n*-gramas pode ser transformado em pares de bigramas. No exemplo a seguir o pentagrama “Universidade Federal de Minas Gerais” é transformado em 3 bigramas: “Universidade Federal”, “Federal Minas”, “Minas Gerais”. O termo “de” é descartado por ser uma *stop word.*

### 3.2.1.4 Enviar a requisição ao Server

Nessa subetapa, a lista de bigramas produzida visa expressar a semântica do documento de referência utilizado na busca do usuário. Ao enviar a lista de bigramas ao Server, eles são utilizados como descritores de busca visando encontrar os documentos similares cujos termos ocorram de forma adjacente nos documentos do *corpus*. Ou seja, será estabelecido um processo de troca de mensagens entre as aplicações Server e Client através de um protocolo específico de comunicação de rede. A Figura 23 apresenta um esboço desse protocolo utilizado, e que é estabelecido a partir de um serviço de "escuta" disponibilizado pelo Server em $S_0$. Logo após de a aplicação Client identificar os bigramas, ela envia uma requisição $C_1$ ao Server contendo a lista identificada.

No exemplo mostrado, essa lista está limitada a cinco bigramas. O Server então responde confirmando o recebimento da mensagem em $S_2$. Como o *buffer*[29] de transmissão de mensagem está delimitado a 1024 *bytes*, dependendo do tamanho da lista de bigramas, esse processo de envio será refeito até que toda a lista tenha sido enviada para o Server. Ao fim do envio da lista, o Client enviará uma mensagem de aviso, conforme mostrado em $C_2$. Tendo o Server recebido toda a lista, inicia-se o processo de pesquisa dos bigramas no *corpus.* A lista será quebrada em partes de termos par a par. Em seguida, cada um dos termos será pesquisado no *corpus* produzindo uma lista com as respostas dos documentos em que os termos foram localizados, conforme mostrado em $C_3$. A resposta é produzida em formato HTML para ser exibida pela interface de consulta Web. Em seguida, o Client envia uma confirmação de resposta em $C_3$ e o Server envia uma mensagem de fim de comunicação $S_3$.

Da mesma forma que ocorre no envio da lista de bigramas, o tamanho da resposta também está limitado a 1024 *bytes.* Portanto, dependendo do tamanho da resposta podem ocorrer várias trocas de mensagens até que uma mensagem $S_3$ seja enviada, colocando finalmente o Server em processo de "escuta" para atender a uma nova requisição de consulta.

[29] Buffer – Área de memória é utilizada para escrita e leitura de dados contida no pacote TCP.

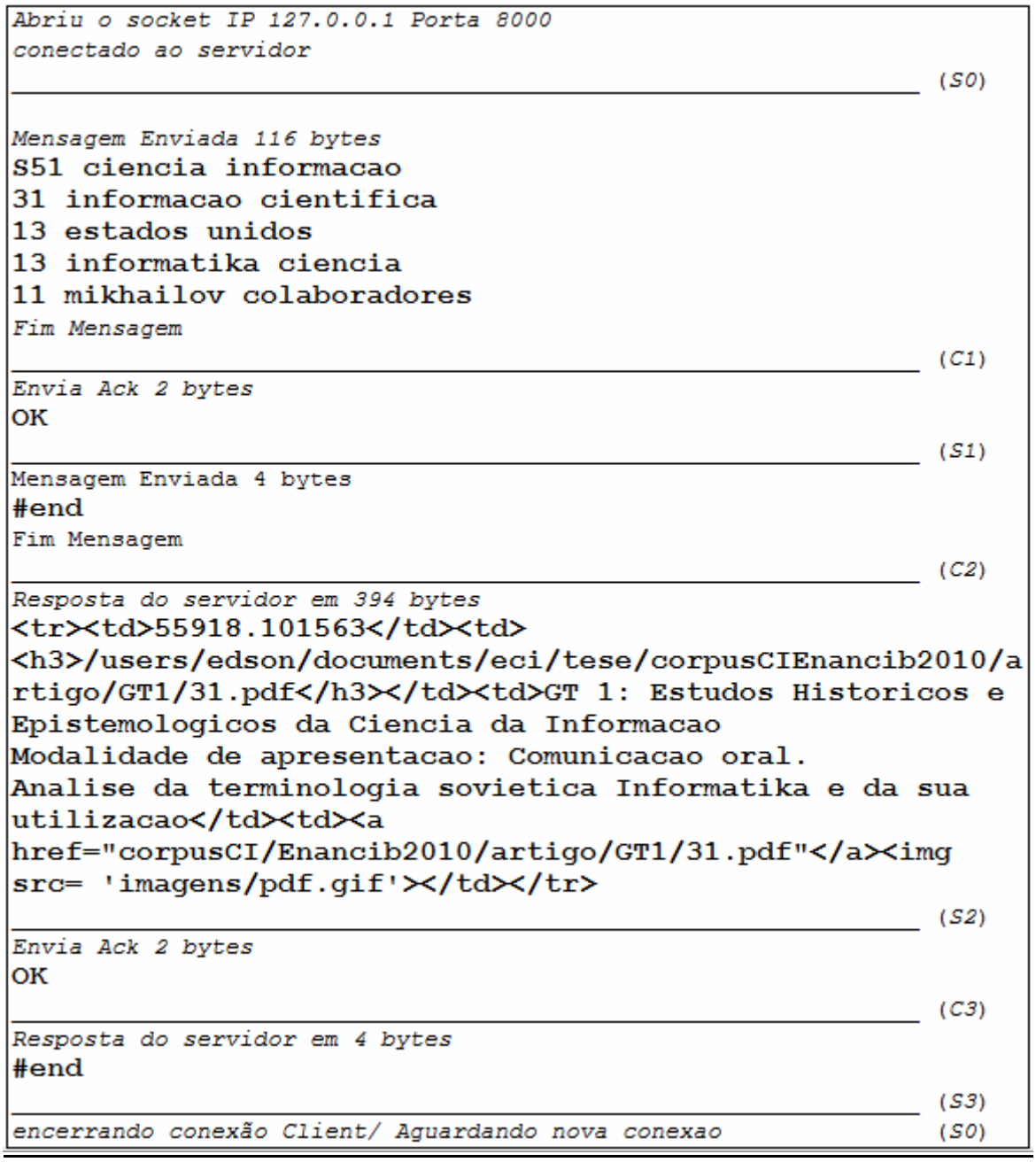



FIGURA 23 – Esboço do protocolo de comunicação entre o Server e o Client.
Fonte: Elaborada pelo autor.

Após ter-se apresentado uma visão geral do protocolo de comunicação entre esses componentes de *software,* cabe ainda apresentar mais alguns detalhes de como ocorre o processo de casamento dos bigramas fornecidos com os termos existentes nos documentos do *corpus*. Na prática, a lista invertida mantida na memória é composta por uma lista de termos mantida em uma estrutura de *hashing*[30]. Sendo que cada termo aponta para uma

[30] Hashing – É uma estrutura de dados que permite acesso rápido ao mapear chaves alfanuméricas em um valor numérico que indica a posição de memória do conteúdo de dados correspondente.

lista encadeada ordenada pelo número do documento, e cada célula da lista de documentos aponta para uma lista encadeada ordenada pelo número da sentença e pela posição do termo dentro da sentença na qual ele foi encontrado. Essa estrutura de memória, dadas suas características, permite o acesso direto a cada um dos termos do bigrama $T_1$ e $T_2$. Dessa forma, é possível fazer um caminhamento pela estrutura buscando primeiramente encontrar ambos os termos em um mesmo documento. Ou seja, é realizado um processamento de alinhamento dos termos. Portanto, caso o número do documento de $T_1$ seja menor que o de $T_2$ ocorre um deslocamento de $T_1$ para a próxima ocorrência da lista de documentos. Caso contrário, $T_2$ seja menor que $T_1$ ocorre um deslocamento de $T_2$ para a próxima ocorrência da lista de documentos, até que os documentos sejam iguais. Tendo sido encontrados os termos em um mesmo documento, é necessário verificar se eles estão em uma mesma sentença, para em seguida verificar se eles são adjacentes. Ou seja, se a distância entre $T_1$ e $T_2$ for igual a um e a sentença e o documento forem os mesmos, significa que ocorreu o casamento dos termos do bigrama com os encontrados no documento. Nesse caso, o coeficiente de relevância será contabilizado para o documento em questão. Caso contrário, um novo caminhamento deverá ocorrer verificando novamente o alinhamento dos termos. Os termos cuja sequência encontrada não forem adjacentes serão descartados. Portanto, não serão contabilizados. A Figura 24 apresenta um esboço do log de processamento, que dá uma ideia de como o processamento descrito anteriormente, ocorre.

```
Pesquisando o termo [0-politica] da Expressão Multipalavra
Pesquisando o termo [1-indexacao] da Expressão Multipalavra
Termo [0-politica] alinhado no docto 19
Termo [1-indexacao] alinhado no docto 19
     Termo [0-politica] alinhado na oracao 3
     Termo [1-indexacao] alinhado na oracao 3
          Termo [0-politica] na oraçao 3 na posição 2
          Termo [1-indexacao] na oraçao 3 na posição 3
               Frequencia de [politica] no docto 19 = 17
               Frequencia de [politica] no corpus = 31
               Termo [0-politica] de peso = 2 coef =  0.028162
               Frequencia de [indexacao] no docto 19 = 31
               Frequencia de [indexacao] no corpus = 12
               Termo [1-indexacao] de peso = 2 coef =  0.132338
```

FIGURA 24 – Log do processamento do alinhamento de busca dos termos.
Fonte: Elaborada pelo autor.

O trecho do log mostrado na Figura 24 é realizado no processo de alinhamento dos termos “política” e “indexação”. Como pode ser visto, esses termos foram localizados no documento número 19 na terceira oração/sentença do documento. Em seguida, é feita a verificação da posição dos termos dentro da sentença. Também pode ser observado que

eles foram localizados respectivamente nas posições 2 e 3. Portanto, os termos são adjacentes, e serão contabilizados no cálculo do coeficiente de relevância.

Uma vez processados todos os bigramas ter-se-á como resultado uma lista com todos os documentos em que houve o casamento entre os termos e os documentos do *corpus.* Sendo que cada documento terá o seu respectivo coeficiente de relevância calculado. Essa lista será finalmente ordenada de forma descrescente pelo coeficiente de relevância e aqueles documentos que tiverem o valor do coeficiente abaixo de um limiar definido através de parâmetro serão descartados. Portanto, são descartadas as respostas menos relevantes. Desse modo, será utilizado um ponto de corte em que somente os documentos com resultados superiores ao percentual informado em relação ao valor do maior coeficiente encontrado pela busca é que serão apresentados como resposta.

Esse processamento pode ser melhor entendido observando o algoritmo mostrado em (3.3), considerando:

- *C corpus* contendo os documentos.
- *B* é o conjunto de bigramas extraídos do documento de referência da busca.
- $Ce_a$ e $Ce_b$ é coeficiente estrutural do termo “a” e do termo “b” respectivamente.
- $B = \{(t_{1a,}\, t_{1b}), (t_{2a,}\, t_{2b}), ..., (t_{na,}\, t_{nb})\}$ – Bigramas formados pelos *n* pares de termos.
- $R_a = \{(d_{1a}, s_{1a}, p_{1a}), ..., (d_{na}, s_{na}, p_{na})\}$ – Respostas da busca realizada do *i*-iésimo termo $t_{la}$ na coleção de documentos *C.* A qual retorna as triplas contendo os termos encontrados: *d* = documento, *s* = sentença, *p* = posição.
- $R_b = \{(d_{1b}, s_{1b}, p_{1b}), ..., (d_{nb}, s_{nb}, p_{nb})\}$ – Mesmo que o anterior, só que referente ao termo “*b”* do bigrama.

Desse modo, as triplas de cada um dos termos “*a” e “b”* do bigrama são comparadas a fim de verificar se são adjacentes.

```
1  para x de 1 ate n faça
2     Ra = busca(txa, C)
3     Rb = busca(txb, C)
4     repita
5        se (dxa < dxb) então
6           próximo dxa
7        senão
8           se (dxa > dxb) então
9              Próximo dxb
10          fimse
11       fimse
12    até dxa = dxb
13    repita
14       se (sxa < sxb) então
15          Próximo sxa                                          (3.3)
16       senão
17          se (sxa > sxb) então
18             Próximo sxb
19          fimse
20       fimse
21    até sxa = sxb
22    se (pxa adjacente pxb) então
23        Pesodoc[I] = Pesodoc[I] + txa * Cea + txb * Ceb
24    fimse
25 fimpara
26 ordena (Pesodoc)
27 mostraRelevantes(doctos)
```

Cabe ressaltar que, diferentemente das ferramentas convencionais de busca que realizam o cálculo do coeficiente de relevância baseado no casamento de termos, nesse experimento, o cálculo está baseado apenas no casamento de bigramas. Outro aspecto importante é o cálculo do coeficiente de relevância poder ser obtido através de três diferentes técnicas, sendo que a definição de qual será utilizada cada momento pode ser determinado através de um parâmetro de processamento. Portanto, a linha de número 23 do pseudo-código mudará dependendo que qual técnica estiver parametrizada num dado processamento. A seguir são apresentadas informações sobre as técnicas utilizadas para calcular o coeficiente de relevância ($W_{docto}$) que estão implementadas no protótipo *software* elaborado pelo autor.

A primeira é a TF-IDF adaptada, obtida através da expressão mostrada em 3.4. Onde:

- $BF_{docto}$ representa a frequência do bigrama nos documentos do *corpus*;
- *N* representa o total de documentos do *corpus*;

- *n* representa o número de documentos que contém os termos do bigrama de forma adjacente.

$$W_{docto} = \sum_{B \in q} BF_{docto} * \log_{10}\left(\frac{N}{n}\right) \quad (3.4)$$

Essa técnica calcula o peso do documento $W_{docto}$ através do somatório da frequência dos bigramas pertencentes à consulta *q*, multiplicados pelo logaritmo na base dez da razão entre o total de documentos da coleção pela quantidade de documentos da coleção que contêm o bigrama. Ou seja, quanto maior a frequência do bigrama e menor a ocorrência dele em documentos distintos, maior é a sua relevância. Desse modo, são considerados a frequência de ocorrência do bigrama e o inverso de sua frequência que é computado para relativizar o quão relevante o bigrama é para discriminar o documento no *corpus.* Ou seja, bigramas que ocorrem em muitos documentos do *corpus* produzirão como resultado do logaritmo um valor muito pequeno que, multiplicado ao valor do primeiro termo da fórmula, a frequência do bigrama, torna-o menos significante no resultado final do somatório. Enquanto bigramas raros terão um valor maior de IDF que multiplicado ao termo TF, produzirá uma maior parcela a ser agregada no valor de relevância calculado.

A segunda técnica que pode ser utilizada é a Cosine Similarity Vector (CSV) adpatada. O valor da similaridade pode ser obtido através da expressão mostrada em (3.5).

$$Similaridade(Q,D) = \cos\Phi = \frac{W_{docto} * W_{consulta}}{\sqrt{W_{docto}^2 * W_{consulta}^2}} \quad (3.5)$$

Onde:

- $W_{docto}$ representa o peso do documento, apurado pela expressão mostrada em (3.4);
- $W_{consulta}$ representa o peso da consulta, apurado pela expressão mostrada em (3.6).

$$W_{consulta} = \sum_{B \in q} BF_{consulta} * \log_{10}\left(\frac{N}{n}\right) \quad (3.6)$$

Essa técnica apura o grau de relevância baseado no coseno do ângulo formado entre o vetor que representa os descritores da consulta com cada um dos vetores que

representam os documentos do *corpus* em um espaço *n*-dimensional. Desse modo, cada bigrama extraído do documento de referência é colocado em um eixo do espaço *n*-dimensional, tendo sua magnitude relacionada com um peso atribuído a ele. Uma das possibilidades de atribuição do valor do peso, dentre outras, pode ser dada pela sua frequência de ocorrência. Portanto, o modelo convencional CSV, no qual cada termo é representado em um eixo, neste estudo, foi adaptado para lidar com os bigramas. A partir dessa representação do documento de referência é que os bigramas serão comparados com aqueles que representam os documentos do *corpus.* Em cada documento que houver o casamento de pelo menos um dos bigramas descritores da consulta o seu peso será computado. O resultado final do coeficiente calculado para cada documento é obtido pelo somatório dos casamentos dos bigramas da consulta. O valor resultante é finalmente normalizado resultando em um domínio de respostas com valores que podem variar de 0 até 1. Sendo que, quanto mais próximo de 1, mais similar o documento é da consulta.

A terceira possibilidade de calcular a relevância é realizada pela técnica BM25 adaptada mostrada em (3.7). Dentre as variações propostas, essa foi a formulação escolhida, pois ela permite capturar também o peso da frequência do bigrama no documento de referência. Desse modo, quanto maior for a frequência do bigrama encontrado no documento de referência (na consulta), maior participação ele terá no cálculo do resultado final do coeficiente de relevância.

$$BM25 = \sum_{t \in q} \log\left[\frac{N}{df_t}\right] * \frac{(k_1+1)tf_{td}}{k_1\left[(1-b)+b*\left(L_d / L_{avg}\right)\right]+tf_{td}} * \frac{(k_3+1)}{k_3+tf_{tq}} \qquad (3.7)$$

É a partir do valor apurado nesses cálculos que os documentos da resposta são ordenados por relevância. Finalmente, o Server agrega no conteúdo da resposta uma formatação HTML para retornar ao Client. Conforme se verá na próxima seção.

### 3.2.1.5 Apresentar o resultado da busca

Nessa subetapa, o Client receberá do Server a resposta da busca contendo uma referência de acesso, um *link,* para todos os documentos que foram considerados como similares. Uma página com essas respostas por ordem de relevância será apresentada permitindo ao usuário da consulta visualizar o documento completo a partir de um clique em sua referência. A Figura 25 apresenta um esboço da tela de resposta.

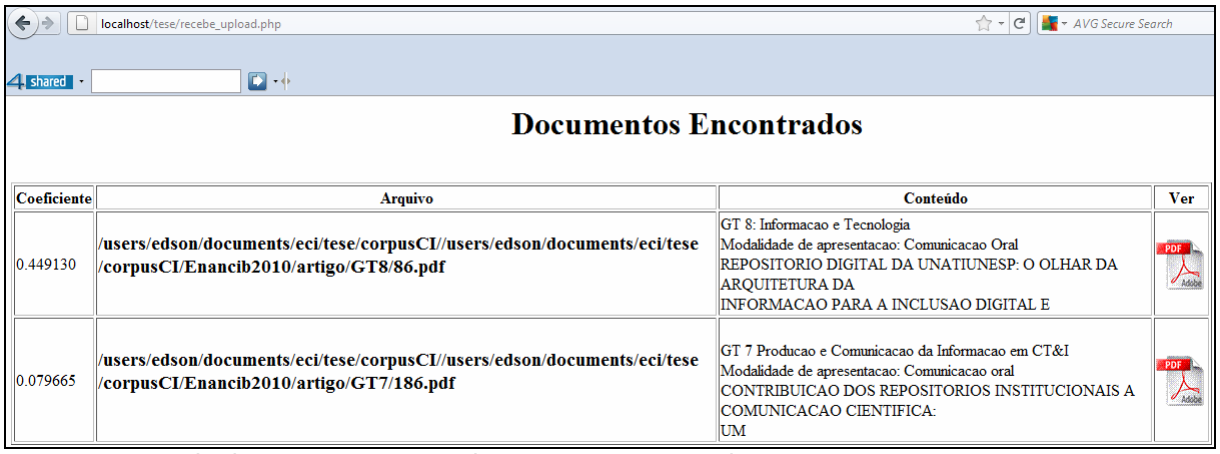


FIGURA 25 –Tela de resposta com os documentos encontrados.
Fonte: Gerada pelo software Buscacomp elaborado pelo autor.

Como podem ser observadas na tela, quatro colunas são apresentadas na interface de resposta: o coeficiente de similaridade, o caminho com o endereço físico do documento no *corpus*, os primeiros duzentos caracteres do texto após ser convertido e filtrado parcialmente e um ícone para acessar o documento na íntegra.

### 3.2.2 Gravar as EM extraídas em arquivos (.heudet)

Esta etapa tem como objetivo criar, a partir de cada documento (PDF) do *corpus* um arquivo com o mesmo nome do documento original, mas com a extensão renomeada para (.heudet). Esse arquivo conterá os bigramas extraídos e sua respectiva frequência observada. A geração desse arquivo poderá ocorrer de duas maneiras: ao executar uma Busca Comparada através da tela de interface do protótipo de *software* proposto, desde que definido em parâmetro a sua criação; ou através da execução de um programa denominado GeraEM, o qual automatiza a chamada de execução do processo de extração de EM baseado em um lista de documentos pré-definida. Portanto, para melhor descrever essa etapa foi dividida nas seguintes subetapas:

- Subetapa 1 – Gerar arquivo a partir do documento de referência da busca;
- Subetapa 2 – Gerar arquivo a partir de uma lista de documentos.

#### 3.2.2.1 Gerar arquivo a partir do documento de referência da busca

Esta subetapa tem como objetivo receber a lista de bigramas gerados na etapa anterior e gravar o seu conteúdo em um arquivo no disco. O arquivo é gravado na mesma pasta do sistema de arquivo onde está localizado o arquivo original (.pdf) e com o mesmo nome, mas com a extensão renomeada para (.heudet).

Esse processamento do Client para gerar o arquivo com os bigramas é semelhante ao processamento de interação com o componente de *software* PHP Buscomp. Só que nesse caso, o Client é executado sem que haja necessidade de se estabelecer comunicação com o Server. Um parâmetro de entrada é dado na chamada da execução do Client a fim de se estabelecer qual será a forma de execução: realizar a busca estabelecendo comunicação com o Server ou apenas gerar o arquivo de extensão (.heudet) com a lista dos bigramas extraídos.

Um exemplo é o arquivo gerado, mostrado na Figura 26. Conforme pode ser observado, são geradas três colunas separadas por um espaço em branco. A primeira com a frequência do total de ocorrência em que os termos do bigrama aparecem juntos no documento. A segunda e a terceira coluna contêm o primeiro e o segundo termo do bigrama respectivamente.

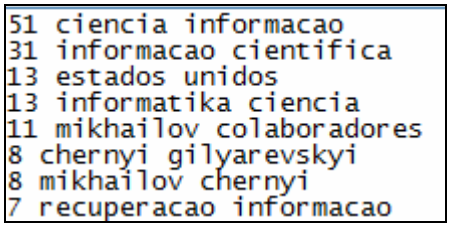



FIGURA 26 – Fragmento do arquivo (.heudet).
Fonte: Gerada pelo *software* Client.

### 3.2.2.2 Gerar arquivos a partir de uma lista de documentos

Esta subetapa tem como objetivo extrair os bigramas de todos os documentos do *corpus* (.pdf) e gravar esse conteúdo em arquivos. Esses arquivos contendo as EM são utilizados na quarta etapa da metodologia com o intuito de comparar as EM obtidas pelo algoritmo determinístico proposto pelo autor, com o resultado produzido pelas treze medidas de associação estatíticas produzidas pelo pacote NSP.

Para facilitar o processo de geração desses arquivos com as EM foi elaborado um componente de *software* em C++, denominado GeraEM. Esse aplicativo processa uma lista com os documentos (.pdf) existentes no *corpus* e executa a chamada do componente de *software* Client uma vez para cada arquivo da lista de documentos. Desse modo, ele automatiza a chamada de execução do programa Client passando como parâmetro de entrada a forma de processamento e o nome do documento a ser processado. Como resultado, para cada arquivo processado será gravado na mesma pasta um arquivo com o mesmo nome do documento original, mas com a extensão (.heudet). Esses arquivos gravados são utilizados apenas na quarta fase da metodologia.

## 3.3 Descrição da terceira fase

O objetivo desta fase é extrair de todos os arquivos do *corpus* as EM, utilizando um conjunto de treze medidas de associação estatísticas, produzindo para cada documento treze arquivos com os bigramas extraídos, sendo um arquivo para cada uma dessas técnicas. É importante ressaltar que esse processo estatístico de extração trabalhará com os mesmos documentos do *corpus* após terem sido convertidos para o formato texto, extensão (.txt). Portanto, o conteúdo textual processado é o mesmo que foi utilizado no processo de determinístico proposto pelo autor desta tese e descrito na seção 3.2.2.2. De modo que os resultados com os bigramas extraídos pelas quatorze técnicas distintas possam ser comparados na quarta fase.

### 3.3.1 Extrair as EM através do pacote NSP

Para realização dessa fase utilizar-se-á o *software* Ngram Statistics Package (NSP) proposto por Pedersen et al. (2011) sobre General Public Licence (GNU) disponível na web. O NSP permite a extração de *n*-gramas composto por de dois a quatro termos. Nesse experimento foi utilizado apenas a extração de bigramas, tendo em vista que o *software* utiliza o mesmo critério que transforma *n*-gramas em um conjunto de *n* – 1 bigramas, tal qual adotado pelo processamento no *software* Client.

O NSP é composto por um conjunto de componentes de *software*, sendo que cada um implementa um tipo de medida de associação estatística. O NSP é escrito em Perl[31] e roda no sistema operacional Linux. A principal proposta desse software é de auxiliar na identificação de *n*-gramas, *collocations* e palavras associadas tendo como base apenas arquivos em formato de texto puro. O NSP faz uso do conceito de expressões regulares que possibilita uma boa flexibilidade no processo de tokenização. Além disso, ele trabalha com diversas medidas de associação distintas que podem ser utilizadas para identificação de EM. O pacote do *software* distribuído consiste em dois programas núcleos e três utilitários principais.

O programa count.pl, utiliza como entrada um arquivo de texto puro e gera um arquivo de saída com uma lista de todos os *n*-gramas identificados e com as suas respectivas frequências de ocorrência por ordem descendente.

[31] Perl é uma linguagem de programação multiplataforma.

O programa statistic.pl utiliza o arquivo gerado na saída pelo count.pl e roda de cada vez uma medida de associação estatística informada pelo usuário para calcular um *score* para cada *n*-grama. Os *n*-gramas e os seus respectivos *scores* são gravados em um arquivo de saída também em ordem decrescente. O *score* calculado pode ser usado para decidir se existe ou não evidência suficiente para rejeitar a hipótese nula (que o *n*-grama não é uma EM).

Os programas utilitários, que são instalados na pasta bin/utils do pacote, utilizam, na entrada, as saídas geradas pelos programas núcleos count.pl e statistic.pl. Dentre eles, destaca-se o rank.pl que utiliza como entrada dois arquivos gerados pelo statistic.pl e calcula o coeficiente de correlação de Spearman dos *n*-gramas, que são comuns para ambos os arquivos. Tipicamente esses dois arquivos devem ser produzidos pela execução do programa statistic.pl, a partir de um mesmo arquivo origem gerado pelo count.pl, só que utilizando duas medidas estatísticas diferentes. O valor de saída de rank.pl pode ser usado para medir quão similares essas duas medidas são. Um valor próximo de 1 indica semelhança e próximo de -1 indica que são opostos.

Para um melhor entendimento dessa fase ela foi dividida em duas etapas descritas a seguir:

- Etapa 1 – Converter os arquivos (.txt) para (.count);
- Etapa 2 – Converter os arquivos (.count) para cada uma das medidas NSP.

A seguir apresenta-se uma explicação detalhada de cada uma das etapas.

#### 3.3.3.1 Converter os arquivos (.txt) para (.count)

O programa count.pl é o responsável por identificar os bigramas e as respectivas frequências de ocorrências para cada um de seus termos. Ele recebe como entrada o arquivo de texto (.txt) e trabalha para identificar os *tokens* sendo que cada *token* corresponde a uma sequência contígua de caracteres que combina com um conjunto de expressões regulares, que podem ou não, ser informadas pelo usuário, a fim de realizar o processo de tokenização. Dado um arquivo texto e um conjunto de expressões regulares, o texto é tokenizado, isso é, quebrado em *tokens*. Dessa forma, cada palavra que compõe o texto, a qual é considerada como sendo um *token,* fica delimitada por dois caracteres separadores. Para que esse processamento possa ser realizado, os caracteres de *new line* do texto são substituídos por espaços. Apesar de esse comportamento ser *default*, ele pode ser modificado.

Ao executar o count.pl é possível configurar uma série de opções para remover partes do texto, retirar *stop words*, etc. Entretanto, essas opções são desnecessárias de ser parametrizadas neste experimento, tendo em vista que os arquivos utilizados como entrada já foram previamente normalizados durante o processamento da primeira fase da metodologia materializado no documento com extensão (.txt). O arquivo produzido na saída pelo processamento do count.pl é gravado na mesma pasta do arquivo de entrada, com o mesmo nome do arquivo original, mas com a sua extensão renomeada para (.count). Esse arquivo tem uma estrutura conforme pode ser visto no exemplo da Figura 27 mostrada a seguir.

| Número da linha | Conteúdo do arquivo |
|---|---|
| 1 | 5 |
| 2 | ciencia<>informacao<>2 3 4 |
| 3 | sistemas<>informacao<>2 2 4 |
| 4 | banco<>dados<>3 3 5 |
| 5 | ciencia<>computacao<>1 3 1 |
| 6 | processamento<>dados<>2 1 5 |

FIGURA 27 – Conteúdo da saída produzido pelo count.pl.
Fonte: gerada pelo NSP.

Na primeira linha, é informado um número com a quantidade de bigramas extraídos. A partir da segunda linha, aparecem os bigramas extraídos demarcados pelo símbolo diamante “<>” seguidos de três números separados por espaço em branco. O primeiro desses números informa a quantidade de vezes que o *n*-grama ocorreu no arquivo de entrada. No exemplo na linha dois, o bigrama “ciencia informacao” ocorreu duas vezes. O segundo número, nessa mesma linha do arquivo, informa quantas vezes o primeiro termo do bigrama, “ciencia” ocorreu como sendo o termo mais à esquerda em todos os bigramas extraídos, ou seja, três vezes. Sendo duas vezes na própria linha dois e uma vez na linha cinco. O terceiro número informa quantas vezes o segundo termo do bigrama, “informação”, ocorreu como sendo o termo mais à direita em todos os bigramas extraídos, ou seja, quatro vezes. Sendo duas vezes na própria linha dois e duas vezes na linha três.

Para realizar o processo de geração dos arquivos com a extensão (.count) foi elaborado um *script Shell,* mostrado no Apêndice A, denomindo “GeraCount”. O *script* busca dentro do *corpus,* a partir da pasta base no sistema de arquivos, todos os arquivos com a extensão (.txt) existentes e, para cada um deles, gera um arquivo com a extensão (.count). Esse arquivo serve de base para a identificação das EM na próxima subetapa.

### 3.3.3.2 Converter os arquivos (.count) para cada uma das medidas NSP

As medidas de associação estatísticas são implementadas separadamente em pacotes Perl, esses arquivos possuem em seu nome a extensão (.pm). Ao rodar o statistic.pl

deve-se fornecer como parâmetro o nome da medida a ser usada e o nome do arquivo gerado pela saída do processamento do programa count.pl. O NSP disponibiliza treze diferentes medidas de associação que podem ser utilizadas para identificação de bigramas, conforme mostrado na Tabela 3.

TABELA 3 – Relação das medidas de associação estatística implementadas pelo NSP

| Nro | Nome da Medida de Associação Estatística | Parâmetro informado para statistic.pl |
|---|---|---|
| Mutual Information | | |
| 01 | Log-likelihood Ratio | Ll |
| 02 | Pointwise Mutual Information | Pmi |
| 03 | Mutual Information | Tmi |
| 04 | Poisson Stirling Measure | Ps |
| Fisher's Exact Test | | |
| 05 | Left Fisher | leftFisher |
| 06 | Right Fisher | rightFisher |
| 07 | Fisher Two-tailed Test | Twotailed |
| Chi Squared | | |
| 08 | Phi Coeficcient | Phi |
| 09 | Tscore | Tscore |
| 10 | Person's Chi Square Test | X2 |
| Dice | | |
| 11 | Coeficiente Dice | Dice |
| 12 | Jaccard Coeficient | Jaccard |
| Odds | | |
| 13 | Odds Ratio | Odds |

Fonte: Extraída da documentação do NSP.

O arquivo de saída gerado pelo statistic.pl tem a seguinte estrutura conforme mostrado na Figura 28.

| Número da linha | Conteúdo do arquivo |
|---|---|
| 1 | 5 |
| 2 | ciencia<>informacao<>**0.6667** 3 4 4 |
| 3 | sistemas<>informacao<>**0.5714** 2 2 4 |
| 4 | banco<>dados<>**0.5000** 3 3 5 |
| 5 | ciencia<>computacao<>**0.4451** 1 3 1 |
| 6 | processamento<>dados<>**0.4012** 2 1 5 |

FIGURA 28 – Conteúdo da saída produzido pelo statistic.pl.
Fonte: Gerada pelo NSP.

O arquivo gerado é ordenado de forma decrescente pela relevância do valor calculado, a partir de uma medida de associação informada como parâmetro para o

processamento. O valor calculado foi apresentado em destaque (negrito) somente para identificar a sua localização na estrutura do arquivo. Os números que aparecem em seguida ao valor calculado são os mesmos contidos no arquivo de entrada (.count), portanto, já foram descritos anteriormente. O programa statistic.pl recebe como parâmetro a opção (frequency *n*), a qual descarta os *n*-gramas que tenham uma frequência abaixo do valor definido por *n*. Neste experimento, o valor utilizado para *n* foi igual a quatro. Essa informação indica para o programa quantas serão as ocorrências consideradas para que um bigrama possa ser considerado como relevante.

Para realizar o processo de geração dos arquivos com a extensão específica para cada uma das treze medidas, elaborou-se um *script Shell* denomindo "GeraEM.exe" mostrado no Apêndice A. O *script* busca dentro do *corpus* todos os arquivos com a extensão (.count) existentes e para cada um deles gera treze arquivos com o mesmo nome do documento original com a extensão renomeada para o nome do parâmetro que identifica cada uma das treze medidas de associação estatística. As extensões dos arquivos gerados são as seguintes: ll, pmi, tmi, ps, leftfisher, rightfisher, twotailed, phi, tscore, x2, dice, jaccard, odds. Sendo assim, o programa statistic.pl foi executado uma vez para cada um dos treze parâmetros para cada um dos arquivos com a extensão (.count) correspondente a um documento do *corpus*. Por fim, esse processamento resultou na geração de 2.522 arquivos com o nome do documento e uma extensão correspondente a cada uma das técnicas utilizadas. Os arquivos criados foram armazenados nas mesmas pastas dos seus correspondentes arquivos de entrada. O total de arquivos gerados corresponde a 194 documentos vezes 13 diferentes medidas de associação estatística. Todos os arquivos gerados contêm as EM extraídas de forma ordenada por relevância.

O processamento do statistics.pl se baseia na montagem de uma tabela de contingência para cada bigrama extraído pelo count.pl. Portanto, ao se utilizar a linha dois (ciência<>informação<>3 4 5) da Figura 28, como referência para se montar a tabela de contigência, obtem-se o resultado conforme mostrado na Figura 29.

| Termos | `Informacao` | `~informacao` | Totais |
|---|---|---|---|
| `Ciencia` | $n_{11}$ | $n_{12}$ | $n_{1p}$ |
| `~ciencia` | $n_{21}$ | $n_{22}$ | $n_{2p}$ |
| | $n_{p1}$ | $n_{p2}$ | $n_{pp}$ |

FIGURA 29 – Matriz de contigência preenchida com os termos.
Fonte: Elaborada pelo autor.

Sendo que os valores informados correspondem a Figura 28:

- $n_{11}$ – Quantas vezes os termos "`ciencia`" e "`informacao`" formaram um bigrama igual a 3 mostrado na linha 2;

- $n_{21}$ – Quantas vezes o termo “`informacao`” aparece em outros bigramas em que o termo “`ciencia`” não aparece, 2 vezes, mostrado na linha 3;
- $n_{12}$ – Quantas vezes o termo “`ciencia`” aparece em outros bigramas em que o termo “`informacao`” não aparece, 1 vez, mostrado na linha 5;
- $n_{22}$ – Quantas vezes os termos “`ciencia`” e “`informacao`” não aparecem nos bigramas 2 vezes, mostrados nas linhas 4 e 6.

Feita essa apuração dos valores, a Figura 30 apresenta a tabela de contingência resultante gerada para analisar o relacionamento entre os termos do primeiro bigrama.

| Termos | *informacao* | ~informacao | Totais |
|---|---|---|---|
| *ciencia* | 3 | 1 | 4 |
| ~ciencia | 2 | 2 | 4 |
| | 5 | 3 | 8 |

FIGURA 30 – Matriz de contigência completa.
Fonte: Elaborada pelo autor.

A Figura 31 apresenta um esquema que representa os arquivos gerados para cada documento do *corpus* após processados os programas do pacote NSP.

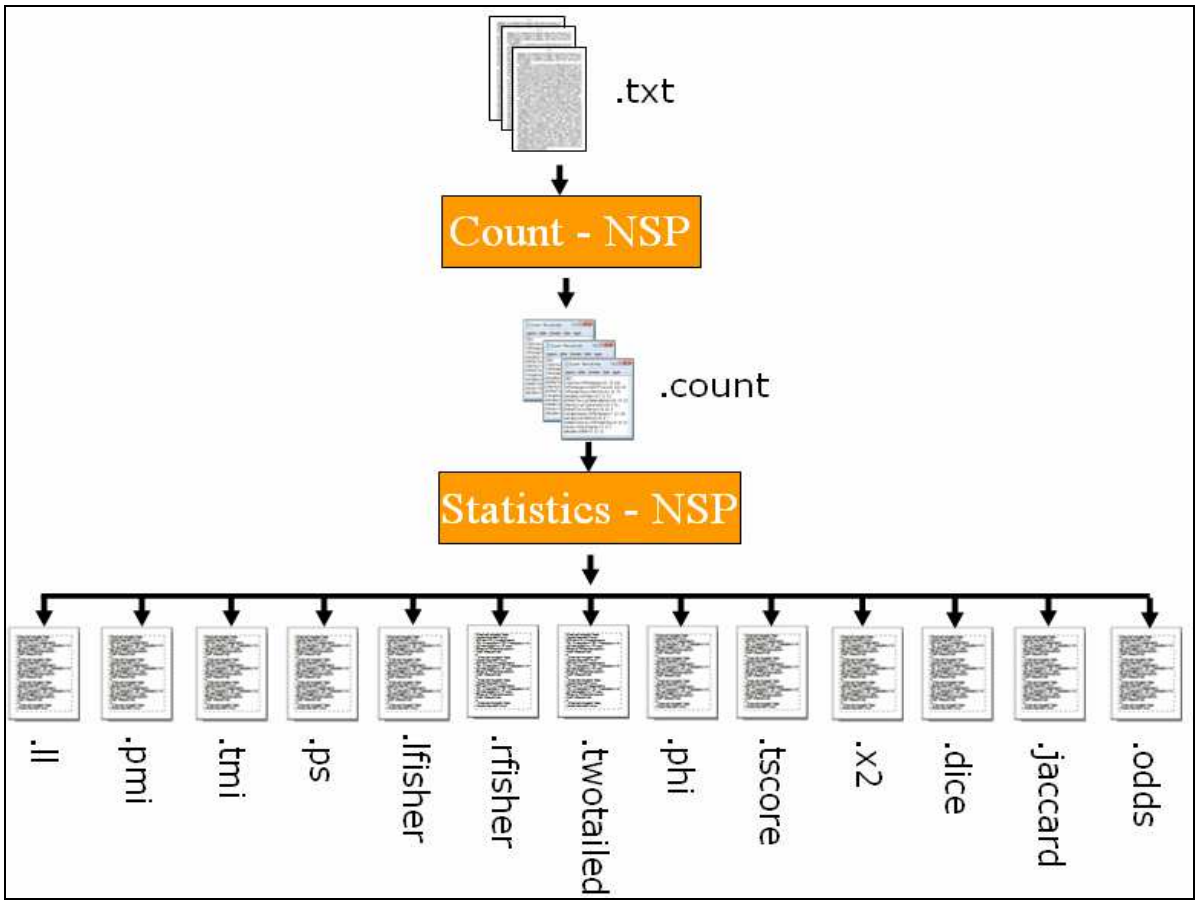



FIGURA 31 – Arquivos gerados pelo processamento do pacote NSP.
Fonte: Elaborada pelo autor.

## 3.4 Descrição da quarta fase

O objetivo desta fase é avaliar o resultado da extração das EM obtidas com o uso das técnicas estatísticas implementadas no pacote de *software* NSP e compará-las com as EM obtidas com o uso da técnica Heudet implementada pelo protótipo de *software* proposto pelo autor. A partir desses resultados definir a melhor estratégia para implementar um protótipo de um sistema de recuperação da informação que utilize para realizar a busca no *corpus* as EM extraídas de um documento de referência. Ou seja, a Busca Comparada. Adicionalmente, realizar experimentos exploratórios para validar as funcionalidades do protótipo de *software* proposto.

### 3.4.1 Validar a Busca Comparada

Para um melhor entendimento, essa fase foi dividida nas seguintes etapas descritas a seguir:

- Etapa 1 – Validar as EM obtidas pelos métodos estatísticos - NSP;
- Etapa 2 – Comparar as EM obtidas pelo NSP e Heudet;
- Etapa 3 – Analisar as funcionalidades da Busca Comparada.

A seguir apresenta-se uma explicação detalhada de cada uma das etapas.

#### 3.4.1.1 Validar as EM obtidas pelos métodos estatísticos - NSP

Diante de tantas alternativas de medidas de associação possibilitadas pelo *software* NSP, optou-se por realizar um ensaio a fim de validar o quão cada uma dessas medidas corrobora na identificação dos bigramas. O objetivo é fazer uma análise exploratória dos resultados produzidos pelas técnicas estatísticas a fim de avaliar qual é a melhor maneira de confrontá-las com os resultados obtidos pela extração determinística. Para realizar o experimento, foram tomados como base todos os artigos publicados no Enancib de 2010 contendo 194 artigos formados por entre 20 a 25 páginas, totalizando 682.537 termos normalizados, sendo 46.888 distintos. Esses documentos serviram como dados de entrada para o processamento do *software* Server a fim de gerar os arquivos (.txt) conforme descrito na primeira fase.

Após executadas a primeira e a segunda etapa da terceira fase, todos os arquivos com os bigramas gerados por cada uma das treze medidas de associação estatística estão disponíveis para serem avaliados. O objetivo é comparar o quão similares são as EM

extraídas por cada uma dessas técnicas. Nesse sentido, o pacote NSP disponibiliza um aplicativo denominado rank.pl que compara os resultados das EM extraídas de um mesmo documento através de duas técnicas de medida de associação diferentes. Ele utiliza o Coeficiente de Correlação de Spearman, que serve para medir quão similares são os arquivos de resposta gerados pelo statistic.pl. Ou seja, dados dois arquivos com a relação das EM extraídas e suas frequências observadas no *corpus*, a correlação é calculada pela expressão mostrada em (3.8).

$$r = 1 - \frac{6\sum_{i=1}^{i=n} D_i^2}{n(n^2 - 1)} \tag{3.8}$$

Onde, *n* é o número total de *n*-gramas distintos do *corpus.* $D_i$ é a diferença entre as medidas das duas listas na posição *i* e *r* é o valor da correlação. O valor calculado de *r* varia de -1 até 1. Sendo que, os valores próximos de -1 indicam que os valores apurados são opostos, próximos de zero, que eles não se relacionam e próximos de 1, que os valores medidos são de mesma ordem.

No intuito de comparar os arquivos, foi elaborado e executado em *script shell* denominado “Compara”, mostrado no apêndice A, que processa um produto cartesiano entre cada uma das treze técnicas. Dessa forma, para cada documento, os treze arquivos são comparados par a par gerando 169 comparações para cada um dos 194 documentos totalizando 32.786 processamentos de comparação. Sendo que o resultado da comparação par a par é direcionado para um arquivo com o mesmo nome do arquivo original, mas com a extensão modificada para (.rank). Um exemplo desse arquivo pode ser visto na Figura 32.

| Número da linha | Conteúdo do arquivo |
|---|---|
| 1 | `Rank correlation coefficient = -0.0134` |
| 2 | `Rank correlation coefficient = -0.0134` |
| 3 | `Rank correlation coefficient = 1.0000` |
| ... | ... |
| 169 | `Rank correlation coefficient = 0.0030` |

FIGURA 32 – Estrutura do arquivo (.rank).
Fonte: Gerado pelo *software* NSP.

Todos os 194 arquivos (.rank) foram gerados dentro das mesmas pastas onde foram lidos como arquivos de entrada, e estão disponibilizados no *corpus* com o mesmo nome do arquivo original, mas com a extensão renomeada para .rank. Cabe, então, analisar os resultados. Nesse sentido, foi elaborado um programa em C++ denominado “Rank.exe” capaz de processar todos esses arquivos, pegando como informação apenas o valor atribuído ao *rank* a cada combinação. O objetivo é de criar na memória uma matriz de correlações em três dimensões. Sendo os eixos *x* e *y* um plano para correlacionar as treze

técnicas e as ocorrências do eixo z, correspondendo a cada um dos documentos. O resultado desse processamento gerou uma estrutura de dados alocada na memória volátil do computador, possibilitando comparar os resultados das diversas técnicas. Sendo que cada plano da estrutura é constituído por uma matriz simétrica, na qual a diagonal principal tem sempre o valor igual a um. Isso se dá em função dessas posições no plano correlacionarem uma mesma técnica. Após a matriz estar gerada, realizou-se um processamento de totalização em que cada correlação par a par foi acumulada no eixo *z* na posição 195, em um novo plano cartesiano. Desse modo, nesse plano foi armazenado o somatório das correlações da combinação par a par de cada uma das técnicas de todos os 194 documentos. A Figura 33 apresenta um esboço dessa estrutura.

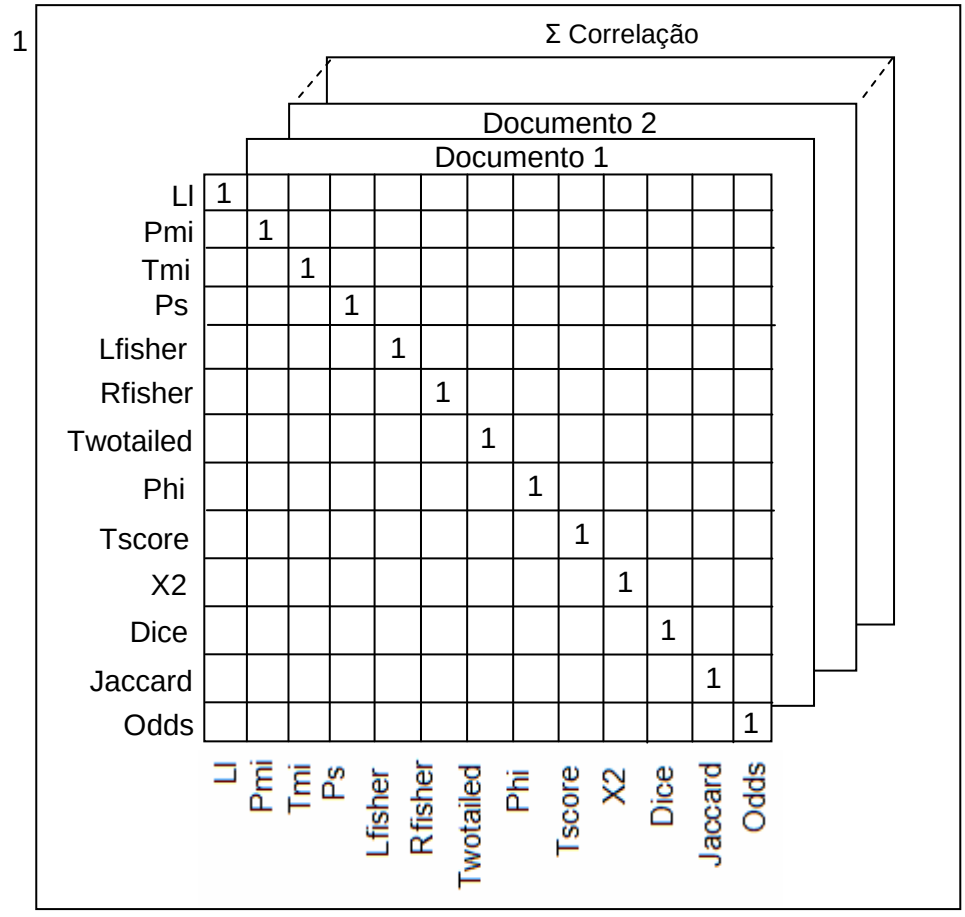



FIGURA 33 – Estrutura tridimensional para alocar os valores das comparações.
Fonte: Elaborada pelo autor.

Por ser a matriz resultante simétrica consideraram-se apenas os valores acima da diagonal principal, correspondentes combinação das treze medidas duas a duas ($C_{13,2}$) totalizando 78 possibilidades. Desse modo, esses valores foram ordenados a fim de mostrar o quão essas medidas de associação se correlacionam. A Tabela 4 apresenta o valor calculado de cada uma das 78 combinações ordenadas, da mais para a menos similar.

Como foram processados 194 documentos, e, considerando que o valor da correlação de Spearman para cada comparação entre duas técnicas pode variar de -1 até 1, ao se realizar o somatório de todos os documentos, assume-se que o valor pode variar de -194 a 194.

TABELA 4 – Correlação par a par das medidas de associação estatísticas.

| | | | | | |
|---|---|---|---|---|---|
| Dice | jaccard | 194.00 | twotailed | lfisher | 115.57 |
| phi | x2 | 194.00 | lfisher | rfisher | 115.57 |
| twotailed | rfisher | 194.00 | pmi | ps | 115.13 |
| ll | tmi | 193.92 | jaccard | tscore | 98.04 |
| ll | ps | 192.64 | dice | tscore | 98.03 |
| ps | tmi | 192.64 | phi | tscore | 97.39 |
| jaccard | phi | 189.78 | tscore | x2 | 97.36 |
| dice | phi | 189.78 | odds | tscore | 76.93 |
| jaccard | x2 | 189.77 | pmi | tscore | 42.45 |
| dice | x2 | 189.77 | tmi | rfisher | -73.29 |
| odds | phi | 188.37 | tmi | twotailed | -73.29 |
| odds | x2 | 188.37 | tscore | twotailed | -73.51 |
| odds | pmi | 182.41 | tscore | rfisher | -73.51 |
| jaccard | odds | 180.78 | jaccard | twotailed | -74.31 |
| dice | odds | 180.78 | jaccard | rfisher | -74.31 |
| phi | pmi | 177.24 | dice | twotailed | -74.32 |
| pmi | x2 | 177.23 | dice | rfisher | -74.32 |
| jaccard | pmi | 172.38 | pmi | twotailed | -75.22 |
| dice | pmi | 172.38 | pmi | rfisher | -75.22 |
| ps | tscore | 163.52 | ps | twotailed | -76.00 |
| phi | tmi | 162.64 | ps | rfisher | -76.00 |
| tmi | x2 | 162.63 | odds | twotailed | -76.39 |
| ll | phi | 162.54 | odds | rfisher | -76.39 |
| ll | x2 | 162.53 | ll | rfisher | -76.46 |
| tmi | tscore | 161.43 | ll | twotailed | -76.46 |
| ll | tscore | 161.29 | phi | rfisher | -77.00 |
| phi | ps | 159.70 | phi | twotailed | -77.00 |
| ps | x2 | 159.69 | rfisher | x2 | -77.18 |
| jaccard | tmi | 158.24 | twotailed | x2 | -77.18 |
| dice | tmi | 158.24 | tmi | lfisher | -120.99 |
| jaccard | ll | 158.12 | ps | lfisher | -124.35 |
| dice | ll | 158.12 | pmi | lfisher | -124.59 |
| jaccard | ps | 157.53 | jaccard | lfisher | -124.86 |
| dice | ps | 157.52 | dice | lfisher | -124.86 |
| odds | tmi | 147.30 | ll | lfisher | -124.89 |
| ll | odds | 147.09 | phi | lfisher | -125.28 |
| odds | ps | 143.24 | lfisher | x2 | -125.44 |
| pmi | tmi | 118.82 | odds | lfisher | -126.11 |
| ll | pmi | 118.43 | tscore | lfisher | -129.75 |

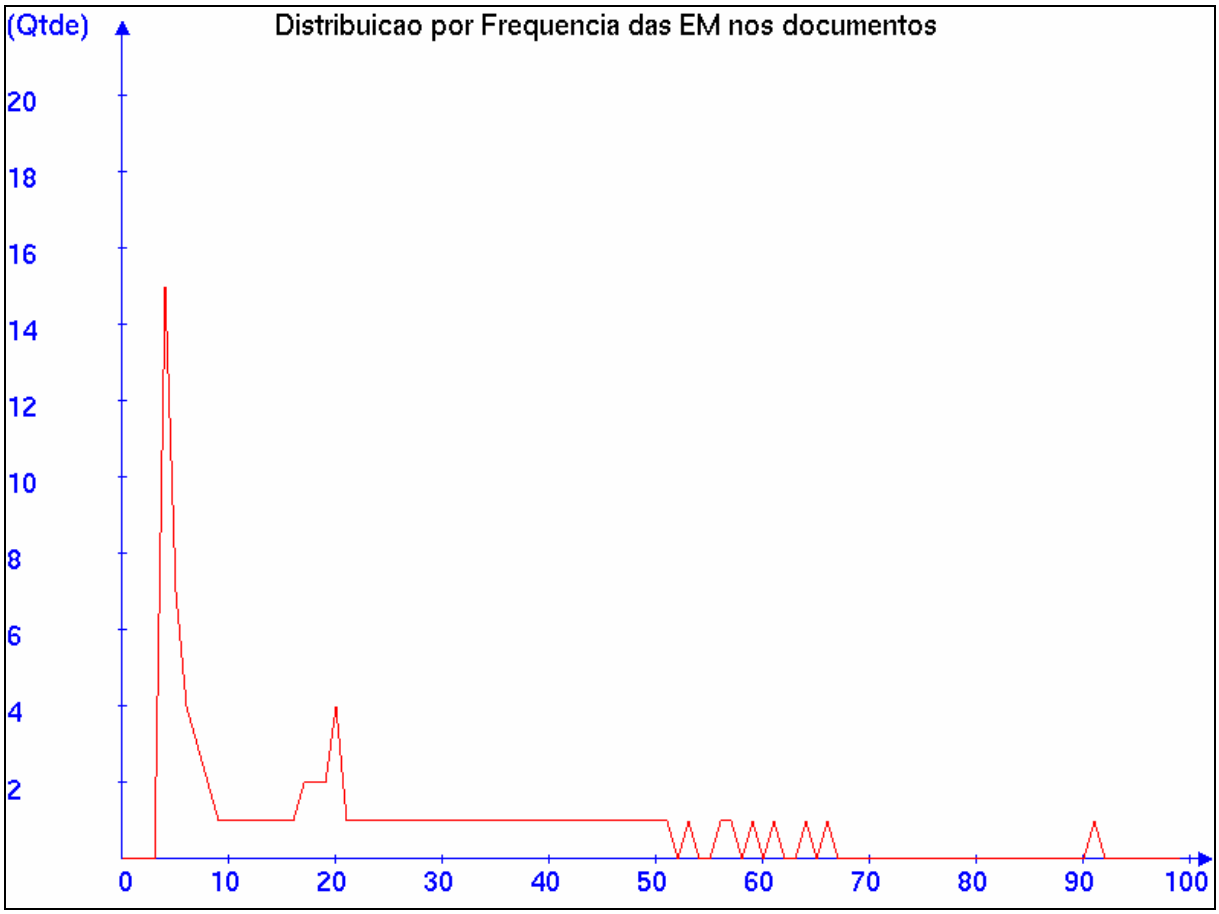
Fonte: Elaborada pelo autor.

Conforme pode ser observado, as técnicas dice e jaccard, phi e x2, twotailed e rfisher mostradas nas três primeiras linhas da tabela geraram resultados totalmente similares. Isso significa que esses pares de técnicas extraíram o mesmo conjunto de EM e com a mesma ordem de relevância. A partir da quarta linha da tabela, começaram a surgir diferenças, sendo que, a partir da quadragésima oitava linha, o valor se tornou negativo, ou seja, os resultados dos conteúdos dos arquivos, a partir desse ponto, são distintos em relação às EM extraídas ou em relação à ordenação por relevância. Nesse caso, pode ocorrer que, apesar

de o conjunto de EM extraídas pelas diversas técnicas ser bastante similar, a ordem que cada bigrama aparece nos arquivos seja diferente em função da relevância calculada por cada técnica.

Para entender como se dá a distribuição da frequência de co-ocorrência dos bigramas encontrados no *corpus*, uma outra análise foi proposta. Para representar essa distribuição de forma gráfica elaborou-se um programa em C++ denominado “Gráfico.exe”. O objetivo desse programa é gerar um gráfico que mostra no eixo X o percentual de bigramas extraídos e no eixo Y os relaciona com co-ocorrência das quantidades. Ou seja, mostrar qual é a quantidade de bigramas gerados distribuídos de forma percentual dentre todo o conjunto de bigramas. Ao observar o gráfico, constata-se que um pouco menos de 10% dos bigramas ocorrem em uma quantidade de co-ocorrências maior, com cerca de até 16 casos, enquanto a maioria ocorre em uma quantidade perto do ponto de corte definido na pesquisa, que é de quatro casos. O Gráfico 1 apresenta a distribuição de frequência obtida pelo processamento.

GRÁFICO 1 – Distribuição de frequência dos bigramas extraídos do *Corpus*.




Fonte: Elaborado pelo autor.

Após realizada essa análise exploratória, percebe-se que as diversas técnicas estatísticas produzem resultados diferentes. Desse modo, torna-se necessário identificar onde essas diferenças ocorrem. Portanto, elaborou-se mais um experimento exploratório de modo que fosse possível avaliar o conteúdo dos arquivos gerados e compará-los incluindo também os resultados obtidos pela técnica determinística. Esse é o tema apresentado na próxima seção.

### 3.4.1.2 Comparar NSP *versus* Heudet

Nesta etapa, os resultados das EM extraídas dos documentos do *corpus* através dos processamentos determinísticos e estatísticos resultantes da segunda e terceira fases são comparados em termos quantitativos e qualitativos. Além disso, serão avaliadas questões tais como: o tempo de resposta computacional e facilidade de implementação das técnicas utilizadas pelo pacote de *software* NSP, comparando-a com a técnica Heudet proposta pelo autor desta tese para realização do processo de extração de EM.

Conforme afirma Ramish (2009, p. 67) como a distribuição de probabilidades do vocabulário da língua é zipfiana[32], um fenômeno conhecido como “cauda longa”. Isso indica que as palavras que estão na cauda da distribuição por frequência são as palavras mais raras e portanto, melhor discriminam um texto dentro de um *corpus*. Consequentemente, os dados de frequência tornam-se esparsos e as medidas de associação apresentam pouca confiabilidade ao serem extraídas a partir de métodos probabilísticos. Dessa forma, espera-se que se tenham bons resultados ao utilizar a técnica exaustiva determinística proposta.

A fim de operacionalizar o tratamento das informações produzidas, foi elaborado um componente de *software* desenvolvido em C++ denominado “AtuMySql.exe”. O objetivo desse programa é de ler todos os arquivos com as EM gerados pelas 14 técnicas durante o processamento da segunda e terceira fases e inserir esses dados em tabelas de um banco de dados MySql. Desse modo, com os dados organizados em tabelas estruturadas, facilita analisá-los, utilizando a Strutured Query Language (SQL) que implementa os operadores para comparação de conjuntos. Ou seja, a SQL permite comparar o conteúdo das respostas obtidas por cada uma das técnicas. Os dados armazenados em tabela foram: o código da técnica utilizada para a extração da EM, o código do documento, o número sequencial da EM, o valor do coeficiente calculado, o primeiro e o segundo termos do bigrama.

Desse modo, ao ser executado, o programa AtuMySql carregou todos os arquivos com as EM extraídas em uma tabela MySql denominada “docmetrica”. A Figura 34 apresenta um esboço de apenas uma pequena parte dessa tabela, para se dar uma ideia de sua estrutura e do conteúdo armazenado.

[32] A Lei de Zipf é uma lei empírica baseada na distribuição da frequência da ocorrência das palavras em um *corpus* de texto. Ela demonstra que o resultando do produto entre a frequência de ocorrência da palavra pela posição em que a palavra se encontra na lista ordenada por distribuição de frequência, se mantêm constante, mas somente para as palavras de alta ocorrência.

| Metrica | Documento | Ordem de Relevância | termo1 | termos2 |
|---|---|---|---|---|
| dice | 8 | 1 | rio | janeiro |
| pmi | 8 | 1 | relatorio | parcial |
| odds | 8 | 1 | rio | janeiro |
| ps | 8 | 1 | ciencia | informacao |
| lfisher | 8 | 1 | ciencia | informacao |
| rfisher | 8 | 1 | informacao | comunicacao |
| jaccard | 8 | 1 | rio | janeiro |
| tscore | 8 | 1 | ciencia | informacao |
| em | 8 | 1 | ciencia | informacao |
| x2 | 8 | 1 | rio | janeiro |
| phi | 8 | 1 | rio | janeiro |

FIGURA 34 – EM extraídas pelas treze técnicas estatísticas
Fonte: Gerada pelo *software* PhpMyAdmin.

Ao observar a Figura 34, percebe-se que, no exemplo foram retornados quatro bigramas distintos como sendo os mais relevantes (rio janeiro, relatório parcial, ciencia informacao, informacao comunicacao). Todas as EM foram extraídas de um mesmo documento, o número oito. Todas as EM mostradas possuem a mesma relevância igual a um. Isso permite concluir que técnicas distintas podem identificar EM em uma ordem de relevância diferente, embora esse não seja o caso no exemplo apresentado. Cabe ainda ressaltar que os conteúdos são mostrados sem acentos e em minúsculas por já estarem normalizados.

Portanto, ainda falta identificar o quão diferentes são as EM extraídas pelas treze técnicas estatísticas e compará-las com as extraídas pela técnica Heudet proposta. Nesse sentido, algumas consultas foram submetidas ao SGBD a fim de produzir a Tabela 5 que apresenta os valores quantitativos de EM obtidas por cada uma das técnicas e uma linha com esses valores totalizados.

TABELA 5 – Resultado da extração de EM.

| Técnica | Qtde EM | EM a descartar | Qtde EM Normalizadas |
|---|---|---|---|
| **Heudet** | 7.734 | 0 | 7.734 |
| **Dice** | 7.832 | 86 | 7.746 |
| **Jaccard** | 7.832 | 86 | 7.746 |
| **Lfisher** | 7.832 | 86 | 7.746 |
| **Ll** | 7.832 | 86 | 7.746 |
| **Odds** | 7.832 | 86 | 7.746 |
| **Phi** | 7.832 | 86 | 7.746 |
| **Pmi** | 7.832 | 86 | 7.746 |
| **Os** | 7.832 | 86 | 7.746 |
| **Rfisher** | 7.832 | 86 | 7.746 |
| **Tmi** | 7.832 | 86 | 7.746 |
| **Tscore** | 7.832 | 86 | 7.746 |
| **Twotailed** | 7.832 | 86 | 7.746 |
| **X2** | 7.832 | 86 | 7.746 |
| **Totais** | 109.550 | 1.118 | 108.432 |

Fonte: Elaborada pelo autor.

A coluna "**Qtde EM**" foi produzida como sendo o resultado do comando SQL mostrado em (3.9) que foi executado para mostrar quantas foram as EM geradas em cada uma das técnicas utilizadas.

```
Select cod_metrica, count(*)
  from docmetrica
 group by 1
 order by 1
```
(3.9)

Para compatibilizar o processo de extração de EM geradas pelas técnicas estatísticas através do NSP com a técnica Heudet, foram descartados todos os bigramas nos quais pelo menos um dos termos contivesse apenas um caracter. A coluna "**Qtde a Descartar"** apresentada, quantifica esses casos. Ela é produzida como resultado do comando SQL (3.10). Já a coluna "**Qtde EM Normalizada**" apresenta o resultado normalizado, ou seja, o total extraído por cada uma das técnicas subtraindo os bigramas considerados como irrelevantes. No total, 1118 bigramas foram descartados da tabela docmetrica.

```
Select cod_metrica, count(*)
  from docmetrica
 where length(txt_termo1) = 1
    or length(txt_termo2) = 1
group by 1
order by 1
```
(3.10)

Após realizado o processo de normalização dos dados, foram executados inicialmente, dois comandos. O primeiro mostrado em (3.11) para verificar quantos são os bigramas extraídos por todas as técnicas. O total encontrado foi de 5.841 bigramas distintos.

```
select count(distinct(concat(txt_termo1,txt_termo2)))
  from docmetrica
```
(3.11)

O segundo comando verifica quanto foi o total de EM extraídas considerando os casos repetidos, quando pertencerem a documentos diferentes. Foi encontrado um total de 7.844 EM, extraídas por todas as técnicas. Esse valor pode ser calculado pelo SQL mostrado em (3.12).

```
select count(distinct(concat(nro_doc,txt_termo1,txt_termo2)))
  from docmetrica
```
(3.12)

Para separar todas as EM comuns que foram obtidas pelas quatorze técnicas, criou-se uma tabela denominada “emcomum” que foi carregada pelo comando SQL mostrado em (3.13).

```
Insert into emcomum
select nro_doc, txt_termo1, txt_termo2
  from docmetrica
 group by 1,2,3
having count(*) = 14
```
(3.13)

Ao executar esse comando, foram inseridas 7.636. Ou seja, dentre todas as EM distintas obtidas, 7.636 correspondentes a 97,35% do total, foram encontradas por todas as quatorze técnicas utilizadas. A diferença apurada entre as EM extraídas pelas técnicas estatísticas e determinística foi de 208 bigramas, correspondentes a 2,65% do total. E é através do aprofundamento da análise dessas EM distintas que se esperam obter meios de comparar as vantagens e desvantagens obtidas pelo uso das técnicas estatísticas com a determinística proposta nesta tese. Sendo assim, criou-se uma nova tabela “emdistintas”, obtida pela diferença do conjunto das EM distintas pelas EM comuns. O comando SQL utilizado é mostrado em (3.14).

```
insert into emdistintas
select A.cod_metrica, A.nro_doc, A.txt_termo1, A.txt_termo2
  from docmetrica A
 where not exists (select * from emcomum B
                   where B.nro_doc = A.nro_doc
                     and B.txt_termo1 = A.txt_termo1
                     and B.txt_termo2 = A.txt_termo2)
```
(3.14)

Para entender melhor esse conjunto de EM distintas o comando SQL mostrado em (3.15) foi executado e o resultado apresentado na Figura 35.

```
Select cod_metrica, count( * )
  from emdistintas
 group by 1
 order by 1
```

(3.15)

| cod_metrica | count( * ) |
|---|---|
| dice | 110 |
| heudet | 98 |
| jaccard | 110 |
| lfisher | 110 |
| ll | 110 |
| odds | 110 |
| phi | 110 |
| pmi | 110 |
| ps | 110 |
| rfisher | 110 |
| tmi | 110 |
| tscore | 110 |
| twotailed | 110 |
| x2 | 110 |

`

FIGURA 35 – Totais de EM distintas
Fonte: Gerada pelo *software* PhpMyAdmin.

Como pode ser observado na Figura 35, 98 das EM foram extraídas exclusivamente pela técnica Heudet e 110 foram extraídas exclusivamente pelas técnicas estatísticas através do NSP. A Figura 36 mostra uma representação desses conjuntos.

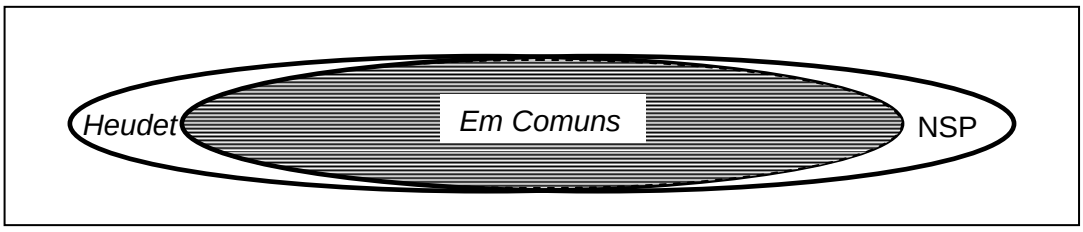



FIGURA 36 – EM extraídas comparando a técnica determinística com as estatísticas
Fonte: Elaborada pelo autor.

Identificados os quantitativos das EM distintas, cabe agora avaliar qualitativamente as características desses bigramas. Ou seja, interessa avaliar o que levou as técnicas estatísticas a considerar certos bigramas como sendo relevantes, e por que a técnica Heudet não os considerou e vice-versa. Portanto, verificou-se através de uma inspeção documento a documento, cada um dos 208 bigramas distintos visando identificar e agrupá-

los por características comuns. Dentre os 110 bigramas encontrados exclusivamente pelas técnicas estatísticas, foram identificados quatro motivos principais:

- 83 casos, correspondentes a 75,5% não deveriam ser considerados como bigramas. Esse erro produzido pelos algoritmos estatísticos se explica pelo fato de eles, diferentemente da técnica proposta, não levarem em consideração as características estruturais do documento. Esses algoritmos consideram o texto como sendo um saco de palavras, do inglês *bag of words.* Dessa forma, eles não identificam a divisão do texto em sentenças e extraem EM, mesmo nos pontos limítrofes de sentenças adjacentes. Portanto, agrupam palavras que pertencem a sentenças distintas por isso não podem ser considerados como tendo um significado agrupado.
- 18 casos, correspondentes a 16,4% também não deveriam ser considerados como bigramas. Essa característica dos algoritmos estatísticos do NSP ocorre por considerar como sendo *tokens* caracteres tais como: “%” (percentual); “&” (e comercial); e a “,” (vírgula) mesmo quando utilizada como separador dentre valores numéricos. Sendo assim, eles separam um termo em dois, que, na prática, possuem um melhor sentido quando mantidos juntos, como por exemplo: CT&I é tratado como sendo os termos “CT” e “I”; um número 45,99 é tratado como sendo os termos “45” e “99”.
- 4 casos, correspondentes a 3,6% também deveriam ser considerados como bigramas. Entretanto, a forma como os termos do bigrama são constituídos diverge devido a características próprias de cada um dos algoritmos. No caso do NSP os caracteres “<” (sinal de menor), “>” (sinal de maior), “@” (arroba), “#” (número) são considerados como caracteres de separação de palavras, já no caso do Heudet esses mesmos caracteres são considerados como parte integrante da palavra. Ou seja, isso é apenas uma diferença de abordagem no processo de tokenização que pode ser facilmente adaptada no programa fonte. Entretanto, optou-se por não considerar esses caracteres como *token*s de quebra de palavras por considerar que agregados à palavra, eles expressam um melhor significação do termo. Como por exemplo: html e <html>, nitidamente, o segundo termo se mostra como uma tag, em vez de apenas uma palavra solta. O mesmo ocorre com “@” que pode ser parte de um endereço de e-mail.
- 5 casos, correspondentes a 4,5%, o uso do ponto final como delimitador da separação do texto em sentenças, pode ser considerado como causador de um falso positivo. Ou seja, o algoritmo Heudet provocou a quebra do texto em uma

sentença de forma errônea. Por exemplo, em termos abreviados terminados por vogal e seguido de letra maiúscula, exemplo “Dra. Railly”, pois a heurística implementada em Heudet considera que ponto final antecedido por letra minúscula e precedido por letra maiúscula indica uma divisão de sentenças.

Portanto, observa-se, através dessa análise qualitativa dos bigramas extraídos que 91,9% dos bigramas gerados exclusivamente pelo NSP adicionam imprecisão no processo de identificação de EM. Por outro lado, ao se analisarem os 98 bigramas encontrados exclusivamente pela técnica Heudet, foi identificado apenas um motivo para essa diferença, conforme mostrado a seguir:

- Devido à maneira distinta da forma dos algoritmos implementados pelo NSP e pelo Heudet em tratarem os caracteres “%” (Percentual e “&” (e comercial), os termos das EM gerados foram consideradas diferentes. Afinal os termos de Heudet incluem esses caracteres como parte do termo. Ou seja, conforme apresentado anteriormente por parte das EM obtidas exclusivas do processo estatístico, esses casos são apenas uma diferente forma de abordagem do processo de tokenização no qual se considera que o modo como a abordagem estatística lida com a questão, insere imprecisão por considerar separados termos que deveriam estar juntos.

Dessa forma, conclui-se que o algoritmo Heudet apresenta vantagens em relação ao uso das técnicas estatísticas, isso se dá pelo fato de ele levar em consideração a estrutura do documento. Esse ganho foi medido empiricamente neste experimento, no qual ao se utilizar a abordagem Heudet, 101 EM são descartadas, isso representa um ganho de precisão de 1,29%. Em relação ao desempenho, verificou-se que o tempo de extração de todo o *corpus* pelo algoritmo Heudet consumiu 182 segundos, rodando no ambiente Windows Vista, enquanto pelo NSP consumiu 235 segundos rodando, no ambiente Ubuntu versão 10.4. Todos os processamentos foram executados em um *notebook* core ™ 2 DUO Cpu T6400 2.0 Ghz.

No trabalho publicado anteriormente por Silva e Souza (2012), esse mesmo experimento foi realizado, entretanto, utilizando o parâmetro *n,* da frequência, com valor igual a três. Ou seja, para um bigrama ser identificado em um documento, é necessário que ele co-ocorra pelo menos três vezes. Portanto, o volume de bigramas extraídos foi maior do que no experimento replicado nesta tese, a qual utilizou o valor de *n* igual a quatro.

Esses resultados são apresentados na Tabela 6. A coluna (A) mostra o valor total das EM extraídas pelas quatorze diferentes técnicas. A coluna (B) mostra a quantidade de EM extraídas em que um ou mais de seus termos têm apenas um caracter, em média 155

casos. Esses casos foram discartados e o resultado apurado é mostrado na coluna (C). A coluna (D) mostra quantidade de EM comuns encontradas comparando Heudet com cada uma das demais técnicas obtidas pelo pacote NSP. Finalmente, a coluna (E) apresenta a diferença entre os totais de EM apurados pelas técnicas NSP e pela Heudet. Em média, essa diferença foi de 565 casos, relacionados a duas situações: na primeira, 223 casos correspondentes às características distintas utilizadas pelo processo de tokenização entre as técnicas do pacote NSP e a Heudet; na segunda, 343 casos correspondentes principalmente a EM extraídas pelo NSP em pontos limítrofes de sentenças adjacentes.

TABELA 6 – Resultados da extração das EM.

| Técnica | **(A) Quantidade de EM extraída** | **(B) Ruído** | **(C) A – B** | **(D) Comuns Com heudet** | **(E) C – D** |
|---|---|---|---|---|---|
| **Odds** | 15054 | 155 | 14899 | 14324 | 575 |
| **X2** | 15055 | 155 | 14900 | 14329 | 571 |
| **Os** | 15062 | 154 | 14908 | 14344 | 564 |
| **Jaccard** | 15063 | 155 | 14908 | 14338 | 570 |
| **Ll** | 15063 | 154 | 14909 | 14345 | 564 |
| **Tscore** | 15063 | 153 | 14910 | 14345 | 565 |
| **Phi** | 15064 | 155 | 14909 | 14338 | 571 |
| **Dice** | 15064 | 155 | 14909 | 14339 | 570 |
| **Twotailed** | 15065 | 158 | 14907 | 14364 | 543 |
| **Lfisher** | 15065 | 158 | 14907 | 14365 | 542 |
| **Pmi** | 15067 | 159 | 14908 | 14333 | 575 |
| **Tmi** | 15068 | 154 | 14914 | 14349 | 565 |
| **Rfisher** | 15068 | 154 | 14914 | 14351 | 563 |
| **Média** | 15063 | 155 | 14908 | 14343 | 565 |
| **Heudet** | 14755 | - | - | - | - |

Fonte: Silva e Souza (2012).

Portanto, 14.343 correspondentes a 96,21% das EM extraídas são idênticas, independente da técnica utilizada. Os 223 casos, correspondents a 1.5%, são EM extraídas exclusivamente pela técnica Heudet, as quais representam um ganho na precisão. Os 342 casos restantes, correspondents a 2,29% são considerados ruídos, portanto, inserem imprecisão no resultado. Comparando os dois experimentos, verificou-se que quando o volume de EM extraídas foi maior, com o parâmetro da frequência igual a três, a técnica

Heudet apresentou melhor resultado na eliminação dos ruídos do processo de aquisição de EM.

#### 3.4.1.3 Analisar as funcionalidades da Busca Comparada

Tendo em vista as vantagens obtidas com o método proposto, optou-se por desenvolver o protótipo de Busca Comparada baseado apenas no algoritmo Heudet. Para avaliar o funcionamento da ferramenta proposta de Busca Comparada, foram executados cinco experimentos descritos no próximo capítulo.

# 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Como foi demonstrado no capítulo anterior, através da execução das fases da metodologia proposta, a qual buscava avaliar o resultado do método Heudet comparando-o com as técnicas estatísticas implementadas pelo pacote NSP, constatou-se que o método proposto apresenta um melhor resultado em termos de desempenho e precisão. Desse modo, a ferramenta de Busca Comparada foi elaborada usando exclusivamente o método Heudet proposto. Portanto, o objetivo, após a definição da técnica e a construção da ferramenta de Busca Comparada, passa a ser a validação de uso através de experimentos controlados. Esses experimentos são descritos a seguir.

## 4.1 Primeiro experimento exploratório

Antes de proceder a avaliação dos resultados da busca propriamente dita, faz-se necessário avaliar as características do conjunto de termos resultantes do processo de representação do conteúdo textual.

Para possibilitar uma melhor avaliação desses termos, os quais são indexados para busca, elaborou-se um programa em C++, denominado atuLexico.exe. O objetivo é transferir o conteúdo indexado na memória volátil do computador para um banco de dados MySql. Desse modo, utilizando a linguagem SQL fica mais fácil avaliar suas características. Portanto, foi criada uma tabela, denominada “Léxico”, contendo os seguintes atributos: o termo do léxico, a quantidade de documentos que o termo ocorre e a frequência total do termo no léxico.

A primeira avaliação a ser feita é para verificar se existem termos que podem ser incluídos na lista de *stop words.* Desse modo, o comando sql (4.1) foi executado e o seu resultado inspecionado manualmente. O objetivo é encontrar os termos cuja frequência de ocorrência geral corresponde a pelo menos 10% da máxima frequência ocorrida e que também tenha uma ocorrência em acima de 90% do total de documentos do *corpus.* A Figura 37 apresenta o resultado obtido.

```
select * from lexico
 where qtd_freq > (select max(qtd_freq) from lexico)*0.1
  and qtd_doc > 194 *0.9
```
(4.1)

| Txt_termo | Qtd_doc | Qtd_Freq |
|---|---|---|
| in | 186 | 1510 |
| and | 184 | 1923 |
| sao | 193 | 3767 |
| cada | 188 | 1236 |
| pode | 184 | 1442 |
| alem | 187 | 1257 |
| ainda | 186 | 1318 |
| forma | 190 | 1884 |
| sobre | 190 | 2781 |
| assim | 188 | 1856 |
| estudo | 181 | 1287 |
| partir | 185 | 1410 |
| relacao | 186 | 1412 |
| segundo | 183 | 1389 |
| trabalho | 184 | 2082 |
| pesquisa | 189 | 3277 |
| processo | 186 | 1996 |
| informacao | 194 | 11940 |
| informacoes | 181 | 2088 |
| universidade | 185 | 1779 |

FIGURA 37 – Termos com alta frequência que ocorrem em muitos documentos.
Fonte: Elaborada pelo autor.

Tomando como base esse resultando optou-se por incluir na lista de *stop words* os seguintes termos: in, and, sao, cada, pode, alem, ainda, sobre, assim. A lista de *stop words* passou a conter, após essas novas inclusões, 211 termos. A lista completa é mostrada no Apêndice B.

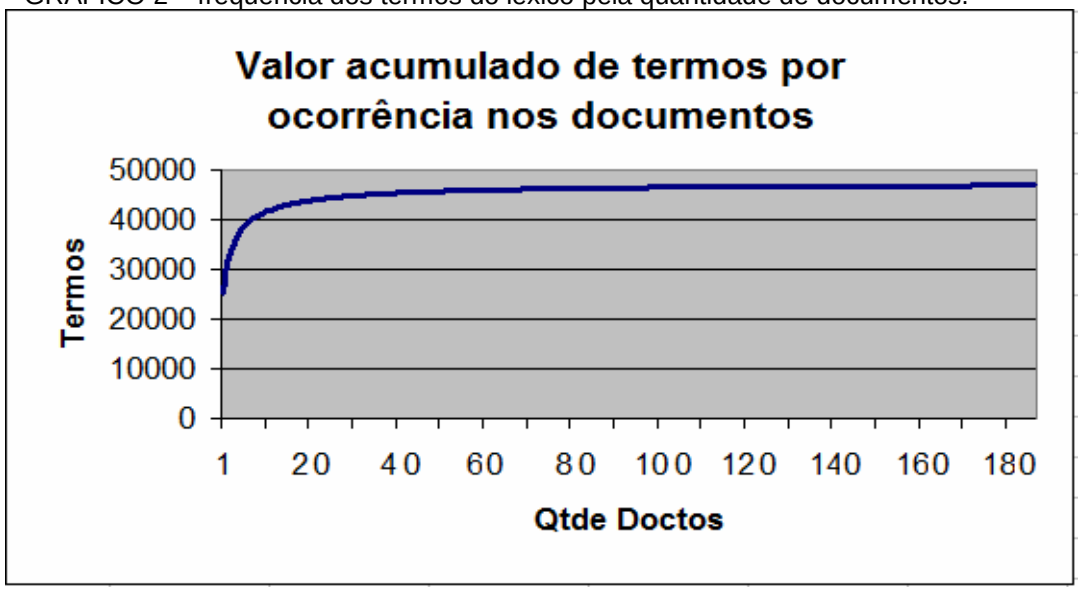
Após incluídas as *stop words* na lista, o programa Server foi executado. O processamento desse programa gera uma contabilização dos dados processados. O resultado mostrou que o tamanho total do texto após todos os documentos do *corpus* terem sido convertidos do formato pdf para texto puro é igual a 9.156.994 *bytes*. Realizadas as etapas de filtragem e retirada de *stop words,* a quantidade de termos identificados pelo *parser* de palavras foi igual a 662.194. Esse conjunto de termos a serem indexados tem o tamanho de 5.186.649 *bytes*. Ou seja, a ferramenta utilizada no processo de indexação propiciou um taxa de compressão de 43,36% de redução no tamanho do conteúdo textual a ser indexado. Mas, como o processo de indexação armazena apenas cada termo uma única vez, a quantidade de termos distintos encontrada foi igual a 46.878 e totalizando a 408.556 bytes. Em outras palavras, esse é o tamanho do léxico indexado.

Apresentados os números globais do processo de indexação, cabe agora fazer um estudo do conteúdo do léxico gerado. Do total de 46.878 termos distintos, 21.266 correspondendo a 45,54% são termos que ocorrem uma única vez. Esses seriam, portanto, os melhores descritores de busca ainda que, muitos desses termos sejam valores numéricos, ruídos ou até mesmo palavras escritas com erro de ortografia.

Por outro lado, dos 662.194 termos existentes no *corpus*, somente os termos “informacao”, “ciencia”, “conhecimento” e “pesquisa”, os quais aparecem 11.940, 4.226, 3.435 e 3.277, correspondem a 1,8%, 0,638%, 0,52% e 0,49% respectivamente. Ou seja, esses podem ser considerados os piores descritores para essa base em específico, pois são termos de uso geral da área da CI. Para se ter uma ideia geral da distribuição da frequência em que os termos do léxico ocorrem, a melhor maneira é plotá-los em um gráfico.

Nesse sentido, o Gráfico 2 foi elaborado com o objetivo de mostrar a quantidade de documentos que cada termo ocorre de forma cumulativa. Ou seja, conforme já descrito anteriormente, como 21.266 termos do total de 46.878 existentes no léxico, ocorrem em apenas um documento. Como a frequência de ocorrência de termos em dois documento corresponde a 6.205, o próximo ponto plotado no gráfico corresponde ao valor acumulados dos dois primeiros casos, e assim sucessivamente. Desse modo, constata-se que somente os termos que ocorrem em até 10 documentos totalizam 41.257, o que corresponde a 88% do léxico. O restante dos termos ocorrem numa frequência superior a 10 documentos.

GRÁFICO 2 – frequência dos termos do léxico pela quantidade de documentos.




Fonte: Elaborado pelo autor.

Os dados para construção do Gráfico 2 foram obtidos a partir da execução do comando (4.2) no MySql.

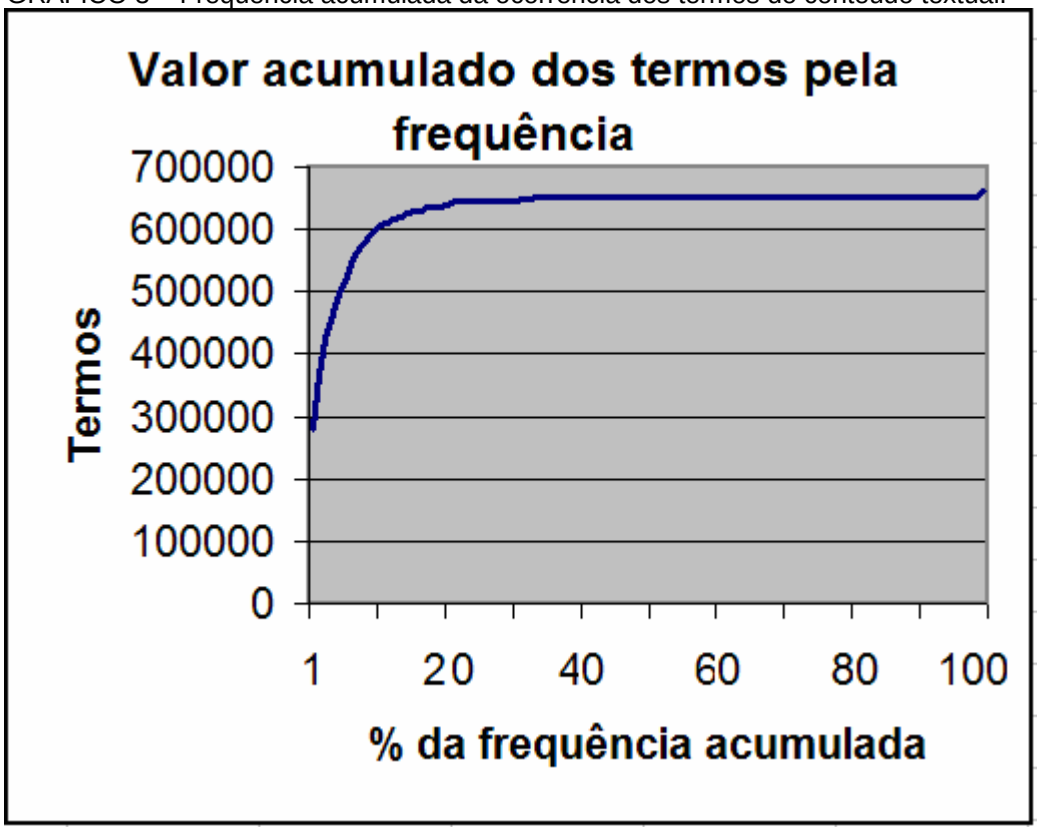
```
SELECT qtd_doc, count(*)
  FROM lexico
 GROUP BY 1
 ORDER BY 1
```
(4.2)

Dando prosseguimento à análise do léxico, cabe avaliar agora qual é a distribuição da frequência em que os termos ocorrem. Nesse sentido, o Gráfico 3 foi elaborado com o objetivo de mostrar a distribuição por quantidade de ocorrências dos termos no conteúdo textual completo a ser indexado de forma cumulativa. Considerando que o conteúdo total dos termos correspondem a 662.194 constata-se que tem-se uma distribuição não uniforme desses dados. Por exemplo, existem 21.268 termos que ocorrem em apenas um único documento. Por outro lado, existe um único termo “informacao” que ocorre 11.942 vezes em todos os documentos. Em outras palavras, a distribuição dos termos se caracteriza por haver muitos termos com baixa frequência e poucos termos numa frequência muito alta. Como os termos que ocorrem uma única vez são 21.266 e duas vezes são 6.682, a curva mostrada no Gráfico 3 apresenta esses valores distribuídos percentualmente. Desse modo, constata-se que somente os primeiros 10% da distribuição da frequência já correspondem a quase o total conteúdo textual. O restante dos termos, cerca de pouco mais de 50.000, estão distribuídos pelos 90% dos demais termos do conteúdo indexado.

Os dados para construção do Gráfico 3 foram obtidos a partir da execução do comando (4.3) no MySql.

```
SELECT qtd_freq, count(*)
  FROM lexico
 GROUP BY 1
 ORDER BY 1
```
(4.3)

GRÁFICO 3 – Frequência acumulada da ocorrência dos termos do conteúdo textual.




Fonte: Elaborado pelo autor.

## 4.2 Segundo experimento exploratório

Inicialmente, para avaliar o uso desse aplicativo foram realizadas buscas no *corpus* utilizando-se como referência vinte documentos aleatórios, dois para cada GT. O cálculo de similaridade utilizado foi o Cosine Similarity Vector. Essa técnica calcula o somatório dos coeficientes de correlação apurados para cada um dos bigramas extraídos do documento de referência e identificados no *corpus.* De modo que, desde que haja pelo menos um bigrama coincidente, entre os extraídos do documento de referência com o *corpus*, ele já passa a ser considerado como parte da resposta. Portanto, pode haver muitos documentos calculados com valores de coeficiente residual. Ou seja, valores bem pequenos. Sendo assim, é conveniente que todo processamento de seleção deva trabalhar com um ponto de corte. A definição desse limiar permite selecionar como resposta apenas os documentos em que o cálculo do seu coeficiente de similaridade seja maior que um percentual parametrizado em relação ao valor do máximo coeficiente apurado entre todos os documentos obtidos como resposta. A Tabela 7 apresenta um estudo exploratório desse comportamento, em que foi observada a quantidade de documentos retornados, considerando vinte buscas realizadas para diversos limites utilizados como ponto de corte.

TABELA 7 – Documentos retornados considerando o ponto de corte.

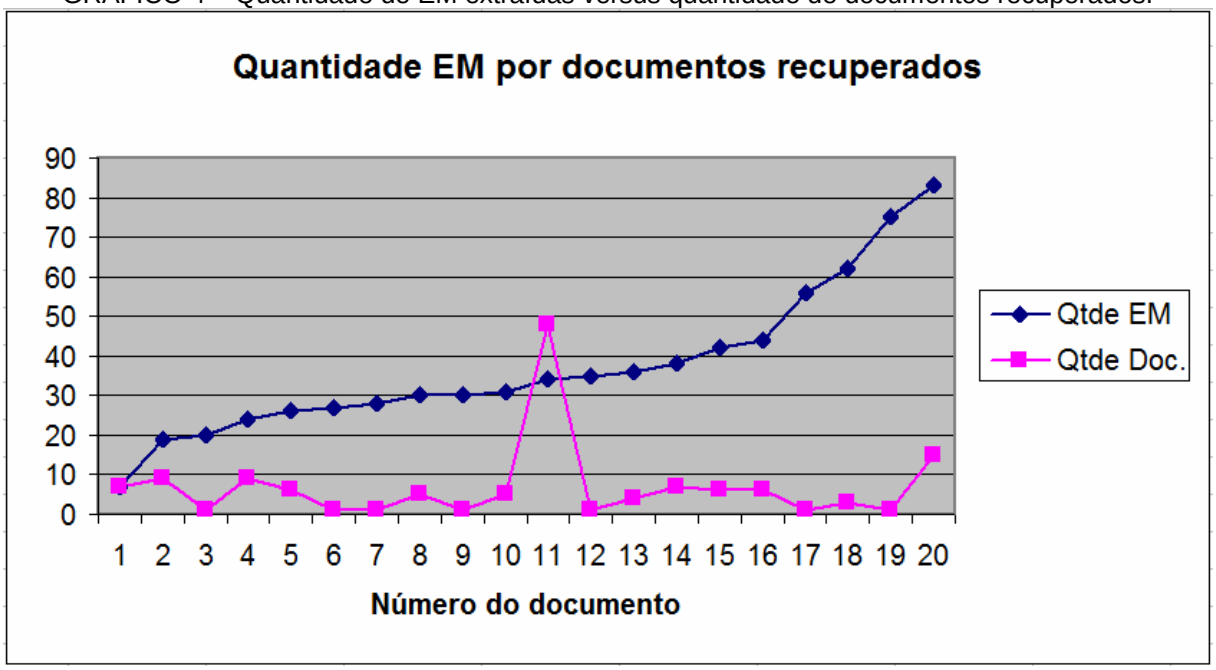
| Ponto de Corte (%) | Quantidade média de documentos retornados |
|---|---|
| 1 | 6,9 |
| 10 | 3,5 |
| 20 | 2,1 |
| 30 | 1,8 |
| 40 | 1,3 |
| 50 | 1,2 |
| 60 | 1,15 |
| 70 | 1,1 |
| 80 | 1 |
| 90 | 1 |
| 100 | 1 |

Fonte: Elaborada pelo autor.

Ao analisar a Tabela 7, percebe-se que o número de documentos retornados reduz à medida que o limiar de corte aumenta, até que apenas um único documento é retornado. Ou seja, apenas o mais similar.

O Gráfico 4 foi gerado utilizando-se como ponto de corte o valor igual a 1%. Ele expressa no eixo das abscissas cada um dos vinte documentos pesquisados e no eixo das ordenadas duas grandezas em valores absolutos. Sendo a primeira, a quantidade de EM identificadas no documento, representada pela curva em azul e, a segunda, a quantidade de documentos retornados pela busca, representada pela curva rosa. Dessa forma, cada coordenada mostrada no gráfico relaciona o valor obtido dessas grandezas através da busca realizada para cada documento. Para facilitar a análise do comportamento dessas grandezas, os documentos foram ordenados de forma crescente pela quantidade de EM extraídas. Entretanto, ao analisar o gráfico constata-se que não há uma relação de dependência direta entre a quantidade de EM identificadas no documento com a quantidade de documentos recuperados. Esse resultado leva a supor que existem outros fatores que contribuem para influenciar esses comportamentos, como, por exemplo: a frequência de ocorrência dos bigramas pesquisados nos demais documentos do *corpus.*

GRÁFICO 4 – Quantidade de EM extraídas versus quantidade de documentos recuperados.




Fonte: Elaborado pelo autor.

A Tabela 8 apresenta a quantidade de documentos em que foram extraídas as EM dentro de determinadas faixas de valores. Ou seja, de 1 a 10; de 11 a 20, etc. Ao analisar esses dados, percebe-se que da maior parte dos documentos foi extraída entre 20 a 60 EM, sendo que o valor médio encontrado foi de 43,5 EM por documento.

TABELA 8 – EM identificadas nos documentos.

| EM identificadas por intervalo de quantidade | Total de documentos |
|---|---|
| 1 a10 | 4 |
| 11 a 20 | 19 |
| 21 a 30 | 51 |
| 31 a 40 | 38 |
| 41 a 50 | 26 |
| 51 a 60 | 27 |
| 61 a 70 | 16 |
| 71 a 80 | 41 |
| 81 a 90 | 1 |
| 91 a 100 | 1 |
| 100 a 110 | 0 |
| 111 a 120 | 1 |
| 121 a 130 | 0 |
| 131 a 140 | 1 |
| Média por documento | 43,5 EM |

Fonte: Elaborada pelo autor.

Existem vários fatores que impactam no processo de RI utilizando EM, dentre os quais destacam-se:

- o fator de co-ocorrência utilizado, que é o limite inferior para se considerar um bigrama como sendo relevante. Neste trabalho o valor utilizado foi quatro;
- o número de EM extraídas, quanto maior esse número, maiores são as chances de se encontrarem documentos similares. Portanto, maior a necessidade de se utilizar um ponto de corte para excluir os coeficientes de similaridade calculados com valores distantes do valor máximo;
- o tamanho do documento. Documentos menores normalmente possuem menor frequência de co-ocorrência dos bigramas;
- o tamanho do *corpus.* Quanto maior a quantidade de documentos existentes na base, maiores são as chances de se encontrar similares;
- os critérios adotados no cálculo do coeficiente de similaridade, que interferem no cálculo da relevância.

## 4.3 Terceiro experimento exploratório

Visando avaliar outros aspectos do comportamento desse aplicativo, elaborou-se um experimento que visa submeter uma busca para cada um dos 194 documentos existentes no *corpus*. Para realizar esse experimento, de busca exaustiva, um novo componente de *software* foi elaborado, denominado “ConsEM.exe”. Esse programa funciona como um robô de consulta, evitando a necessidade de processar as buscas uma a uma através da interface do usuário. O objetivo é automatizar o processo de consulta dos documentos e do registro das respostas produzidas que servirão para avaliar os resultados obtidos.

Assim esse novo programa foi utilizado para processar as 194 consultas e contabilizar quantas foram as EM extraídas para cada documento de referência. A partir desse processamento, verificou-se como sendo 38,6 a quantidade média de bigramas extraídos dos documentos, sendo que os valores máximos e mínimos foram respectivamente 134 e 7 bigramas.

Para entender qual é a influência que a quantidade de bigramas utilizados como descritores impacta na quantidade de documentos recuperados, foi implementado no programa uma requisição solicitando a quantidade de bigramas a serem considerados para o processo de busca. Dessa maneira, a cada documento de referência processado, os seus bigramas são extraídos e inseridos em uma árvore binária, a qual os ordena de forma decrescente pela frequência de co-ocorrência. Consequentemente, no momento de processar a busca dos documentos similares, somente são considerados os “*n”* primeiros

bigramas recuperados da estrutura de árvore. Ou seja, os mais frequentes. Portanto, nesse experimento foram realizadas 194 buscas para cada valor de "*n*" arbitrado. Sendo que, como foram definidos 17 valores distintos para *"n*", cada documento de referência foi consultado 17 vezes, totalizando 3.298 consultas no *corpus.*

Desse modo, foi possível calcular a quantidade de documentos recuperados e a partir deles apurar o valor da quantidade média de documentos recuperados para cada valor de *n.* A Tabela 9 mostra os valores calculados, considerando 1% como sendo o valor de ponto de corte do fator de relevância. Ou seja, são considerados apenas os documentos cujo coeficiente de similaridade calculado corresponda a um valor maior ou igual a 1% do maior coeficiente apurado por documento. Cabe ressaltar que as consultas são realizadas de tal forma que a cada instante é comparado um documento do *corpus* com os demais até que todos tenham sido consultados. Nesse contexto, a cada consulta, sempre o documento retornado como sendo o mais relevante é o próprio documento utilizado como referência da busca. Afinal, nenhum documento pode ser mais similar do que o próprio documento. Isso faz com que, nesse caso, o valor do coeficiente similaridade seja máximo.

TABELA 9 – Comparação da quantidade de descritores *versus* documentos retornados.

| Sequência | Limite *n* de bigramas usados na busca | Quantidade média de documentos |
|---|---|---|
| 1 | 1 | 31,78 |
| 2 | 5 | 20,37 |
| 3 | 10 | 18,91 |
| 4 | 15 | 15,97 |
| 5 | 20 | 15,61 |
| 6 | 25 | 16,17 |
| 7 | 30 | 14,66 |
| 8 | 35 | 14,76 |
| 9 | 40 | 16,28 |
| 10 | 45 | 15,54 |
| 11 | 50 | 14,73 |
| 12 | 55 | 14,78 |
| 13 | 60 | 15,17 |
| 14 | 65 | 14,89 |
| 15 | 70 | 14,68 |
| 16 | 75 | 14,63 |
| 17 | 999 | 14,81 |

Fonte: Elaborada pelo autor.

Ao analisar os dados apresentados na Tabela 9, verifica-se que, ao usar apenas um bigrama - o que na prática funciona com uma busca convencional por palavras-chave, são retornados em média 31,78 documentos. Na medida em que aumenta-se o número de descritores, por exemplo 15, o número de documentos retornados cai pela metade. Ou seja, ocorre melhora na precisão da busca. A partir desse ponto, mesmo aumentando os

descritores até atingir o valor total de EM extraídas, a variação da quantidade média de documentos retornados apresenta uma variação insignificante. Isso leva a concluir que não é necessário estender a busca para todos os bigramas extraídos. É possível limitar a busca para apenas uma parte das EM extraídas mantendo a precisão. Essa estratégia melhora o desempenho da busca.

## 4.4 Quarto experimento, teste de usabilidade

O objetivo desse teste é verificar o uso da ferramenta de busca avaliando os resultados retornados e o tempo de resposta demandados pelas buscas. Nesse sentido, foram realizadas duas buscas aleatórias. Os resultados obtidos são descritos a seguir.

Em todas as pesquisas realizadas, foi informado como ponto de corte o percentual igual a 50%. Ou seja, são apresentandos como respostas da consulta apenas os documentos cujo coeficiente de relevância calculado atingir pelo menos 50% do valor da máxima relevância alcançada na busca. Desse modo, somente serão considerados como respostas aqueles documentos cujo ângulo formado pelos vetores do documento e da consulta estiver na faixa entre zero até cinquenta graus.

Ao pesquisar o documento intitulado “Uma abordagem baseada em métricas de redes complexas para estabelecimento do grau de influência de termos em documentos”, cujo resumo é apresentado no Apêndice F, foram encontrados 39 bigramas com as respectivas frequências de ocorrência conforme mostrados na Tabela 10. Como o tema desse artigo é bem específico dentro da coleção utilizada, pode-se perceber pelos próprios bigramas mostrados na tabela que nessas condições parametrizadas nenhum documento similar foi encontrado.

TABELA 10 – Bigramas e frequência de ocorrência extraídas do documento 172.pdf.

| **Bigrama** | F | **Bigrama** | F | **Bigrama** | F |
|---|---|---|---|---|---|
| redes complexas | 24 | information retrieval | 5 | correspondente numero | 4 |
| recuperacao informacao | 16 | peso termo | 5 | degree centrality | 4 |
| et al | 13 | maior grau | 5 | centralidade rede | 4 |
| termos documentos | 12 | rede medida | 5 | tal modelo | 4 |
| atribuicao pesos | 11 | pesos termos | 5 | numero ligacoes | 4 |
| termo documento | 9 | funcao utiliza | 5 | medida normalizada | 4 |
| coeficiente agrupamento | 9 | calculo similaridade | 5 | precisao interpolada | 4 |
| metricas redes | 8 | documentos consultas | 5 | relacoes sintaticas | 4 |
| grau proximidade | 7 | grau influencia | 5 | analise redes | 4 |
| modelo vetorial | 6 | metricas rede | 5 | consultas documentos | 4 |
| documentos coleção | 6 | complex networks | 4 | similaridade documentos | 4 |
| grau intermediação | 6 | distancias geodesicas | 4 | metodos tradicionais | 4 |
| grau centralidade | 6 | distancia geodesica | 4 | complexas palavras | 4 |

Fonte: Elaborada pelo autor.

Por outro lado, ao pesquisar documentos que lidam com termos de uso mais comum do jargão da área relacionada à coleção, o número de respostas tende a crescer. Por exemplo, ao pesquisar o documento intitulado “Repositório digital da Unati-UNESP: o olhar da arquitetura da informação para a inclusão digital e social de idosos” foram encontrados 12 bigramas conforme mostrados na Tabela 11.

TABELA 11 – Bigramas e frequência de ocorrência extraídas do documento 86.pdf.

| **Bigrama** | F | **Bigrama** | F | **Bigrama** | F |
|---|---|---|---|---|---|
| deste trabalho | 4 | construcao participativa | 4 | digitais idosos | 4 |
| producao intelectual | 4 | informacional digital | 4 | vidotti 2008 | 4 |
| tecnologias informacao | 4 | acessibilidade comportamento | 4 | digital repositorio | 4 |
| grupos focais | 4 | meio grupos | 4 | digital idosos | 4 |

Fonte: Elaborada pelo autor.

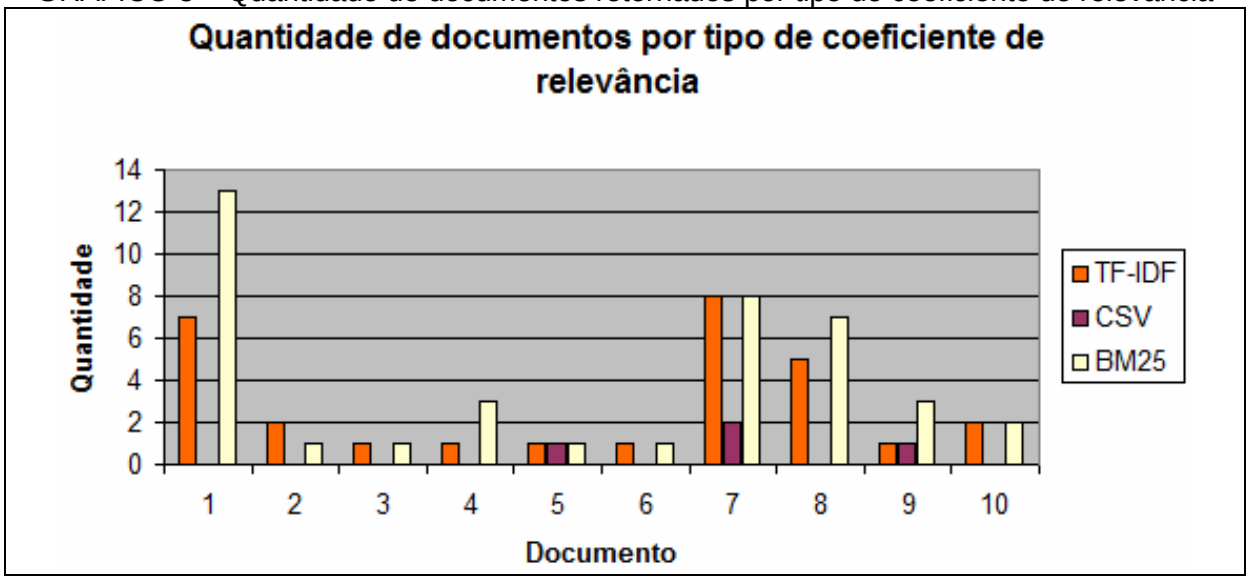
Nesse caso, foram selecionados os seguintes artigos intitulados: “Representação interativa e folkosonomia assistida para repositórios digitais” (384.pdf) e “Contribuição dos repositórios institucionais à comunicação científica: um estudo da Univeridade Federal do Rio Grande do Sul” (186.pdf). Os resumos desses artigos são apresentados no Apêndice F.

## 4.5 Quinto experimento, comparando coeficientes de relevância

Neste experimento o objetivo é comparar as respostas produzidas pelas três diferentes formas de apurar o coeficiente de relevância implementadas pela ferramenta de busca: TF-IDF, CSV e BM25. Nesse sentido, foram retirados aleatoriamente do *corpus* dez documentos. O objetivo é fazer com que os documentos do *corpus* sejam diferentes dos documentos utilizados como referência da busca. Isso fará com que o coeficiente máximo calculado não seja o valor do casamento do documento de busca com ele próprio. Portanto, a coleção de documentos indexada passou a ser de 184 documentos. Desse modo, os dez documentos retirados foram utilizados, um por um, como sendo o documento de referência da Busca Comparada. Para cada documento foram realizadas três buscas, uma para cada técnica. No total foram realizadas trinta consultas através da interface padrão de busca do usuário. Para realizar as consultas os parâmetros de configuração definidos foram: ponto de corte igual a 60%; o limite de bigramas extraídos no documento de referência utilizado foi de até vinte bigramas; parâmetros de ajuste da técnica BM25 foram definidos como sendo K1 e K3 = 2 e B = 0.75. Os resultados obtidos são apresentados no Gráfico 5, o qual mostra a quantidade de documentos retornados pelas três técnicas para cada um dos dez documentos de referência utilizados.

Conforme demonstra o Gráfico 5, as técnicas TF-IDF e BM25 tiveram resultados próximos em termos de quantidade de documentos retornados. Sendo que a BM25 retornou mais documentos do que a TF-IDF em quatro casos e menos, em apenas um caso. Nos demais casos, a quantidade retornada foi a mesma. Por outro lado, CSV não retornou nenhum documento em sete casos e em apenas um a quantidade foi a mesma das demais técnicas empregadas. Esses resultados mostram que a CSV foi a técnica mais restritiva no processo de seleção e que BM25 a que retornou mais documentos.

GRÁFICO 5 – Quantidade de documentos retornados por tipo de coeficiente de relevância




Fonte: Elaborado pelo autor.

Esses resultados tão restritivos obtidos pela CSV podem ser melhor entendidos se considerarmos as características envolvidas em seu processo de cálculo. Em uma consulta composta por até 20 bigramas utilizando CSV, para que um documento seja retornado com o ponto de corte definido em 60%, é necessário que no documento exista cerca de doze bigramas iguais, dos vinte existentes no vetor de consulta. Ou seja, é necessário haver um casamento de muitos termos simultaneamente. Nesse caso, cujo coeficiente varia entre zero e um, para que a similaridade seja máxima, é necessário que todos os bigramas da consulta existam no documento. Por outro lado, as outras duas técnicas trabalham no sentido de encontrar o documento mais relevante. Esse valor apurado é que servirá de base para filtrar os demais documentos que tenham valor de coeficiente maior ou igual a 60% em relação ao valor máximo apurado. Portanto, sempre se terá pelo menos um documento de resposta, desde que ocorra o casamento com os documentos do *corpus* de pelo menos um dos bigramas que compõem a consulta.

Portanto, BM25 pode ser considerada, nesse caso da busca em base de teses, dissertações e artigos, como sendo a técnica mais apropriadade para ser utilizada. Afinal, ela leva em consideração as diferenças de tamanho existentes entre esses tipos de documentos; considera também o peso que a EM tem no documento de referência ao ponderar a frequência do bigrama da consulta no processo de cálculo do coeficiente.

# 5 CONCLUSÕES

Neste capítulo o objetivo é apresentar uma reflexão sobre todo o andamento do processo de pesquisa empreendido. Nesse sentido, verifica-se que os resultados dos experimentos demonstram que o uso da técnica Heudet melhora a precisão da busca. Esse ganho foi medido empiricamente neste trabalho, no qual se demonstrou que, ao utilizar-se essa abordagem proposta, 101 EM foram descartadas, isso representa um ganho de precisão de 1,29%. Cabe ressaltar que neste estudo foi utilizado como valor do parâmetro da frequência de co-ocorrência dos termos do bigrama igual a quatro. Em trabalho publicado anteriormente por Silva e Souza (2012), correlato a esse mesmo tema, o qual utilizou um valor igual a três para esse mesmo parâmetro, também já havia demonstrado resultados similares com ganho de precisão de 1.5%. Portanto, com o número de co-ocorrência menor, implica em um número ainda maior de bigramas para esse mesmo *corpus*, o que resultou em um acréscimo do percentual de descarte.

Com relação ao algoritmo de indexação proposto, constatou-se que ele propiciou uma compressão de 43,36, no conteúdo do *corpus* utilizado, após ter sido convertido para texto, ou seja, reduzindo de 9 Mbytes para 5 Mbytes. Esse é um fator importante, porque grandes bases de documentos exigirão computadores com grande capacidade de memória real. Realizar a compressão sem perder a capacidade de recuperação de dados contribui positivamente no resultado global de uma ferramenta de SRI. Apesar de não terem sido utilizadas neste trabalho, outras técnicas de compressão para criação da lista invertida podem ser implementadas, visando minimizar o uso de memória, a fim de permitir indexar mais documentos utilizando-se uma mesma quantidade de recursos computacionais.

Com relação ao desempenho do *software* na identificação das EM, verificou-se que o tempo de extração de todo o *corpus* consumido pelo algoritmo proposto consumiu 182 segundos, menor do que o consumido pelo processamento do pacote NSP, correspondente a 235 segundos. Embora na prática não haja uma forma de se fazer uma comparação real de tempo, pois cada ferramenta é executada em um sistema operacional diferente a importância dessa medida é para validar a viabilidade do uso da técnica proposta em termos de ferramenta para uso *online* de recuperação da informação. Do mesmo modo, o tempo consumido pelo processo de conversão de todos os documentos do *corpus* do formato pdf para texto, filtragem, normalização dos dados e indexação em memória consumiu 125 segundos para colocar o serviço de busca ativo. Isso demonstra também a viabilidade do algoritmo proposto para uso real como ferramenta de indexação de base de documentos.

Afinal, mesmo que o número de documentos cresça, esse processo, normalmente, é executado apenas uma vez para colocar o serviço de consulta disponível.

Outra medida importante é o tempo de resposta da consulta do usuário. Em todos os testes realizados, o tempo de busca não ultrapassou 4 segundos, um tempo bastante aceitável para realização da Busca Comparada. É importante enfatizar que o custo do algoritmo de consulta não cresce de forma linear com o tamanho do *corpus*, pois o acesso aos termos da busca é realizado numa estrutura em memória com acesso direto ou direto mais um curto caminhamento através de uma lista de colisões em uma estrutura de *hash.* Desse modo, a tendência é de o tempo de resposta *online* se manter mais ou menos constante em diferentes tamanhos de bases de dados indexadas.

Obviamente, o mesmo não ocorre com o algoritmo de indexação da base que cresce lineramente com o tamanho do conjunto de documentos a ser indexado. Como já foi destacado anteriormente, esse processamento de carga é necessário apenas na inicialização do serviço de busca. Após o serviço estar disponibilizado, ele pode permanecer disponível por tempo indeterminado e incluir novos documentos de forma incremental, mantendo, ao mesmo tempo, o serviço de busca disponível.

Vistos esses números, conclui-se que o algoritmo Heudet apresenta vantagens em relação ao uso das técnicas estatísticas. Isso se dá pelo fato de ele levar em consideração a estrutura do documento. Afinal, ele considera o documento como um conjunto de sentenças, em vez de, um conjunto de palavras como é trabalhado pelos algoritmos estatísticos do pacote NSP. Em relação ao desempenho, verificou-se que há viabilidade de uso, porque o tempo consumido durante o processamento de todas as funcionalidades do *software* foram bem razoáveis.

Os resultados dos experimentos demonstram também que o uso da técnica Heudet melhora a precisão da busca, tendo em vista que, ao combinar o uso de vários descritores no processo de busca torna o resultado apresentado como resposta a união entre as combinações de vários bigramas concomitantes.

No teste empírico realizado, verificou-se que usando-se a partir de quinze EM utilizadas como descritores, a quantidade de documentos retornados reduziu pela metade em comparação com um acesso realizado com apenas um descritor. Desse modo, quanto mais bigramas, dentre os extraídos do documento de referência forem coincidentes com os encontrados nos documentos do *corpus,* maior será o valor apurado no cálculo da relevância. Afinal, a relevância é apurada pelo somatório das relevâncias de cada par de

bigramas possibilita produzir um resultado mais restrito, que tenha não apenas um termo similar, mas sim um conjunto de bigramas relacionados.

A técnica BM25, considerada o estado da arte dos SRI, foi a técnica que apresentou o melhor resultado nos testes empíricos realizados. Mostrou-se como a mais apropriada para ser utilizadas em processos de recuperação de documentos de base de dados de teses, dissertações e artigos, por ponderar a relevância em função do tamanho do documento e pelo peso do bigrama da consulta em função de sua frequência no documento de referência. Desse modo, os bigramas da consulta têm sua importância relativizada correspondendo à sua relevância no documento da busca, o que não é possível de ser obtida pelas demais técnicas.

# 6 TRABALHOS FUTUROS

Neste capítulo, sob à luz dos resultados obtidos retoma-se o objetivo apresentado neste trabalho: “Propor e analisar comparativamente uma metodologia de recuperação de informação que utiliza um documento como referência para a busca em um *corpus* específico. Ou seja, a Busca Comparada, sendo que a função de similaridade será medida através da ocorrência de expressões multipalavra.”

Conforme os resultados demonstraram ao comparar-se a busca tradicional por palavras-chave com o método proposto de Busca Comparada, o número de documentos retornados tende a ser menor, e, baseado na construção de significado, a partir de vários bigramas coincidentes, traz uma melhora na precisão das respostas. Desse modo, a técnica elaborada de extração das EM para utilizá-las como descritores em uma ferramenta de busca se mostrou bastante viável, podendo ser implementada de forma integrada com ferramentas de RI em base de dados de documentos digitais.

Ao avaliar os resultados obtidos, verifica-se que eles não apenas atendem aos objetivos propostos, como ao mesmo tempo abrem um caminho para se retormar a motivação deste trabalho, as bibliotecas digitais de teses, dissertações e artigos. O protótipo desenvolvido, durante todo o tempo de maturação e construção pessoal de conhecimentos das técnicas e teorias envolvidas no processo de recuperação de informação, materializam na concepção de uma metodologia automatizada de recuperação da informação que combina técnicas existentes e também novas ideias produzindo um conjunto teórico prático, possibilitando a implementação de uma nova ferramenta de *software.* Essa ferramenta pode ser integrada a um processo de busca que considera os metadados dos documentos. Essas funcionalidades juntas aumentam a capacidade seletiva das respostas produzidas. Um outro aspecto a ser observado é que o projeto do *software* foi desenvolvido em camadas e numa arquitetura cliente servidor, o que possibilita uma boa escalabilidade. Desse modo, a ferramenta pode ser facilmente adaptada para realizar pesquisas em múltiplas bases de dados, sendo cada base de dados considerada como um serviço de busca disponibilizado em um computador provedor do um serviço diferente.

Portanto, este trabalho poderá derivar no futuro em um trabalho aplicado a fim de operacionalizar o uso dessa ferramenta em um ambiente real de uma biblioteca digital. Isso poderá ser feito através da integração dessa ferramenta proposta com uma aplicação existente, ou mesmo implementar outras funcionalidades a ela, de forma a agregar metadados dos documentos em um banco de dados relacional, a fim de se utilizar para a

busca a soma desses vários critérios e de forma simultânea. Tudo isso se conduz ao encontro da motivação de tornar as ferramentas de busca cada vez mais precisas em seus resultados retornados.

Contudo, o trabalho de pesquisa não para aqui. Novas heurísticas que visam agregar a extração de sentido do texto, ou mesmo melhorar o processo de compressão de dados, poderão ser implementadas e validadas possibilitando uma retro-alimentação entre o trabalho teórico e o prático.

# REFERÊNCIAS


1 BAEZA-YATES, Ricardo; RIBEIRO-NETO, Berthier. Modern Information Retrieval. New York: ACM Press, 1999. 511p.

2 BANERJEE, Satanjeev; PEDERSEN, Ted. The design, implementation and use of NSP, 2000.

3 BARROS, Enéas Martins. Gramática da Língua Portuguesa**.** São Paulo, Editora Atlas, 1991.

4 BERGER, Peter L.; LUCKMANN, Thomas. A construção da realidade. 23 ed. Petrópolis: Vozes, 2003. 249p.

5 CALZOLARI, Nicoletta et al. Towards best practice for multiword expressions in computational lexicons. Em Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002), pp. 1934–1940, Las Palmas, Canary Islands, 2002.

6 CHEN, Jisong, YEH, Chung-Hsing, CHAU, Rowena. A multi-word term extraction system. PRICAI 2006, LNAI 4099, pp. 1160 – 1165, 2006. Springer-Verlag Berlin Heidelberg, 2006.

7 CINTRA, Anna Maria Marques. Elementos de linguística para estudos de indexação. **Ciências de Informação,** v.12, n. 1, p. 5-22, 1983.

8 CIPRO NETO, Pasquale; INFANTE, Ulisses. Gramática da língua portuguesa. São Paulo. Ed. Scipione, 2009. 584p.

9 DIAS, Gael; LOPES, José Gabriel Pereira; GUILLORÉ, Sylvie. Mutual expectation: a measure for multiword lexical unit extraction. In Proceedings of Vextal, 1999.

10 FARACO, Carlos Emílio; MOURA, Francisco Marto, Gramática, 7. ed. São Paulo: Ática, 1990. 487p.

11 EVERT, Stefan; KRENN, Brigitte. Using small random samples for the manual evaluation of statistical association measures. Computer Speech and Language, 19(4):450–466, 2005.

12 KURAMOTO, Hélio. Uma abordagem alternativa para o tratamento e a recuperação da informação textual: os sintagmas nominais. **Ciência da Informação**, Brasília v. 25, n. 2, mai/ago, p. 182-196, 1995.

13 LADEIRA, Ana. Paula. Processamento de linguagem natural: caracterização da produção científica dos pesquisadores brasileiros. 2010. 262f. Tese (Doutorado em Ciência da Informação), Escola de Ciência da Informação da UFMG, Belo Horizonte, 2010.

14 LANCASTER, F. Wilfrid; WARNER, Amy. J. Information retrieval today. Information Resource, 1993.

15 MAIA, Luiz Cláudio Gomes; SOUZA, Renato Rocha. Uso de sintagmas nominais na classificação automática de documentos eletrônicos. **Perspectivas em Ciência Informação**, Belo Horizonte, v. 15, p. 154-172 , 2010.

16 MANNING, Chistopher D.; SCTÜTZE, Hinrich. Foundations of statistical Natural Language Processing. MIT, 2003. 680p.

17 MANNING, Christopher D.; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich An introduction to information retrieval. Ed. Cambridge online, 2009.

18 MIKHEEV, A. Periods, capitalized words, etc. Computacional Linguistics, 28(3), 289-318, 2002.

19 NUNES, M. G. V., VIEIRA, R.; LIMA, Vera Lúcia Strobe. SBC-Comissão Especial do Processamento da Linguagem Natural. 2007. Disponível em: <http://www.nilc.icmc.usp.br/cepln>. Acesso em: mar. 2011.

20 OLIVEIRA, F.A.D. Processamento de linguagem natural: princípios básicos e a implementação de um *analisador sintático* de sentenças da língua portuguesa. Disponível em: <http://www.inf.ufrgs.br/~gppd> acesso em 25 de fevereiro de 2011.

21 OLIVEIRA, Itamar Leite de; WAZLAWICK, R. S. A Modular Connectionist Parser for Resolution of Pronominal Anaphoric References in Multiple Sentences. In: International Joint Conference on Neural Network, 1998, anchorage, Alaska. IEEE World Conference on Computational Intelligence – IEEE/WCCI-98, 1998. v. 2. p. 1194-1199.

22 PEARCE, Darren. A comparative evaluation of collocation extraction techniques. In Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002), Las Palmas, Canary Islands, Spain, May, 2002. European Language Resources Association.

23 PECINA, Pavel. Lexical association measures and collocation extraction. Language Resources and Evaluation (LREC 2010) 44(1-2): 137-158, 2010.

24 PEDERSEN, Ted et al. The Ngram Statistics Package. Disponível em: http://www.d.umn.edu/~tpederse/nsp.html. Acesso em: ago. 2011.

25 PORTELA, Ricardo; MAMEDE Nuno; BAPTISTA, Jorge. Mutiword Identificação. In Terceiro Simpósio de Informáctica (INFORUM 2011), Oct. 2011, pp.

26 RAMISCH, Carlos. Multiword terminology extraction for domain specific documents. Dissertação – Mathématiques Appliqueées, École Nationale Supérieure d'Informatiques, Grenoble, 2009.

27 RANCHHOD, Elisabete Marques. O lugar das expressões 'fixas' na gramática do Português. in Castro, I. and I. Duarte (eds.), Razão e Emoção, vol. II, Lisbon: INCM, pp. 239-254, 2003.

28 RAYSON, Paul; PIAO, Scott; SHAROFF, Serge; EVERT, Stefan. MOIRÓN, Begoña Villada. Multiword expressions: hard going or plain sailing? Springer Science Business Media B. V, 2009.

29 ROBERTSON, S. E.; SPARCK Jones, K. Relevance weighting of search terms, **Journal of the American Society for Information Science**, Volume 27, 1976 pp. 129–146, 1976.

30 RODRIGUES, Jorilson; CARICATTI, André. A pragmática no contexto da identificação de autoria de textos. **Ciências de Informação,** v.38, n. 1, p. 124-133, 2009.

31 ROUSSINOV, Dmitri. Towards Combined Aspect Verification Model. (no prelo).

32 RUSSELL, Stuart J; NORVIG, Peter. Inteligência Artificial. Rio de Janeiro: Campus, 2004. 1021p.

33 SAG, I. A. et al. Multiword expressions: a pain in the neck for nlp. Em Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing CICLing-2002), volume 2276 of (Lecture Notes in Computer Science), pp. 1–15, London, UK. Springer-Verlag, 2002.

34 SALTON, Gerard; MCGILL, M.J. Introduction to modern information retrieval. New York: McGraw-Hill Book Company, 1983. 488p.

35 SARACEVIC, Tefko. Ciência da Informação: origem, evolução, relações. **Perspectivas em Ciência da Informação**, Belo Horizonte, v. 1, n. 1, p. 41-62, jan/jun 1996.

36 SARMENTO, Luís. Simpósio Doutoral Linguateca 2006. Disponível em: http://www.linguateca.pt/documentos/SimposioDoutoral2005.html: out. 2011.

37 SETZER, Valdemar W. Meios eletrônicos e educação: uma visão alternativa. São Paulo: Escrituras. 2001.

38 SILVA, Edson Marchetti; SOUZA, Renato Rocha. Information retrieval system using multiwords expressions (MWE) as descriptors. **JISTEM** - Journal of Information Systems and Technology Management Vol. 9, No. 2, Mai/Aug. 2012, pp.213-234.

39 SILVA, João Ricardo Martins Ferreira. Shallow processing of portuguese: from sentence chunking to nominal Lemmatization. 2007, 196f. Dissertação (mestrado em Informática), Faculdade de Ciências, Universidade de Lisboa, 2007.

40 SILVA, Joaquim Ferreira; LOPES, Gabriel Pereira. A local maxima method and fair dispersion normalization for extracting multi-word units from corpora. Sixth meeting on Mathematics of Language, pp. 369-381, 1999.

41 SMITH, Barry; CEUSTERS Werner. Ontological realism: A methodology for coordinated evolution of scientific ontologies. Ontology, *5*(3), 1. 2010.

42 SINGHAL, Amit; SALTON, Gerald; MITRA, Mandar; BUCKLEY, Chris. Document length normalization. **Information Processing and Management - IPM** , vol. 32, no. 5, pp. 619-633, 1996.

43 SOUZA, Renato Rocha. Uma proposta de metodologia para a escolha automática de descritores utilizando sintagmas nominais. 2005. 215f. Tese (Doutorado em Ciência da Informação), Escola de Ciência da Informação, UFMG, Belo Horizonte, 2005.

44 SOUZA, Renato Rocha. Sistemas de recuperação de informações e mecanismos de busca na web: panorama atual e tendências. **Perspectivas em Ciência Informação**, Belo Horizonte, v. 11, n. 2, Aug. 2006.

45 TÁLAMO, M.F. Informação: organização e comunicação. Seminários de Estudos de Informação da Universidade Federal Fluminense, 1, 1996 Anais... Niterói Rio de Janeiro: EDUFF, 2007. p.11-14.

46 VIGOTSKY, Lev Semenovich. Pensamento e Linguagem. Edição eletrônica: Ed. Ridendo Castigat, 1987.

47 VILLAVICENCIO, Aline et al. Identificação de expressões multipalavra em domínios específicos. **Linguamática**, v. 2, n. 1, p. 15-33, abril, 2010.

48 WANG, Lijuan; LIU, Rong. A Rapid Method to Extract Multiword Expressions with Statistic Measures and Linguistic Rules. WISM 2011, Part II, LNCS 6988, pp. 234–241, 2011.

49 YAGONOVA, E. V.; PIVOVAROVA, L.M. The Nature of Collocations in the Russian Language. The Experience of Automatic Extraction and Classification of the Material of News Texts. Automatic Documentation and Mathematical Linguistics, 2010, Vol. 44, No. 3, pp. 164–175. Allerton Press, Inc., 2010.

50 ZHANG, Wen; et al. Improving effectiveness of mutual information for substantival multiword expression extraction. **Expert Systems with Applications**, Elsevier, v. 36, 2009.

51 ZHAI, Cheng Xiang. Statistical Language Models for Information Retrieval A Critical Review**.** Foundation and Trend in Information retrieval. Vol. 2, no 3, 2008. p, 137-213.

# APÊNDICE A

**Script GeraCount**

```
#Autor Edson marchetti
#!/bin/sh
path="/windows/Users/Edson/Documents/ECI/TESE/CorpusCI/Enancib2010/Artigo"
freq="--frequency 4"
cd $path
find . -name *.txt -print > tmp
for i in `sed "s/\.//" tmp`
do
   origem=$path$i
   destino=$path`dirname $i`/`basename $i .txt`".count"
   echo "executando count para $i"
   rm $destino 2> /dev/null
   count.pl $destino $origem
   origem=$destino
done
```

**Script GeraEM**

```
#Autor Edson marchetti
#!/bin/sh
path="/windows/Users/Edson/Documents/ECI/TESE/CorpusCI/Enancib2010/Artigo"
freq="--frequency 4"
cd $path
find . -name *.txt -print > tmp
for i in `sed "s/\.//" tmp`
do
  for op in dice left right twotailled rfisher lfisher jaccard ll tmi ods
pmi phi x2 ps tscore
  do
      echo "executando coeficiente de $op $origem"
      destino=$path`dirname $i`/`basename $i .txt`".$op"
      rm $destino 2> /dev/null
      statistic.pl $freq $op $destino $origem
  done
done
```

**Roteiro do processamento na ambiente Linux**

Para executar o pacote NSP no ambiente Linux é necessário executar os seguintes passos:

1. Executar o programa Server.exe com a opção “CONVERTETXTPDF S” definido em server.ini, a fim de gerar o arquivo com a extensão .txt;
2. Logar no linux e abrir um terminal e executar os comandos:
   - sudo mount /dev/sda1 /windows
   - cd NSP
   - sh GeraCount
   - sh GeraEm

- sh Rank

Obs: Utilizar frequência = 4 como parâmetro do processamento.

Como o sistema de arquivos do Linux reconhece a estrutura do Windows, todos os arquivos são gerados dentro da estrutura de arquivos Windowns definidas para o *corpus.*

No ambiente Windows deve executar os programa Rank.exe e Client.exe (com parâmetro “GERAEM S”. Em seguida após inicializar o ambiente Wampserver executar o programa AtuMySql. Esse programa irá inserir os conteúdos, das EM geradas pelo pacote NSP e mais pelo algoritmo determinístico Heudet (através do programa GeraEm), na tabela docmetrica dentro do banco de dados Métrica.

**Script compara**

```
#Autor Edson marchetti
#!/bin/sh
path="/windows/Users/Edson/Documents/ECI/TESE/CorpusCI/Enancib2010/Artigo"
cd $path
find . -name *.txt -print > tmp
for i in `sed "s/\.//" tmp`
do
   arqBase=$path`dirname $i`/`basename $i .txt`.rank
   for n1 in dice jaccard ll odds phi pmi ps tmi tscore twotailed lfisher
rfisher x2
   do
       for n2 in dice jaccard ll odds phi pmi ps tmi tscore twotailed
lfisher rfisher x2
       do
           echo "$arqBase = $n1  -  $n2\n"
           arqbase=$path$i
           arq1=$path`dirname $i`/`basename $i .txt`.$n1
           arq2=$path`dirname $i`/`basename $i .txt`.$n2
           cmd="rank.pl $arq1 $arq2 >> $arqBase"
           echo "$cmd"
           rank.pl $arq1 $arq2 >> $arqBase
       done
   done
done
```

# APÊNDICE B

## Lista de Stop Words - Normalizadas

a
agora
ainda
alguem
algum
alguma
algumas
alguns
ampla
amplas
amplo
amplos
ante
antes
ao
aos
apos
aquela
aquelas
aquele
aqueles
aquilo
as
ate
atraves
cada
coisa
coisas
com
como
contra
contudo
da
daquele
daqueles
das
de
dela
delas
dele
deles
depois
dessa
dessas
desse
desses
desta
destas
deste
destes
deve
devem
devendo
dever
deverá
deverão
deveria
deveriam
devia
deviam
disse
disso
disto
dito
diz
dizem
do
dos
e
ela
elas
ele
eles
em
enquanto
entre
era
essa
essas
esse
esses
esta
estamos
estao
estas
estava
estavam
estavamos
este
estes
estou
eu
fazendo
fazer
feita
feitas
feito
feitos
foi
for
foram
fosse
fossem
grande
grandes
ha
isso
isto
ja
la
lhe
lhes
lo
mas
me
mesma
mesmas
mesmo
mesmos
meu
meus
minha
minhas
muita
muitas
muito
muitos
na
nao
nas
nem
nenhum
nessa
nessas
nesta
nestas
ninguem
no
nos
nossa
nossas
nosso
nossos
num
numa
nunca
o
os
ou
outra
outras
outro
outros
para
pela
pelas
pelo
pelos
pequena
pequenas
pequeno
pequenos
per
perante
pode
podendo
poder
poderia
poderiam
podia
podiam
pois
por
porem
porque
posso
pouca
poucas
pouco
poucos
primeiro
primeiros
propria
proprias
proprio
proprios
quais
qual
quando
quanto
quantos
que
quem
sao
se
seja
sejam
sem
sempre
sendo

sera
serão
seu
seus
si
sido
so
sob
somos
sua
suas
talvez
tambem
tampouco
te
tem
tendo
tenha
ter
teu
teus
ti
tido
tinha
tinham
toda
todas
todavia
todo
todos
tu
tua
tuas
tudo
ultima
ultimas
ultimo
ultimos
um
uma
umas
uns
vendo
ver
vez
vindo
vir
vos
in
and
são
cada
pode
alem
ainda
sobre
assim

# APÊNDICE C

## Configuração do componente de software Server

O componente de *software* denominado Server é o responsável por processar todos os documentos do *corpus* através das etapas dois até quatro descritas da primeira fase da metodologia. Esse componente é constituído de um programa executável cuja função principal é preparar os documentos do *corpus* em uma estrutura de dados, de lista invertida, em memória, a qual permite uma rápida recuperação da informação através de uma busca por termos. Ele pode ser instalado separadamente em um computador que terá o papel de prover um serviço de consulta aos dados através da porta de comunicação de rede. O arquivo de configuração (server.ini) permite definir diversos parâmetros para o processamento do programa.

Para esse *software* ser executado é necessário instalar em um diretório os seguintes arquivos:

- server.exe – programa executável gerado através de código fonte C++;
- libtet.dll – biblioteca *freeware* versão simplificada, sob licença de TET LIB;
- server.ini – arquivo texto usado para configurar as funcionalidades do *software*;
- lista.txt – arquivo texto com a relação do caminho e nome dos documentos a processar;
- StopWord.txt – arquivo com a lista de palavras que serão descartadas.

O *software*, durante o seu processamento gera um arquivo (server.log) que contém várias informações do processamento intermediário que podem ser geradas ou não conforme definido no arquivo server.ini. Cabe ressaltar que o processo de geração de log consome tempo de processamento e por isso normalmente deve estar desativado.

A seguir apresenta-se cada uma das opções de configurações do arquivo server.ini. O arquivo tem a seguinte estrutura: o caracter # na primeira coluna indica que a linha é de comentário; a variável aparece sempre como sendo uma palavra cujas letras estão em maiúsculas; entre o nome e o valor da variável deve sempre existir pelo menos um caracter em branco.

A seguir são apresentadas as variáveis que podem ser configuradas.

| Nro | Nome da Variável | Nome da Variável |
|---|---|---|
| 1 | NIVELLOG | Define como será a forma de gravação do arquivo (server.log). Ao informar: 0 (zero) salva todas as mensagens informativas de acompanhamento do processamento no arquivo de log; 1 (um) salva somente mensagens de erro e demais mensagens que forem explicitamente informadas através de parâmetro. Como regra deve-se usar o valor igual a 1, pois reduz o tamanho do log gerado. |
| 2 | STOPWORD | Define o caminho e o nome do arquivo que contém as *stop words*. Ao colocar apenas o nome, o arquivo será procurado no mesmo diretório em que a aplicação for executada. |
| 3 | PATHCORPUS | Caminho absoluto, em relação à raiz do sistema de arquivos, onde o *corpus* de documentos está localizado. Não é necessário colocar o caracter de barra "/" ao fim do caminho informado. |
| 4 | NOMLISTA | Define o *path* (opcional se o arquivo estiver na mesma pasta do executável) e o nome do arquivo que contém a lista com os nomes dos documentos a processar. |
| 5 | NOMLISTAINC | Define o *path* (opcional se o arquivo estiver na mesma pasta do executável) e o nome do arquivo que contém a lista complementar com os nomes dos documentos a processar a indexação incremental na coleção de documentos. |
| 6 | PORTA | Número da porta TCP/IP em que o serviço será disponibilizado. O valor default é a porta 8000. |
| 7 | MOSTRASW | Grava no arquivo server.log quais são as *stop words* que o processamento utilizará. Os valores informados podem ser S ou N. |
| 8 | MOSTRATEXTO | Grava no arquivo server.log uma versão de cada documento indexado em formato texto, após extrair os cabeçalhos, as referências, identificar e expandir as siglas e retirar a acentuação das palavras. Os valores informados podem ser S ou N. |
| 9 | MOSTRAORACAO | Grava no arquivo server.log os segmentos de texto identificando e numerando as orações. Os valores informados podem ser S ou N. |
| 10 | MOSTRATERMO | Grava no arquivo server.log identificando e numerando os termos. Os valores informados podem ser S ou N. |
| 11 | MOSTRAACRONIMO | Grava no arquivo server.log todos os acrônimos que foram identificados durante o processamento do texto. Os valores informados podem ser S ou N. |
| 12 | MOSTRAINDEXADO | Grava no arquivo server.log todos os termos que foram indexados. Os valores informados podem ser S ou N. |

| | | |
|---|---|---|
| 13 | CONVERTEPDFTXT | Grava um arquivo com o mesmo nome do documento processado e no mesmo diretório trocando a extensão de (.pdf) para (.txt). O arquivo txt gerado é a versão final utilizada na indexação dos termos, retirando as *stop words* as pontuações e transformando todos os termos em minúsculas. Esse arquivo gerado servirá de base para o processamento da quarta fase da metodologia. Os valores informados podem ser S ou N. |
| 14 | GERALEXICO | Define se gravará no arquivo server.log todos os termos com as informações da frequência no *corpus* e também a quantidade de documentos da coleção que contem esse termo. |
| 15 | PERCORTE | Define o valor do percentual de corte para não recuperar o documento. O percentual será aplicado ao maior coeficiente a fim de desprezar valores abaixo desse limiar definido. O valor informado pode variar de 0 a 100. Sendo que 0 traz todos, 100 traz apenas o mais similar de todos. |
| 16 | ECHOIDX | Define se gravará no arquivo server.log as totalizações do processo de indexação, antes de colocar o servidor de consulta no ar. Os valores informados podem ser S ou N. |
| 17 | ECHOMSG | Define se gravará no arquivo server.log todas as requisições de consulta por termos solicitadas pela aplicação Client e as respostas enviadas pelo Server. Os valores informados podem ser S ou N. |
| 18 | ECHOPROC | Define se gravará no arquivo server.log as requisições de consulta por termos solicitadas pela aplicação Client. Os valores informados podem ser S ou N. |
| 19 | ECHOCALC | Define se gravará no arquivo server.log os valores intermediários utilizados no cálculo do coeficiente de ranqueamento. Os valores informados podem ser S ou N. |
| 20 | OPCCALC | Define qual será a forma utilizada no cálculo do valor da relevância. Existem três possibilidades implementadas:<br>1 – TF-IDF;<br>2 – Cosine Similarity Vector;<br>3 – BM25. |
| 21 | K1 | Parâmetro de ajuste utilizado no cálculo da técnica BM25. Está relacionado com o ajuste da escala da frequência do termo no documento pesquisado em relação ao valor final apurado do coeficiente de relevância. Se K1 = 0 corresponde ao modelo binário, não considera a frequência do termo. Ao aumentar o valor de K1 implica em aumentar a relevância atribuída à frequência do termo. |

| | | |
|---|---|---|
| 22 | K3 | Parâmetro de ajuste utilizado no cálculo da técnica BM25. Está relacionado com o ajuste da influência da frequência do bigrama informado na consulta em relação ao valor final apurado do coeficiente de relevância. Se K3 = 0 implica em desconsiderar a frequência em que o bigrama foi encontrado na consulta. Ao aumentar o valor de K3, implica em aumentar a relevância em que a frequência do bigrama da consulta terá na apuração do valor final da relevância. |
| 23 | B | Parâmetro de ajuste utilizado no cálculo da técnica BM25. Está relacionado com o ajuste da influência em que o tamanho do documento afeta no valor final apurado do coeficiente de relevância. Valor de B deve ser ($0 \leq b \leq 1$). Se B = 0 implica que não há normalização de tamanho. Se B = 1 implica ponderação máxima ao valor do tamanho. |

A seguir são apresentadas como as informações são geradas no arquivo server.log de acordo com cada um dos parâmetro definidos:

MOSTRASW

```
Existem 211 palavras na lista de stop words
Mostrando as stop words do hashing
    0 - Stop Word [estiveram]
    5 - Stop Word [vos]
    8 - Stop Word [sou]
  ...
  210 - Stop Word [por]
  211 - Stop Word [nos]
Fim das Stop Words
```

MOSTRATEXTO

```
GT 10: Informacao e Memoria
Modalidade de apresentacao: Poster
CIENCIA DA INFORMACAO E MUSEUS DE ARTE: DIALOGOS E INTERACOES NO
ACESSO AS INFORMACOES DO ACERVO DO NUCLEO DE ARTE
CONTEMPORANEA DA PARAIBA
Thais Catoira
Universidade Federal da Paraiba
Resumo: O objeto de estudo dessa pesquisa - que resultara em uma
dissertacao a ser defendida...
```

MOSTRAORACAO / MOSTRATERMO

```
ORAÇÃO 0
GT 10: Informacao e Memoria

TERMO 1 - gt
TERMO 2 - 10
TERMO 3 - informacao
TERMO 4 - memoria

ORAÇÃO 1
Modalidade de apresentacao: Poster

TERMO 1 - modalidade
TERMO 2 - apresentacao
TERMO 3 - poster

ORAÇÃO 2
CIENCIA DA INFORMACAO E MUSEUS DE ARTE: DIALOGOS E INTERACOES NO

TERMO 1 - ciencia
TERMO 2 - informacao
TERMO 3 - museus
TERMO 4 - arte
TERMO 5 - dialogos
TERMO 6 – interações
...
```

MOSTRAACRONIMO

```
***  Acronimos do documento  ***
Acronimo (pibic)  programa institucional bolsas iniciacao cientifica
Acronimo (probex) programa bolsas extensao extensao
Acrônimo (ibict)  instituto brasileiro informacao ciencia tecnologia
Acronimo (bdtd)   biblioteca digital teses dissertacoes
Acronimo (seer)   sistema eletronico editoracao revistas)
Acronimo (capes)  coordenacao aperfeicoamento pessoal nivel superior
Acronimo (ppgci)  programa programa posgraduacao ciencia informacao
Fim Acronimos
```

MOSTRAINDEXADO

```
Mostrando os Descritores do hashing
   53 - Termo [01] - Qtde doc [    1] Freq [   1]
        Doc [    0] Freq [   1]
        Posicao oracao-palavra [  102]-[   7] peso[1]
   56 - Termo [06] - Qtde doc [    2] Freq [   2]
        Doc [    1] Freq [   1]
        Posicao oracao-palavra [  213]-[   1] peso[1]
        Doc [    2] Freq [   1]
        Posicao oracao-palavra [  146]-[   1] peso[1]
      ...
```

GERALEXICO

```
Mostrando os termos distintos
raca        9    14
cafe       10    30
each       21    28
fase       73   206
lado      120   342
gira        4     4
vale       57   110
escrava     3     3
cerrado     1     1
vendido     1     1
enxerga     3     3
anexado     3     3
...
```

ECHOIDX

```
*** Totalização dos documentos indexados ***

Arquivo:
/users/edson/documents/eci/corpusCI/Enancib2010/Poster/GT1/292.pdf
Tamanho do arquivo: 2239 Kbytes
Qtde Páginas: 7
Data da Criação: 01/04/2011  11:12
Qtde Termos: 1131  Tamanho: 8887  bytes
Qtde Termos distintos: 574  Tamanho: 4815  bytes

Arquivo:
/users/edson/documents/eci/corpusCI/Enancib2010/Poster/GT2/114.pdf
Tamanho do arquivo: 2239 Kbytes
Qtde Páginas: 9
Data da Criação: 23/03/2011  14:48
Qtde Termos: 1259  Tamanho: 9954  bytes
Qtde Termos distintos: 394  Tamanho: 3320  bytes

           Total geral

Total de Documentos; 4 Tamanho Total: 8959 Kbytes
Qtde Termos: 4663  Tamanho: 35 Kbytes
Qtde Termos distintos: 1426  Tamanho: 11 Kbytes

*** Fim da Totalização dos documentos indexados ***
```

ECHOMSG

```
Mensagem Recebida 152 bytes
politica indexacao
indexacao artigos
artigos periodicos
secao periodicos
biblioteca central
universidade federal
federal paraiba
vocabulario controlado

Mensagem Recebida 4 bytes
#end

Processando consulta

Resposta da Consulta
doc = 19 coef 3.786003
doc = 58 coef 0.016525
doc = 20 coef 0.015776
doc = 62 coef 0.003878
Foram encontrados 4 documentos similares

*** Retorno servidor ***
Docto  Peso
Mensagem Enviada 473 bytes
<tr><td>
3.786003</td><td><h3>/users/edson/documents/eci/tese/corpusCI//users
/edson/documents/eci/tese/corpusCI/Enancib2010/Poster/GT2/274.pdf</h
3></td><td>GT 2 - Organizacao e Representacao do Conhecimento
Modalidade de apresentacao: poster
AVALIACAO DA POLITICA DE INDEXACAO DE ARTIGOS DA SECAO DE
PERIODICOS          DA          BIBLIOTECA          CENTRAL</td><td><a
href="corpusCI//users/edson/documents/eci/tese/corpusCI/Enancib2010/
Poster/GT2/274.pdf"</a><img src= 'imagens/pdf.gif'></td></tr>
Resposta = OK
```

ECHOPROC

```
Pesquisando o termo [0-politica] da Expressão Multipalavra
Pesquisando o termo [1-indexacao] da Expressão Multipalavra
Termo [0-politica] alinhado no docto 19
Termo [1-indexacao] alinhado no docto 19
      Termo [0-politica] alinhado na oracao 3
      Termo [1-indexacao] alinhado na oracao 3
            Termo [0-politica] na oraçao 3 na posição 2
            Termo [1-indexacao] na oraçao 3 na posição 3
                  Frequencia de [politica] no docto 19 = 17
                  Frequencia de [politica] no corpus = 31
                  Termo [0-politica] de peso = 2 coef =  0.028162
                  Frequencia de [indexacao] no docto 19 = 31
                  Frequencia de [indexacao] no corpus = 12
                  Termo [1-indexacao] de peso = 2 coef =  0.132338
      Termo [0-politica] alinhado na oracao 9
      Termo [1-indexacao] alinhado na oracao 9
            Termo [0-politica] na oraçao 9 na posição 5
            Termo [1-indexacao] na oraçao 9 na posição 6
                  Frequencia de [politica] no docto 19 = 17
                  Frequencia de [politica] no corpus = 31
                  Termo [0-politica] de peso = 2 coef =  0.153459
                  Frequencia de [indexacao] no docto 19 = 31
                  Frequencia de [indexacao] no corpus = 12
...               Termo [1-indexacao] de peso = 2 coef = 0.231591
```

ECHOCALC

```
Doc [1]
      Seq EM [0] freq em [10]
Doc [2]
      Seq EM [0] freq em [6]
      Seq EM [3] freq em [2]
...
Mostra calculo opcao 2 Tot Doctos [193]
Nro doc [1] Nome arq [Enancib2010/artigo/GT1/40.pdf] Tam doc [49605] Max
Freq [69] Posicao [0]
  Qtd termos[7330] Tam termos[29308] Qtd Lexico[3642] Tam Lexico[15113]
  [ciencia-informacao]
    FreqDoc[1] FreqCon[49] Doc big[166] IDF[6.544922e-002]
    pesoDoc[6.544922e-001] PesoCon[3.207012e+000]
    Num [2.098964e+000] DenDoc [4.283601e-001] DenCon [1.028492e+001]
  [informacao-cientifica]
    FreqDoc[0] FreqCon[31] Doc big[118] IDF[2.136753e-001]
    pesoDoc[0.000000e+000] PesoCon[6.623934e+000]
      Num [2.098964e+000] DenDoc [4.283601e-001] DenCon [5.416143e+001]
    [informatika-ciencia]
      FreqDoc[0] FreqCon[13] Doc big[0] IDF[0.000000e+000]
      pesoDoc[0.000000e+000] PesoCon[0.000000e+000]
      Num [2.098964e+000] DenDoc [4.283601e-001] DenCon [5.416143e+001]
    [estados-unidos]
      FreqDoc[0] FreqCon[12] Doc big[32] IDF[7.804073e-001]
      pesoDoc[0.000000e+000] PesoCon[9.364888e+000]
      Num [2.098964e+000] DenDoc [4.283601e-001] DenCon [1.418626e+002]
    [mikhailov-colaboradores]
      FreqDoc[0] FreqCon[11] Nro doc contem big[0] IDF[0.000000e+000]
      pesoDoc[0.000000e+000] PesoCon[0.000000e+000]
      Num [2.098964e+000] DenDoc [4.283601e-001] DenCon [1.418626e+002]
Total match [1]
```

# APÊNDICE D

## Configuração do componente de software Client

O componente de *software* denominado Client é o responsável por receber o documento informado para servir de referência no processo de busca. O processo realizado corresponde ao descrito nas etapas dois até cinco da segunda fase da metodologia. Esse componente é constituído de um programa executável cuja função principal é receber como parâmetro o documento de referência para transformar esse conteúdo em um conjunto de bigramas e submetê-lo a uma consulta termo a termo através de uma conexão via *socket* ao serviço de busca (Server). Esse programa tem os seus dados de configuração informados no arquivo (client.ini). Ele pode ser instalado separadamente em um computador que disponibiliza serviço de acesso a páginas web. Esse componente funciona como um Comum Gateway Interface (CGI) que responde a uma aplicação escrita na linguagem PHP. Ele tem o papel de requisitar uma consulta ao servidor de consulta (server) e gerar com saída uma página Hypertext Markup Language (html) de resposta para o browser requisitante.

Para esse *software* ser executado é necessário instalar em um diretório os seguintes arquivos:

- client.exe – programa executável gerado através de código fonte C++;
- libtet.dll – biblioteca *freeware* versão limitada, sobre licença de TET LIB;
- client.ini – arquivo texto usado para configurar as funcionalidades do *software*;
- index.html – página web da interface de consulta;
- consulta_docto.php – programa php que comunica com o CGI client.exe a fim de gerar a página de resposta para o requisitante;
- recebe_upload.php – programa php que é chamada pelo consulta_docto a fim de processar o upload do documento utilizado como referência da busca;
- StopWord.txt – arquivo com a lista de palavras que serão descartadas.

Durante o processamento, o *software* gera um arquivo (client.log) que contém várias informações do processamento intermediário que podem ser geradas ou não, conforme definido nos parâmetros contidos no arquivo client.ini. Esse arquivo é definido com a mesma estrutura do arquivo de configuração descritas no apêndice C. A seguir apresentam-se as variáveis a serem configuradas.

| Nro | Nome da Variável | Nome da Variável |
|---|---|---|
| 1 | NIVELLOG | Define como será a forma de gravação do arquivo (client.log). Ao informar: 0 (zero) salva todas as mensagens informativas de acompanhamento do processamento no arquivo de log; 1 (um) salva somente mensagens de erro e demais mensagens que forem explicitamente informadas através dos parâmetro. Como regra deve-se usar o valor igual a 1, pois reduz o tamanho do log gerado. |
| 2 | STOPWORD | Define o caminho e o nome do arquivo que contém as *stop words*. Ao colocar apenas o nome o arquivo será procurado no mesmo diretório em que a aplicação for executada. |
| 3 | IP | O endereço Internet Protocol (IP) ou a Universal Request Locator (URL) de onde o server está respondendo. |
| 4 | PORTA | Número da porta TCP/IP em que o *software* estabelecerá conexão via *socket* com o serviço busca por bigramas. O número da porta tem de ser o mesmo definido pelo servidor. O valor default é a porta 8000. |
| 5 | PATHCORPUS | Caminho absoluto, em relação à raiz do sistema de arquivos, onde o *corpus* de documentos está localizado. |
| 6 | NOMLISTA | Define o *path* e o nome do arquivo que contém a lista com os nomes dos documentos a processar. |
| 7 | MOSTRASW | Define se gravará no arquivo client.log as *stop words* que o processamento atual utilizará. Os valores informados podem ser S ou N. |
| 8 | MOSTRATEXTO | Define se gravará no arquivo client.log uma versão do documento utilizado como referência da busca em formato texto, após extrair os cabeçalhos e referências, identificar e expandir as siglas e retirar a acentuação das palavras. Os valores informados podem ser S ou N. |
| 9 | MOSTRAORACAO | Define se gravará no arquivo client.log o texto com o mesmo conteúdo do processamento anterior, só que identificando e numerando as orações. Os valores informados podem ser S ou N. |
| 10 | MOSTRATERMO | Define se gravará no arquivo client.log o texto com o mesmo conteúdo do processamento anterior, só que identificando e numerando os termos. Os valores informados podem ser S ou N. |
| 11 | CONVERTEPDFTXT | Define se gravará um arquivo com o mesmo nome do documento utilizado como referência da busca e no mesmo diretório trocando a extensão de (.pdf) para (.txt). O arquivo txt gerado é a versão final utilizada na indexação dos termos, retirando as *stop words* as pontuações e transformando todos os termos em minúsculas. Os valores informados podem ser S ou N. |
| 12 | ECHOMSG | Define se gravará no arquivo client.log as requisições de consulta enviadas a aplicação Server e as respostas retornadas. Os valores informados podem ser S ou N. |

| | | |
|---|---|---|
| 13 | MOSTRAESTRUT | Define se gravará no arquivo client.log a estrutura utilizada para a extração dos bigramas. Os valores informados podem ser S ou N. |
| 14 | QTDOCORRENCIA | Define qual será a quantidade de ocorrências que um bigrama deverá ocorrer no documento para que ele seja considerado como relevante para a consulta. Valor informado deve ser maior que um e menor ou igual a nove. O valor utilizado nesta tese foi igual a 4. |
| 15 | MOSTRABIGRAMA | Define se gravará no arquivo client.log os bigramas extraídos para processar a consulta. Os valores informados podem ser S ou N. |
| 16 | GRAVAARQEM | Essa é uma opção bem específica. Os valores informados podem ser S ou N. Caso seja informado S a funcionalidade do *software* fica sendo especificamente para ser executado através de chamada externa do programa GeraEM.exe, sem estabelecer conexão com o server, com objetivo de atender aos requisitos da quarta fase da metodologia. Caso seja informado N o *software* realiza normalmente a conexão remota com o server. |

A seguir são apresentadas as informações geradas no arquivo client.log, mas somente aquelas que são específicas do processamento desse programa, portanto, ainda não foram apresentadas no apêndice C:

MOSTRAESTRUT

```
Lendo Palavras
1 - (informacao) Frequência 12
      Oracao [1] posicao na oracao [2]
      Oracao [2] posicao na oracao [7]
      Oracao [6] posicao na oracao [3]
      Oracao [14] posicao na oracao [2]
      Oracao [14] posicao na oracao [13]
      Oracao [14] posicao na oracao [17]
      Oracao [19] posicao na oracao [4]
      Oracao [20] posicao na oracao [6]
      Oracao [21] posicao na oracao [7]
      Oracao [23] posicao na oracao [3]
      Oracao [23] posicao na oracao [4]
      Oracao [32] posicao na oracao [11]
2 - (tecnologia) Frequência 7
      Oracao [1] posicao na oracao [3]
      Oracao [44] posicao na oracao [2]
      Oracao [53] posicao na oracao [9]
      Oracao [54] posicao na oracao [7]
      Oracao [59] posicao na oracao [6]
      Oracao [86] posicao na oracao [8]
      Oracao [187] posicao na oracao [19]
3 - (modalidade) Frequência 1
      Oracao [1] posicao na oracao [4]
```

MOSTRABIGRAMA

```
Bigramas utilizados como termos de busca
5 uso busca
5 busca avancada
5 busca compartilhamento
5 busca videos
7 compartilhamento informacoes
4 compartilhamento videos
15 web 20
5 vaz 2008
4 determinado video
4 alem disso
5 postar videos
6 of the
Fim dos Bigramas
```

# APÊNDICE E

## Configuração do componente de software AtuMySql

O componente de *software* denominado AtuMySql é o responsável por carregar no banco de dados MySql os arquivos com as EM extraídas pelas 14 técnicas utilizadas nesta pesquisa inserindo esses conteúdos na tabela “docmetrica”. Ele pode ser instalado em um computador que tenha acesso ao *corpus.* O arquivo de configuração (atumysql.ini) permite definir os seguintes parâmetros para o processamento do programa.

Para esse *software* ser executado é necessário instalar em um diretório dos seguintes arquivos:

- atumysql.exe – programa executável gerado através de código fonte C++;
- libmysql.dll – biblioteca de acesso ao SGBD MySql;
- atumysql.ini - arquivo texto usado para configurar as funcionalidades do *software*;
- lista.txt – arquivo texto com a relação do caminho e nome dos documentos a processar.

O *software* durante o seu processamento gera um arquivo (atumysql.log) com informações do processamento intermediário que podem ser geradas ou não, conforme definido no arquivo atumysql.ini. Esse arquivo de configuração é definido com a mesma estrutura do arquivo de configuração descritas no apêndice anterior.

| Nro | Nome da Variável | Nome da Variável |
|---|---|---|
| 1 | NIVELLOG | Define como será a forma de gravação do arquivo (atumysql.log). Ao informar: 0 (zero) salva todas as mensagens informativas do processamento no arquivo atuMySql.log; 1 (um) salva somente mensagens de erro. |
| 2 | PATHCORPUS | Caminho absoluto, em relação à raiz do sistema de arquivos, onde o *corpus* de documento está localizado. |
| 3 | NOMLISTA | Define o *path* e o nome do arquivo que contém a lista com os nomes dos documentos a processar. |

## Configuração do componente de software Rank

O componente de *software* denominado Rank é o responsável por ler os arquivos gerados pela comparação das 13 técnicas produzidos pelo pacote NSP duas a duas,

extraindo a informação do coeficiente de correlação de Spearman. Ele pode ser instalado em um computador que tenha acesso ao *corpus.* Para esse *software* ser executado é necessário instalar em um diretório os seguintes arquivos:

- rank.exe – programa executável gerado através de código fonte C++;
- rank.ini - arquivo texto usado para configurar as funcionalidades do *software*;
- lista.txt – arquivo texto com a relação do caminho e nome dos documentos a processar.

O *software*, durante o seu processamento gera um arquivo (rank.log) com informações do processamento intermediário que podem ser geradas ou não conforme definido no arquivo rank.ini. Esse arquivo de configuração contém apenas as variáveis NIVELLOG, NOMLISTA e PATHCORPUS definidas com a mesma estrutura do arquivo de configuração já descritas anteriormente.

### Componente de software GeraEM

O componente de *software* denominado GeraEM é o responsável por gerar um arquivo com a extensão (.em) para cada um dos arquivo definidos no arquivo lista.txt. Para cada linha lida do arquivo lista.txt, correspondente a cada documento a ser extraído as EM, esse programa realiza uma chamada de execução externa no programa Client.exe. Ele foi criado apenas para automatizar a execução do programa Client para uma lista pré-definida de documentos. Durante esse processamento, não ocorre a comunicação do Client com o Server.

### Componente de software ConsEM

O componente de *software* denominado ConsEM é o responsável por gerar uma consulta para cada um dos arquivos definidos no arquivo lista.txt. Esse programa realiza uma chamada de execução externa no programa Client.exe. Ele foi criado apenas para automatizar a execução do programa Client para uma lista pré-definida de documentos. Durante o seu processamento ocorre a comunicação do Client com o Server. Durante o processamento desse programa ele grava um arquivo de log permanente denominado arqEM.log que apresenta o total de EM extraída para cada documento.

### Componente de software Monitor

O componente de *software* denominado Monitor é o responsável por monitorar o funcionamento do server. Os serviços *online* disponibilizados são: (1) inicializar o serviço de indexação de documentos; (2) atualizar os parâmetros de configuração; (3) processar a indexação incremental de novos documentos aos já indexados em memória; (4) esvaziar o conteúdo do arquivo de log; (5) visualizar os erros ocorridos; (6) desativar o server finalizando o serviço.

# APÊNDICE F

Documento: 172.pdf

**Uma abordagem baseada em métricas de redes complexas para o estabelecimento do grau de influência de termos em documentos**

Resumo

Nos últimos anos, a área de recuperação de informação tem recebido atenção especial da comunidade científica mundial. Pesquisas relacionadas à melhoria de métodos e algoritmos para recuperação de informação textual tem se ampliado, concentradas em grande parte, no aprimoramento modelo vetorial, em especial na busca de métodos e funções mais eficientes para o cálculo de similaridade entre documentos e consultas. Paralelamente a análise de redes complexas tem despertado o interesse da comunidade científica devido a sua capacidade de representação de problemas complexos de maneira objetiva, oferecendo um arcabouço teórico e prático para o estudo das propriedades e comportamentos dos elementos e relações que compõem os problemas. Recentemente, pesquisas considerando os documentos como redes complexas de palavras vem sendo desenvolvidas. Entretanto, as possibilidades de utilização desta abordagem na resolução de problemas de recuperação e classificação de informação ainda foram pouco exploradas. O presente artigo apresenta uma abordagem baseada em métricas de redes complexas para obtenção de uma função de atribuição de pesos a termos em documentos. A presente abordagem apresentou precisão equivalente ao modelo vetorial quando aplicada para a estimativa de similaridade entre documentos e consultas a partir de uma coleção de referência, o que evidencia a aplicabilidade de métricas de redes complexas de palavras em problemas recuperação da informação.

Documento: 86.pdf

**Repositório digital da UNATI-UNESP: o olhar da arquitetura da informação para a inclusão digital e social de idosos**

Resumo

Os usuários idosos podem utilizar web desenvolver diversas atividades cotidianas. No entanto, verificamos muitos ambientes informacionais digitais não possuem uma Arquitetura da Informação desenvolvida com foco nas necessidades específicas desse público, dificultando sua usabilidade e acessibilidade e, consequentemente dificultando inclusão digital e social desse grupo de usuários. Nesse contexto, objetivamos identificar elementos que viabilizem a inclusão digital e social dos idosos a partir dos estudos de Arquitetura da Informação, Usabilidade, Acessibilidade e Comportamento Informacional, no contexto da Ciência da Informação bem como a aplicação desses elementos em um repositório digital *DSpace* construído para Universidade Aberta da Terceira Idade UNATI – UNESP. Consideramos que um repositório digital que abarque assuntos de interesse e produções de idosos e que apresente elementos inclusivos viabiliza inclusão digital e social. Para a aplicação da pesquisa utilizamos pesquisa-ação, que objetivou a construção participativa do repositório digital UNATI – UNESP, junto os alunos da UNATI – Marília-SP, que se efetivou por meio de grupos focais e com respaldo em estudos que revelaram as necessidades da instituição e dos alunos contribuindo para a identificação de elementos que favorecem inclusão digital e social de usuários idosos.

Documento (384.pdf)

**Representação interativa e folkosonomia assistida para repositórios digitais**

Resumo

A recuperação da informação tem sido muito discutida dentro da Ciência da Informação ultimamente. A busca por informação de qualidade e compatível com a necessidade do usuário tornou-se objeto constante de pesquisa. A utilização da Internet como fonte de disseminação do conhecimento indicou novos modelos de armazenamento de informações, como os repositórios digitais, que têm sido utilizados em ambientes acadêmicos e de pesquisa como principal forma de autoarquivar e disseminar informação, porém com uma estrutura de informação que comporta melhor descrição dos recursos e consequentemente uma melhor recuperação da informação. Desta forma o objetivo deste trabalho é melhorar processo recuperação da informação, apresentando uma proposta de modelo estrutural no contexto da web semântica, abordando o uso de recursos Web 2.0 e Web 3.0 em repositórios digitais, que permita a recuperação semântica da informação, por meio da construção de uma camada de informação chamada Representação Iterativa. O presente estudo caracteriza-se como uma pesquisa descritiva e analítica, com base em análise documental, dividida duas partes: a primeira, caracterizada observação direta não participativa de ferramentas que implementam repositórios digitais, assim como repositórios digitais já instanciados, e a segunda, com característica exploratória, onde sugere um modelo inovador para repositórios, com a utilização de estruturas de representação do conhecimento e participação do usuário na construção de um vocabulário próprio de domínio. Através de um modelo sugerido e proposto - Representação Iterativa - será possível adequar os repositórios digitais utilizem Folksonomia e também vocabulário controlado de domínio de forma a gerar uma camada de informação iterativa, que possibilite retroalimentação da informação, além de recuperação semântica da informação, através de um modelo estrutural desenhado para repositórios. O modelo sugerido resultou na efetivação da tese de que por meio da Representação Iterativa é possível estabelecer um processo de recuperação semântica da informação em repositórios digitais.

Documento (186.pdf)

**Contribuição dos repositórios institucionais à comunicação científica: um estudo da Universidade Federal do Rio Grande do Sul.**

Resumo

Relato do estudo que investigou o uso das teses e dissertações depositadas no Lume - Repositório Digital da Universidade Federal do Rio Grande do Sul, buscando saber quem o usa, quais os documentos mais utilizados, seus respectivos orientadores e programas de pós-graduação. Tem o objetivo de levantar questionamentos e possíveis respostas que evidenciem a importância dos repositórios institucionais para a comunicação da literatura científica em acesso aberto numa instituição de ensino superior. A coleta de dados foi realizada em duas etapasa A primeira delas foi realizada meio da análise estatística dos *downloads* ocorridos no período de 1º de março a 31 maio de 2009, o que permitiu obter informações sobre o uso das teses e dissertações, orientando a seleção dos sujeitos para a etapa seguinte. A segunda etapa, qualitativa, foi desenvolvida mediante entrevistas realizadas com os professores que obtiveram o maior índice de *downloads* por documento. Os entrevistados manifestaram-se sobre os dados coletados opinando sobre o seu significado, importância e possíveis usos. Mais amplamente, os resultados obtidos nas duas etapas apontam para a inegável importância dos repositórios institucionais no processo de comunicação da produção científica da instituição de ensino superior.