UNIVERSIDADE ESTADUAL DO CEARÁ

CENTRO DE CIÊNCIAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

FABIANO TAVARES DA SILVA

LUPPAR: UM SISTEMA DE RECUPERAÇÃO DE INFORMAÇÃO PARA COLEÇÕES FECHADAS DE DOCUMENTOS

FORTALEZA – CEARÁ

2018

FABIANO TAVARES DA SILVA

LUPPAR: UM SISTEMA DE RECUPERAÇÃO DE INFORMAÇÃO PARA COLEÇÕES FECHADAS DE DOCUMENTOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. José Everardo Bessa Maia

FORTALEZA – CEARÁ

2018

Dados Internacionais de Catalogação na Publicação

Universidade Estadual do Ceará

Sistema de Bibliotecas

Silva, Fabiano Tavares da.
Luppar: um sistema de recuperação de informação para coleções fechadas de documentos [recurso eletrônico] / Fabiano Tavares da Silva. - 2018.
1 CD-ROM: il.; 4 ¾ pol.

CD-ROM contendo o arquivo no formato PDF do trabalho acadêmico com 75 folhas, acondicionado em caixa de DVD Slim (19 x 14 cm x 7 mm).

Dissertação (mestrado acadêmico) - Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Mestrado Acadêmico em Ciência da Computação, Fortaleza, 2018.
Área de concentração: Ciência da Computação.
Orientação: Prof. Dr. José Everardo Bessa Maia.

1. Recuperação de informação. 2. Análise de Contexto Local. 3. Semântica Distribucional. 4. Expansão de Consultas. 5. Indexação. I. Título.

FABIANO TAVARES DA SILVA

LUPPAR: UM SISTEMA DE RECUPERAÇÃO DE INFORMAÇÃO PARA COLEÇÕES FECHADAS DE DOCUMENTOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em: 22 de Agosto de 2018

BANCA EXAMINADORA

Prof. Dr. José Everardo Bessa Maia (Orientador)
Universidade Estadual do Ceará UECE

Prof. Dr. Ângelo Roncalli Alencar Brayner
Universidade Federal do Ceará - UFC

Prof. PhD. Paulo Henrique Mendes Maia
Universidade Estadual do Ceará UECE

À minha amada esposa Valkiria, por todo amor, incentivo, apoio e compreensão. Sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.
À minha mãe, Iranir, por sempre acreditar e ter abdicado de sua vida em prol das realizações e da felicidade de seu filho.
À minha madrinha, Silvânia (*In memoriam*).

AGRADECIMENTOS

Agradeço primeiramente a Deus por tudo, desde e sempre por ter me amparado em todos os momentos da minha vida, me ajudando, orientando e incentivando a seguir. Obrigado Deus por esta dádiva chamada vida.

Agradeço ao meu orientador Prof. Dr. José Everardo Bessa Maia, para quem não há agradecimentos que cheguem. Sou grato pelas aulas, orientação, incentivo, elaboração, pela paciência, atenção e dedicação oferecidas antes e durante a construcão deste trabalho. Muito obrigado, pelo amparo em todos os momentos.

Agradeço a minha esposa Valkiria por seu cuidado e dedicação foi que deram a força para seguir em frente. Por ter aceitado se privar de minha companhia pelos estudos, concedendo a mim a oportunidade de me realizar ainda mais. Obrigado pelo constante amor e incentivo. À minha mãe Iranir e ao meu pai Pereira (*in memoriam*) deixo um agradecimento especial, por todas as lições de amor, companheirismo, amizade, caridade, dedicação, abnegação, compreensão e perdão que foi me dado ao longo dos anos. A minha tia Inácia, que tanto contribuiu com a minha formação, se fazendo presente em todos os momentos e vibrando pelo meu sucesso. Tia você é o referencial de crescimento pessoal e educacional. À minha família, tios(as), primos(as), cunhados(as) e ao meu sogro, por apoiarem e compreenderem o meu isolamento em inúmeros fins de semanas.

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional. Não poderia esquecer do Prof. Dr. Gilvan Maia pela orientação acadêmica e de vida. Ao pessoal da secretaria tanto da graduação quando da pós.

Agradeço também aos meus amigos dos seguintes grupos que me apoiaram e incentivaram dia-a-dia: da Secretária das Cidades; da Controladoria Geral do Estado (CGE); meus amigos da graduação (UECE); meus amigos de infância; e especial meus amigos do mestrado.

A todos obrigado por permitirem que esta dissertação seja uma realidade.

“A educação é a arma mais poderosa que você pode usar para mudar o mundo.”

(Nelson Mandela)

RESUMO

Esta dissertação descreve um sistema de Recuperação de Informação (RI) para coleções de documentos aproximadamente uniformes em tamanho e formato. Exemplos de tais coleções são anais de conferências ou revistas científicas, prontuários médicos, sinopses de notícias, entre outros. Todas as coleções possuem em comum o mesmo domínio. A questão chave em RI é trazer a necessidade de informação do usuário em documentos relevantes. A consulta por ele projetada pode não carregar toda sua intenção do que realmente deseja. Na literatura o uso de diversas técnicas de realimentação implícita de consultas ataca diretamente esse problema. Para este trabalho analisaram-se as técnicas existentes desde o uso de tesauro até propor uma abordagem que usa a teoria de Semântica Distribucional para construir uma Análise de Contexto Local. Utilizou-se quatro coleções com distintos aspectos de domínio, sendo uma das coleções originalmente construída em português e disponibilizada para trabalhos futuros. O trabalho propõe avaliar o sistema de RI como um todo: com a expansão de consulta, a recuperação e o ranqueamento com algoritmos de *baseline* equivalentes aos métodos clássicos. Como forma de alcançar êxito nas tarefas que compõem o estado da arte de RI foi desenvolvido um motor de busca com interface Web que permite também o uso de técnicas de *feedback* de relevância. A abordagem é avaliada nas quatro bases de dados em inglês e português e comparada com técnicas semelhantes. Os resultados dos experimentos validaram a abordagem e mostram-se com uma performance competitiva e qualificada para as soluções geradas.

Palavras-chave: Recuperação de informação. Análise de Contexto Local. Semântica Distribucional. Indexação. Expansão de Consultas.

ABSTRACT

This dissertation describes an Information Retrieval (IR) system for collections of documents approximately uniform in size and format. Examples of such collections are annals of conferences or scientific journals, medical records, news synopses, among others. All collections have the same domain in common. The key issue in IR is to bring the need for user information into relevant documents. The query designed may not carry all intention of what really wants. In the literature, the use of several implicit query feedback techniques directly addresses this problem. For this work we will analyze the existing techniques from the use of thesaurus to propose an approach that uses the theory of Distributional Semantics to build a Local Context Analysis. Used four collections with different aspects of domain, one of the collections originally built in Portuguese and made available for future work. The work proposes to evaluate the IR system as a whole: with the query expansion, retrieval and ranking with baseline algorithms equivalent to classical methods. As a way of achieving success in the tasks that make up the state of the art of IR, a search engine with Web interface was developed which also allows the use of relevance feedback techniques. Our approach is evaluated in four databases in English and Portuguese and compared with similar techniques.The results of the experiments were validated and shown with competitive and qualified performance for the solutions generated.

Keywords: Information retrieval. Local Context Analysis. Distributional Semantics. Indexing. Query Automatic Expansion.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

ACL	Análise de Contexto Local
ASCII	American Standard Code for Information Interchange
BM25	Okapi BM25 - Best Matching 25
CBA	Congresso Brasileiro de Automática
FR	*Feedback* de Relevância
LSI	*Latent Semantic Indexing*
MSD	Modelos de Semântica Distribucional
NLTK	*Natural Language Toolkit*
PDF	*Portable Document Format*
RI	Recuperação de Informação
SRI	Sistema de Recuperação de Informação
SVD	*Single Value Decomposition*
TF-IDF	*Term Frequency–Inverse Document Frequency*
TREC	Text REtrieval Conference
VSM	*Vector Space Model*

SUMÁRIO

1 INTRODUÇÃO

Estamos na era da publicação eletrônica e com o avanço da tecnologia e seu uso onipresente, o acúmulo e processamento de dados acontecem em alta velocidade. Tecnologias como internet, dispositivos de memória secundária de grande capacidade e baixo custo e gerenciadores de banco de dados são exemplos de recursos que viabilizam a aquisição massiva de novas bases de dados (LIU; MOTODA, 2007; BAEZA-YATES; RIBEIRO-NETO, 2013). Porém, a análise de grandes quantidades de dados pelo homem é inviável sem auxílio de ferramentas computacionalmente apropriadas (GOLDSCHMIDT; PASSOS, 2015). Criar ferramentas e algoritmos são desafios para reduzir tempo e a precisão no acesso à informação.

Assim como armazenam com facilidade os dados, os usuários também necessitam acessá-los de forma fácil, ou seja, semelhante ao que ocorre no dia-a-dia através da linguagem natural. Para buscar eficientemente a *informação* nessa grande quantidade de dados, um usuário utiliza um sistema de Recuperação de Informação (RI).

A recuperação digital de informação é um problema de interesse de corporações geradoras de informação e conhecimento, registrados na forma de objetos digitais. Exemplos de tais corporações são as empresas de comunicação, em geral, tais como jornais, TVs, Rádios e Sites da Internet, mas também um banco de investimento, uma universidade, um centro de pesquisa, um departamento de estatística governamental, uma agência de fomento e uma empresa de consultoria são outros exemplos de corporações geradoras de conteúdo.

Atualmente as pessoas utilizam RI todos os dias. Por exemplo, na internet com os motores de buscas (Google, Bing e Yahoo), clientes de e-mails, bibliotecas digitais ou no computador pessoal com uso de RI local (Mac Spotlight e Windows Search) ou ainda em ambientes corporativos como em hospitais, médicos buscando exames, ou em tribunais com a busca de processos e em sistemas embarcados.

O problema de um sistema de recuperação de informação pode ser dividido em duas tarefas básicas: *Indexação* e *Recuperação*. Adiante descreve-se em detalhes essas duas etapas. Todavia, de maneira geral, a primeira tarefa diz respeito a como representar os documentos para que sejam computacionalmente eficientes para a segunda tarefa, a *recuperação*, que por sua vez trata dos algoritmos que buscam localizar a informação.

1.1 MOTIVAÇÃO

Neste trabalho refere-se ao *documento* como a unidade de texto indexada em um sistema de RI que está disponível para recuperação (SINGHAL, 2001). Um documento poderia ser um jornal, um parágrafo de frases, entradas de enciclopédias ou uma página da web. Os documentos podem ser classificados em até três categorias: estruturado, não estruturado e semiestruturados. Os dados estruturados são organizados e armazenados em campos, tags e metadados, como exemplo os bancos de dados, o que tornam facilmente utilizáveis por um computador. Nos semiestruturados apenas parte do conteúdo possui estrutura, a exemplo, páginas da web, enquanto os não-estruturado são semanticamente livres de estruturas, o que os tornam um desafio para RI. Exemplos de dados não-estruturado são áudio, vídeo e texto, como o corpo de uma mensagem de e-mail, página web ou processador de texto. Já um *corpus* ou coleção é um conjunto de documentos disponíveis para recuperação. Quando se refere a *termo* atribui-se a palavra ou item lexical que ocorre no documento. Finalmente, uma *consulta* representa a necessidade de informação de um usuário expressa como um conjunto de termos (SINGHAL, 2001).

O processo de RI nasce da necessidade do usuário de obter uma informação. Para isso transcreve-se em linguagem natural uma consulta para o RI que formalmente será traduzida na necessidade de informação. Com a consulta, o sistema de RI tem por objetivo localizar uma coleção de documentos que representam a intenção de busca do usuário. Os documentos são escolhidos entre os mais relevantes para a consulta e com o mínimo de itens irrelevantes. Um documento é relevante se é aquele que o usuário percebe como contendo as informações de valor em relação aos seus interesses pessoais (SCHÜTZE; MANNING; RAGHAVAN, 2008). Dessa forma tem-se como desafios fundamentais para RI a interpretação da consulta e o critério de relevância dos documentos. A resposta do sistema envolve uma subjetividade por depender do julgamento do usuário o que torna o problema não trivial.

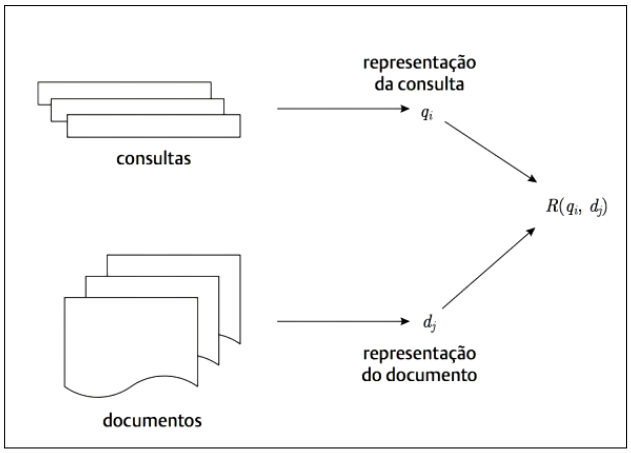
Em busca de ser eficaz em trazer a necessidade de informação do usuário, resolver o problema RI é encontrar uma forma de interpretar o conteúdo dos documentos de uma coleção e classificá-los de acordo com o grau de relevância à consulta do usuário. Essa interpretação do conteúdo de um documento envolve a extração de informações sintáticas e semânticas do texto do documento e sua utilização para satisfazer a necessidade de informação do usuário.

A literatura apresenta três modelos clássicos que sintetizam o raciocínio do parágrafo anterior, que são chamados de Booleano, Vetorial e Probabilístico. Todos os modelos seguem

a representação da Figura 1 e caracterizam-se da seguinte forma (BAEZA-YATES; RIBEIRO-NETO, 2013):

1. Um modelo de RI é um quádrupla $[D, Q, F, R(q_i, d_j)]$
2. D uma representação lógica dos documentos.
3. Q uma representação da necessidade de informação do usuário (consulta).
4. F é um arcabouço para a modelagem dos documentos, consultas e suas relações.
5. $R(q_i, d_j)$ uma função que associa um número real com uma consulta $q_i \in Q$ e uma representação de documento $d_j \in D$. Esta função define uma ordenação entre os documentos com respeito a consulta q_i.

Figura 1 – Modelo de RI clássico



Fonte**:** (BAEZA-YATES; RIBEIRO-NETO, 2013)

No modelo Booleano F é modelado com operações lógicas sobre os conjuntos de Q e D, enquanto que no Vetorial F é formado por operações de álgebra linear com a representação de Q e D por vetores. Já no modelo probabilístico F é composto pelas distribuições de probabilidade onde Q e D são modelados usando teorema de *Bayes*. Ao longo dos anos outros modelos foram propostos, porém, todos tomando como base estes clássicos.

Para estes modelos clássicos e as futuras abordagens, duas variáveis, consultas (Q) e documentos (D), dessa modelagem estão em constante mudança, o que levar a crer que a busca pelo aperfeiçoamento desses modelos ainda é um problema atual. Os documentos D continuam crescendo. Um exemplo disso é a internet que já chega a 4 bilhões de páginas [1] e gera novos desafios de indexação e representação dos documentos. Sendo assim é necessário combinar soluções que sejam rápidas e seletivas. As consultas (Q) e o conceito de relevância

[1] WorldWideWebSize.com | The size of the World Wide Web (The Internet), 2017, http://www.worldwidewebsize.com

evoluem e são realizadas em diferentes contextos e domínios, exigindo soluções que considerem novos parâmetros ao modelo. Apesar de existirem diversos modelos e derivações de técnicas de recuperação de informação, não existe uma solução ideal. É necessário atualizar esses modelos e manter o crescimento da pesquisa em RI. Um fato que descreve bem essa continuidade das pesquisa em RI é a TREC[2], uma conferência anual que exigem cada vez mais da comunidade científica por soluções ao problema de RI.

O processo de recuperação nem sempre é eficaz, sendo que a ineficácia muitas vezes é causada pelo uso impreciso de palavras-chave na consulta. Na prática, o usuário adiciona poucas palavras e com ambiguidade (CARPINETO; ROMANO, 2012). Isso faz com que o usuário necessite refinar sua consulta com várias interações adicionando novas palavras ou então desiste da busca. Uma abordagem comumente utilizada para resolver este problema é através da expansão de consulta (XU; CROFT, 1996) onde se adiciona novas palavras ou frases aos termos da consulta para que o Sistema de Recuperação de Informação (SRI) recupere novos documentos e aproxime-se da necessidade do usuário. Os dois problemas comumente encontrados na consulta inicial são de especificidade ou de abrangência de significado dos termos utilizados.

Adicionar novos termos a uma consulta pode ser feita de forma manual pelo usuário, semiautomático ou automático. Na forma manual o usuário julga o que poderia melhorar sua consulta e reformula os termos da consulta. Na forma semiautomática o sistema assiste o usuário, por exemplo, sugerindo a adição de novos termos. Na expansão automática não há participação do usuário sendo a adição de novos termos realizada de forma implícita. O sistema desenvolvido neste trabalho realiza expansão automática de consulta.

[2] Text REtrieval Conference (Text REtrieval Conference (TREC)) Overview, 2017, http://trec.nist.gov/overview.html

1.2 HIPÓTESE

Os SRI mais populares respondem a milhões de consultas. Em termos gerais, os usuários expressam sua necessidade inserindo sequências curtas de termos que em média são 2 – 3 palavras-chave (CROFT; METZLER; STROHMAN, 2011). Os motores de busca e os mecanismos de recuperação de informações, em geral, são desafiados a entender adequadamente esses textos curtos, a fim de produzir melhores resultados e melhorar a experiência do usuário.

A consulta é o fator chave entre o usuário e os resultados que lhe interessam. Realimentar a consulta é a principal estratégia para chegar a real intenção do usuário. A literatura já demonstrou que há ganhos de 10% e até mais na eficácia dos resultados em RI (LIU *et al.*, 2004; LEE; CROFT; ALLAN, 2008) ao expandir a consulta. A TREC de 2009 deu popularidade a essas técnicas graças aos resultados de avaliação obtidos, onde a maioria dos participantes fez uso e relatou melhorias visíveis no desempenho de recuperação.

A realimentação de consulta pode ser implícita (o sistema se encarrega de automaticamente reformular a busca) ou explícita (usuário participa da reformulação da consulta). Essas duas ações exigem da RI perceber a real intenção do usuário, que está implícita na linguagem natural utilizada na consulta. Reformular a consulta, seja com o apoio do usuário (*Feedback* de Relevância) ou não, envolve estender a consulta para agregar relevância ao significado (semântica). Se para expansão fizer uso dos resultados iniciais da consulta e combinar com tesauros, será Análise de Contexto Local. A partir desse ponto e com a teoria, descrita no capítulo 2 e 3, dos Modelos Semânticos Distribucional, que vem ganhando espaço considerável nos últimos anos com o uso do aprendizado de máquina na construção de tesauros e seus bons resultados. A realização deste trabalho está baseada na seguinte hipótese:

- Modelos de Semântica Distribucional (MSD) fornecem suporte para melhoria de Expansão Automática de Consulta via Análise de Contexto Local (ACL) e *Feedback* de Relevância (FR) em coleções fechadas de documentos.

Entende-se aqui uma coleção fechada de documentos como sendo um repositório eletrônico local de documentos, estável no horizonte de busca e processamento. Isso está em contraste com recuperação de informação na Web, em larga escala e de múltiplos proprietários e em constante evolução. Exemplos de coleções fechadas são uma biblioteca digital ou um repositório de informações clínicas de pacientes.

1.3 OBJETIVOS

Objetivo Geral

Construir um Sistema de Recuperação de Informação (SRI) dotado de Análise de Contexto Local (ACL) baseada em Modelo Semântico Distribucional (MSD) (ACL-MSD), para coleções fechadas de documentos e avaliar o seu desempenho.

Objetivos Específicos

a) Construir uma interface de usuário de SRI para teste e demonstração dos sistemas. A interface deve incluir suporte para *feedback* de relevância explícita.
b) Preprocessar quatro coleções de documentos para serem utilizadas nos testes de desempenho, com conjuntos de consultas e *ground truth* disponíveis, sendo três delas bases públicas internacionais e uma base proprietária em lingua portuguesa.
c) Programar três versões de SRI *baselines*, utilizando os modelos *Booleano*, *Vector Space Model* (VSM) e Probabilístico (Okapi BM25 - Best Matching 25 (BM25)), avaliar e demonstrar o desempenho utilizando a interface de usuário construída. Os SRIs *baselines* possuem a possibilidade de expansão de consultas baseada em tesauros.
d) Implementar e avaliar o SRI com modelo probabilístico e vetorial nos seguintes cenários: com expansão de consultas com Análise de Contexto Local (ACL), com uso de contexto global por tesauro (*WordNet*) e o sistema em *baseline*.
e) Implementar e avaliar expansão de consultas em ACL baseada em Modelo Semântico Distribucional (ACL-MSD) sobre os modelos de SRIs e comparar os ganhos de desempenho proporcionados.
f) Implementar e avaliar o processamento de *feedback* de relevância baseada em ACL-MSD sobre os modelos de SRIs.

1.4 ESTRUTURA DA DISSERTAÇÃO

Após essa introdução, este trabalho continua com a seguinte estrutura: o capítulo 2 com fundamentação teórica necessária para RI e o entendimento das tarefas desenvolvidas; o capítulo 3 com estudo dos trabalhos relacionados a realimentação de consultas, análise de contexto e modelos semânticos distribucionais; o capítulo 4 com o desenvolvimento de fato

do trabalho com os desafios, metodologia aplicada, descrição de algoritmos e ferramentas que compõem a abordagem proposta. No Capítulo 5 apresenta-se a configuração dos testes, execução e validação do SRI; e finalmente, o capítulo 6 com o resumo do trabalho, discussão sobre os resultados, conclusão e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Em 1945, Vannevar Bush no artigo "*As We May Think*" lançou a ideia de usar computadores para pesquisar informações relevantes. Dez anos depois, os primeiros sistemas operacionais introduziram rotinas para recuperação de informação e já com o conceito de palavras indexando documentos. Nos anos 70 e 80 modelos e técnicas para RI começaram emergir. Porém, a ausência de bases textuais maiores e consolidadas deixaram esses modelos sem respostas. Então em 1992 com o início da Conferência de Recuperação de Texto, ou TREC, este disponibilizou grandes coleções de texto e então muitas técnicas antigas foram modificadas e muitas novas técnicas foram desenvolvidas (e ainda estão sendo desenvolvidas) para fazer uma recuperação efetiva em grandes coleções. Os algoritmos desenvolvidos em RI foram os primeiros a serem empregados para pesquisar a *World Wide Web* de 1996 a 1998, ano este que surgiram os primeiros buscadores Web. Desde então com crescimento da tecnologia, que avança para dentro das corporações e do dia-a-dia das pessoas, as aplicações em RI vem se tornando uma constante com diversos desafios (SINGHAL, 2001; SANDERSON; CROFT, 2012).

2.1 RECUPERAÇÃO DE INFORMAÇÃO

A Recuperação de Informação (RI) tem por objetivo localizar dentro de uma coleção, em geral, em computadores, documentos de uma natureza não estruturada (geralmente texto) que satisfaz uma necessidade de informação. Denomina-se de *documento* à unidade de texto indexada em um sistema de RI que está disponível para recuperação. A coleção de documentos chama-se de *corpus*. Neste trabalho foca-se em documentos textuais que são formados por palavras (item lexical), carregado de semântica e sintaxe, do qual se denomina de *termo*. O conjunto de termos que representam uma coleção forma um *vocabulário*.

A tarefa mais comum em RI é quando usuário precisa de informação e usa como entrada uma consulta formada por termos em uma quantidade que representa a sua necessidade, e então o sistema retorna os documentos relevantes. Esse tipo de tarefa em RI é chamada de recuperação *ad hoc*. O sistema de RI então orquestra duas tarefas principais: a *indexação* (representação dos documentos e das consultas) e a *recuperação* (aplicação de um modelo de recuperação). Na indexação são criadas estruturas de dados para representarem os documentos, a exemplo são índices e matrizes que trazem a relação entre os termos do vocabulário e os documentos (Índices invertidos e matrizes de ponderação). Objetivo da indexação é a eficiência da busca. Já na recuperação são executados os modelos de RI, a exemplo o booleano, vetorial e probabilístico,

que possuem como entrada a consulta e a configuração de parâmetros se necessário. O objetivo da recuperação é a eficácia. Os modelos localizam e selecionam os documentos seguindo um critério de relevância para a consulta. Todos os modelos clássicos utilizam o conceito de *relevância*. Um documento é relevante se este para o usuário contém as informações de valor em relação aos seus interesses pessoais. Outras tarefas estão embutidas e fazem total diferença no resultado. Algumas delas são o pré-processamento (limpeza, tokenização, radicalização e etc), realimentação de relevância, expansão de consultas e ranqueamento (SINGHAL, 2001; SCHÜTZE; MANNING; RAGHAVAN, 2008; BAEZA-YATES; RIBEIRO-NETO, 2013).

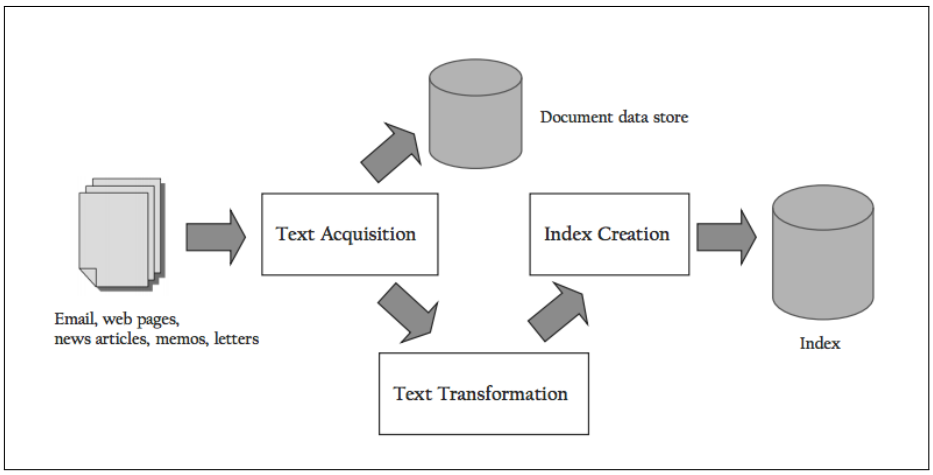
2.1.1 Sistema de Recuperação de Informação

Para apresentar os conceitos anteriormente citados, descreve-se uma arquitetura de alto nível para RI e seu funcionamento. Um exemplo de aplicação real de um sistema de RI é um motor de busca (*search engine*). Como dito anteriormente, esses sistemas dão suporte a duas funções principais: *indexação* e *recuperação*.

No processo de *indexação*, seguindo a Figura 2, inicialmente tem-se uma coleção de documentos brutos que serão preprocessados (limpeza, *scanner*, formatação e etc) e então memorizados em um repositório central (*Text Acquisition*), enquanto isso as informação textuais serão trabalhadas para se construir um saco de palavras (*bag-of-words*) que serão organizadas em estruturas de índices utilizadas no processo de consulta. A estrutura comumente usada é o Índice Invertido onde cada termo terá associado a ele os documentos em que o termo está contido. Ainda no índice poderá conter informações de frequência, posição do termos e outros recursos referente ao termo. Cada termo será considerado um atributo (*feature*). Assim tem-se a criação e armazenamento dos índices (*Index Creation*). Em geral, esse processo é *off-line* (CROFT; METZLER; STROHMAN, 2010).

A outra etapa, *recuperação*, é também conhecida por processo de consulta. Conforme a Figura 3, com uma apresentação em alto nível, o usuário inicia o processo apresentando a necessidade de informação no formato de texto livre em linguagem natural como entrada na interface do sistema. A primeira tarefa é aceitar a consulta do usuário e transformá-la em uma representação semelhante aos documentos armazenados. A segunda tarefa é utilizar uma representação lógica, como vetores ou matrizes, para a consulta e em conjunto com o índice dos documentos, aplica-se um modelo lógico de RI para então ter como resultado a lista ordenada por relevância dos documentos. O sistema organiza esse resultado para apresentar ao usuário.

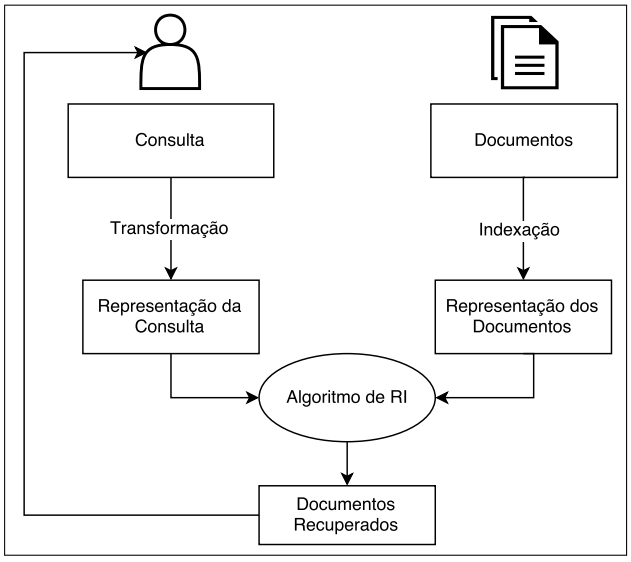
Figura 2 – Processo de indexação



Fonte**:** (CROFT; METZLER; STROHMAN, 2010)

Essa segunda tarefa, chamada de *ranqueamento* é a etapa central desse processo.

Figura 3 – Modelo Conceitual do Processo de Recuperação Básico



A eficiência do *ranqueamento* depende dos índices e a eficácia depende do modelo de recuperação (CROFT; METZLER; STROHMAN, 2010), ou seja, a qualidade da recuperação da informação vai depender da execução em conjunto dessas duas tarefas.

2.1.2 Representação dos documentos

Considere t o número de termos do índice. O conjunto de todos os termos da indexação forma o vocabulário. De modo a obter uma representação lógica para modelos de RI utiliza-se a ponderação de termos. Essa ponderação tem objetivo diferenciar os termos dentro

dos documentos de acordo com suas propriedades. São exemplos a frequência, incidência e, a mais utilizada, o Inverso da Frequência dos Termos (TF-IDF). A fim de notação chama-se $w_{i,j}$ o peso que representa a ponderação do termo k_i no documento d_j da seguinte forma:

$$M = \begin{array}{c} \\ k_1 \\ k_2 \\ \dots \\ k_t \end{array} \begin{array}{c} \begin{array}{cccc} d_1 & d_2 & \dots & d_n \end{array} \\ \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,n} \\ w_{2,1} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ w_{t,1} & \dots & \dots & w_{t,n} \end{bmatrix} \end{array} \tag{2.1}$$

Seja M a matriz 2.1. Cada coluna de M representa um documento $d_j = \{k_1 : w_{1,j}, k_2 : w_{2,j} \dots k_t : w_{t,j}\}$ por um conjunto de termos k e valor númerico associado $w_{i,j}$. O termo k pode ser representado por uma palavra, modelo unigram (1-gram) ou saco de palavras (*bag-of-words*), ou pela combinação de n palavras, modelo n-gram.

Outra representação comumente utilizada é matriz de termos por termos, ou matriz de coocorrência C. Sendo M^T a transposta de M, a matriz $C = M \cdot M^T$ uma matriz de correlação entre termos. Cada elemento $c_{u,v}$ é:

$$c_{u,v} = \sum_{d_j} w_{u,j} \times w_{v,j} \tag{2.2}$$

Quanto maior o número de documentos nos quais os termos k_u e k_v coocorram, maior será essa correlação $c_{u,v}$. A motivação é que termos que ocorrem próximos uns aos outros têm correlações mais fortes do que termos que ocorrem a uma grande distância (BAEZA-YATES; RIBEIRO-NETO, 2013).

Inverso da Frequência dos Termos: TF-IDF

O TF-IDF atribui um peso $w_{i,j}$ usando duas propriedades do termo: frequência e o inverso da frequência do termo. Segue a fórmula mais comum conforme proposto por Salton e Yang (SALTON; YANG, 1973):

$$w_{i,j} = \begin{cases} (1 + \log f_{i,j}) \times \log \frac{N}{n_i} & \text{, se } f_{i,j} > 0 \\ 0 & \text{, caso contrário} \end{cases} \tag{2.3}$$

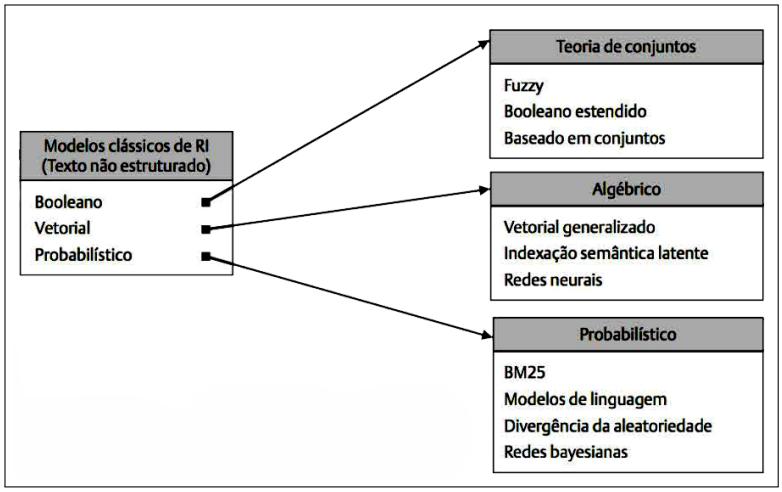
Onde $f_{i,j}$ é a frequência do termo i no documento j, N é número de documentos na coleção e n_i quantidade de documentos que contém o termo i.

Seja $w_{i,j}$ alto quando k ocorre muitas vezes dentro de um pequeno número de documentos (dando assim alta capacidade de discriminação a estes documentos) e baixo quando o k ocorre menos vezes em um documento, ou ocorre em muitos documentos (oferecendo assim um sinal de uma fraca relevância) (SCHÜTZE; MANNING; RAGHAVAN, 2008).

2.2 MODELOS DE RECUPERAÇÃO DE INFORMAÇÃO

Os modelos tradicionais de recuperação de informações, como na Figura 4, podem ser divididos principalmente na categoria de modelo Booleano, modelo Vetorial e modelo Probabilístico (BAEZA-YATES; RIBEIRO-NETO, 2013). Nas seções a seguir, apresenta-se cada modelo e suas características.

Figura 4 – Taxonomia dos modelos de Recuperação clássicos



Fonte**:** (BAEZA-YATES; RIBEIRO-NETO, 2013)

2.2.1 Modelo Booleano

Os primeiros sistemas de RI utilizavam a álgebra booleana onde as consultas eram especificadas pelo usuário usando uma combinação complexa de ANDs, ORs e NOTs booleanos (SINGHAL, 2001).

O modelo Booleano é um modelo de recuperação simples baseado na teoria de conjuntos e na álgebra Booleana. A álgebra booleana é um conjunto de operações lógicas entre dois conjuntos, como conjunção, disjunção e complemento. O modelo Booleano considera que

os termos da indexação estão presentes (1) ou ausentes nos documentos (0).

Formalizando o modelo Booleano, chamaremos de q uma consulta com os termos da indexação. Considere $c(q)$ como qualquer dos componentes conjuntivos da consulta. Dado um documento d_j, sendo $c(d_j)$ seu componente conjuntivo de documento correspondente, então a similaridade entre o documento e a consulta q definida por (BAEZA-YATES; RIBEIRO-NETO, 2013):

$$sim(d_j,q) = \begin{cases} 1 & \text{, se } \exists c(q) \mid c(q) = c(d_j) \\ 0 & \text{, caso contrário} \end{cases} \tag{2.4}$$

Então na equação 2.4 para o modelo Booleano, d_j será relevante caso $sim(d_j,q) = 1$, caso contrário será não relevante. Os sistemas booleanos são simples de implementar e atendem de forma básica, porém, têm várias deficiências, por exemplo, os documentos não saem com resultado ordenado, pois não existe uma noção intrínseca de classificação de documentos. Porém, são passíveis de configuração da ordem pela data ou outro atributo do documento. Outro aspecto é a transferência para o usuário da complexidade de formar uma boa consulta. Entretanto, os usuários avançados sentem mais controle no processo de recuperação (SINGHAL, 2001).

Ausência de crítica, por parte do modelo booleano, na relevância levou surgimento de outros modelos que adicionam o conceito de ranqueamento associando a um valor numérico aos documentos selecionados permitindo assim uma classificação.

2.2.2 Modelo Vetorial

O problema do Booleano é que ele pode trazer uma grande quantidade de documentos nos quais seja impossível localizar algo relevante. É necessário que o conceito de relevância esteja associado a um valor numérico e assim ranquear os documentos para trazer primeiro aqueles que sejam mais próximos da intenção de consulta do usuário.

Como visto anteriormente os documentos podem ser representados por vetores de pesos e atributos. Utilizando essa representação será aplicada a álgebra linear de modo a definir o conceito de similaridade entre consulta e documento. Isso é o modelo Vetorial (SALTON, 1971). No modelo Vetorial, definimos $w_{i,j}$ ($w > 0$ e não binário) como sendo o peso que representa o par termo (k_i) e o documento (d_j). Cada termo são mutuamente independentes. Seja t o número de termos. A representação do documento $d_j = (w_{1,j}, w_{2,j}, ..., w_{t,j})$ e da consulta $q = (w_{1,q}, w_{2,q}, ..., w_{t,q})$ são vetores com t dimensões onde $w_{i,q}$ é o peso associado ao par termo-

consulta $(k_{i,q})$, com $w_{i,q} > O$. Os pesos no modelo vetorial são basicamente os valores da frequência TF-IDF (BAEZA-YATES; RIBEIRO-NETO, 2013).

Assim como no booleano, precisa-se achar a relevância baseada na similaridade de d_j e q, ou seja, $sim(d_j, q)$. Agora que temos dois vetores, o coseno do ângulo será usado para quantificar a correlação (*cosine similarity measure*) (QIAN *et al.*, 2004) e já que como todos os vetores têm tamanhos iguais a t, portanto, pode-se usar o produto interno entre os vetores, da seguinte forma:

$$sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \tag{2.5}$$

Depois de calculado o grau de similaridade pode-se então ordenar de forma decrescente os documentos e ter assim os resultados por sua relevância em relação à consulta.

O modelo vetorial possui uma estratégia simples para ranqueamento para coleções genéricas e já foi comparado com técnicas alternativas e mostrou-se bom e sólido para ranqueamento. É o modelo mais popular e utilizado. Dentre as vantagens podem citar: melhora a qualidade da recuperação; traz documentos que se aproximam da consulta; e possui uma normalização embutida ao processo. Quanto a desvantagem é que os termos são considerados independentes e fazer o contrário é desafiador, pois o oposto poderia trazer resultados ruins (BAEZA-YATES; RIBEIRO-NETO, 2013).

2.2.3 Modelo Probabilístico

Em RI a modelagem probabilística é o uso de um modelo de recuperação que ranqueia os documentos em ordem decrescente de acordo com a probabilidade de relevância do documento com relação à necessidade de informação do usuário.

O modelo probabilístico foi proposto incialmente Robertson e Sparck Jones (ROBERTSON; JONES, 1976). A ideia do trabalho deles segue a seguinte hipótese. Imagine que o usuário executa uma consulta e o modelo de recuperação traz os documentos relevantes e que estes documentos seriam a resposta perfeita. Isso significa que a consulta do usuário é a especificação de um conjuntos de atributos do conjunto perfeito. Como durante a consulta não se conhece esse conjunto de atributos então o que se faz é uma estimativa inicial dessas propriedades através de uma descrição probabilística.

O modelo supõe que exista um conjunto de todos os documentos que o usuário prefira como conjunto resposta para a consulta. Tal conjunto resposta perfeito é chamado de R e deve maximizar relevância da informação para o usuário. Os documentos desse conjunto são previstos como relevantes à consulta q e já os documentos que não estão nesse conjunto são previstos como não relevantes em $\overline{R}$. Dessa forma, probabilisticamente, podemos ranquear os documentos D com a consulta q, o que pode ser chamado de princípio da classificação probabilística. Como mostrado nos modelos anteriores para recuperação de informação precisa-se definir a $sim(d_j,q)$, como a seguir (BAEZA-YATES; RIBEIRO-NETO, 2013):

$$sim(d_j,q) = \frac{P(R|d_j,q)}{P(\overline{R}|d_j,q)} \tag{2.6}$$

Onde $P(R|d_j,q)$ é a probabilidade de que o documento $dj = w_{1,j}, w_{2,j}...w_{t,j}$ com a representação d_j seja relevante para a consulta q. Além disso, $P(\overline{R}|d_j,q)$ é a probabilidade de que o documento d_j não seja relevante para a consulta q.

Utilizando o teorema de *Bayes*:

$$sim(d_j,q) = \frac{P(d_j|R,q) \times P(R,q)}{P(d_j|\overline{R},q) \times P(\overline{R},q)} \tag{2.7}$$

Como $P(R|q)$ e $P(\overline{R}|q)$ são os mesmos para todos os documentos da coleção, aplicando o uso da hipótese da independência binária e então convertendo os produtórios em logaritmos finalmente chegamos a fórmula reduzida.

$$sim(d_j,q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{P_{iR}}{1-P_{iR}}\right) + \log\left(\frac{1-q_{iR}}{q_{iR}}\right) \tag{2.8}$$

O conjunto R é desconhecido, então não sabem-se as probabilidades de P_{iR} e q_{iR}. Com o uso de uma tabela de contingência das incidências de termos é possível estimar esses valores (ROBERTSON; JONES, 1976):

$$P_{iR} = \frac{ri}{R}, q_{iR} = \frac{n_i - r_i}{N-R}, \tag{2.9}$$

Seja N o número de documentos, n_i o número de documentos que contêm o termo k_i. Seja R o número de documentos do resultado da consulta q e r_i a quantidade de documentos relevantes com o termo k_i.

Para este trabalho utiliza-se uma implementação de ranqueamento do modelo probabilístico chamado *Okapi BM25* (ROBERTSON; ZARAGOZA *et al.*, 2009), que resolve muito dos problemas que este modelo impõem para sua completude. A seguir a função BM25 descrita no livro do Russell (RUSSELL; NORVIG, 2010):

$$BM25(d_j, q) = \sum_{i}^{N} IDF(q_i) \cdot \frac{TF(q_i, d_j) \cdot (k+1)}{TF(q_i, d_j) + k \cdot (1 - b + b \cdot \frac{|d_j|}{L})} \tag{2.10}$$

Onde $|d_j|$ é o tamanho do documento d_j em termos. L é o tamanho médio dos documentos da coleção. Já k e b são parâmetros, os quais, em geral são $k = 2,0$ e $b = 0,75$. $TF(q_i, d_j)$ é número de vezes que a palavra q_i aparece no documento d_i. O $IDF(q_i)$ é a frequência inversa da palavra q_i da consulta q (pseudo-documento), dado por:

$$IDF(q_i) = \log \frac{N - DF(q_i) + 0.5}{DF(q_i) + 0.5} \tag{2.11}$$

Onde $TF(q_i)$ é o número de documentos que contêm a palavra q_i.

O BM25 consegue ser superior ao vetorial se os parâmetros forem bem ajustados. A função do BM25, assim como os outros modelos clássicos, utiliza um modelo de palavra que trata todas as palavras como completamente independentes, mas sabe-se que algumas palavras são correlatas. Exemplo: "TV"com "televisão"e "televisor".

Os modelos são o centro do processo de recuperação, porém, sem a devida representação dos documentos e da consulta não haveria bons resultados na recuperação, principalmente da consulta, que carrega a necessidade de informação do usuário. A seguir, apresenta-se a teoria por trás da consulta a fim melhorar os resultados relevantes da recuperação.

2.3 CONSULTA

A consulta é o elemento chave entre a necessidade de informação do usuário e os documentos que devem ser recuperados. A responsabilidade de dizer a intenção da consulta é do usuário, pois, a relevância está diretamente ligada ao seu julgamento do que é recuperado.

O usuário ao utilizar o SRI tem como principal entrada a consulta. Esta é formada por um texto em linguagem natural com palavras ou frases que correlacionam a sua necessidade de informação. A consulta então recebe o mesmo processamento (limpeza, tokenização, indexação,

contagem e etc) recebido pelos documentos passando a ser um pseudo-documento. Semelhante como apresentado nas colunas da matrix 2.1 o texto livre passa a ter a seguinte representação:

$$\begin{array}{c} \\ k_1 \\ k_2 \\ \ldots \\ k_t \end{array} \begin{array}{c} q \\ \left[\begin{array}{c} w_1 \\ w_2 \\ \ldots \\ w_t \end{array}\right] \end{array} \tag{2.12}$$

Seja q o vetor que representa a consulta, k_i o termo do vocabulário e w_i a ponderação caso a palavra incida ou correlacione-se com os termos da consulta.

É um fardo para o usuário projetar boas consultas principalmente quando desconhece a coleção de documentos. Na prática, quando o usuário não encontra os documentos, ele reformula sua consulta até chegar próximo ou exato do que precisa e assim melhorar os resultados. A principal estratégia para superar esse problema é participar ativamente desse processo, ou seja, o sistema de RI apoiar o usuário na formulação das consultas para que seja um processo curto e não um fardo. Isto pode ocorrer de duas formas modificando a consulta inicial para que se agregue informações que apoiem na recuperação de documentos (BAEZA-YATES; RIBEIRO-NETO, 2013):

(a) ***Feedback* de Relevância:** quando o usuário fornece explicitamente informações sobre os documentos relevante.

(b) **Expansão de Consulta:** quando somente as informações relacionadas a consulta são utilizadas para expandi-lá.

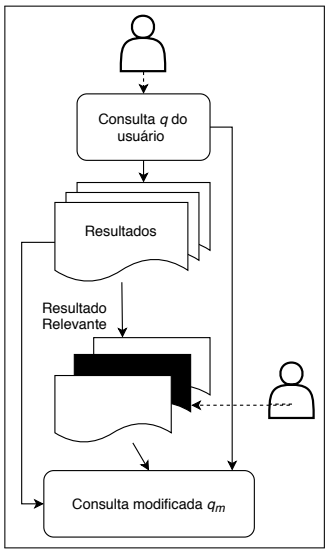
As duas abordagens seguem duas etapas: obter a informação que irá realimentar a consulta original q e transformar q em q_m de modo a utilizar essa informação de forma eficaz. A seguir o processo de cada abordagem.

2.3.1 *Feedback* de Relevância

Feedback de Relevância (FR) captura os resultados que inicialmente são retornados de uma determinada consulta e agrega as informações fornecidas pelo usuário a respeito dos resultados que são relevantes para executar uma nova consulta. Na prática, a ideia central consiste no usuário selecionar os n primeiros resultados da primeira consulta e extrair os termos mais relevantes para assim reformular a consulta com a importância desses termos e o da consulta

original. O esperado é que como usuário informou quais documentos são relevantes a partir da primeira consulta a próxima já reformulada pelo sistema se encarregue de aproximar os documentos relevantes e distanciar os irrelevantes.

Figura 5 – *Feedback* de Relevância



O FR apresenta as seguintes características: (I) o usuário somente participa do julgamento de quais documentos são relevantes (documentos relacionados) e não participa do processo de reformulação da consulta e (II) a tarefa de busca passa a ser em pequenas etapas que são mais fáceis de entender (BAEZA-YATES; RIBEIRO-NETO, 2013). Conforme mostrado na Figura 5, o usuário essencialmente reforça a decisão inicial do sistema, tornando a consulta expandida mais semelhante aos documentos relevantes recuperados (CARPINETO; ROMANO, 2012).

Em geral, na literatura se segue duas linhas de pesquisas que depende de qual modelo de RI é utilizado. Normalmente para o modelo vetorial se usa o método Rocchio (ROCCHIO, 1971) e para o modelo probabilístico utiliza o princípio do ranqueamento probabilístico (HARPER; RIJSBERGEN, 1978). Um estudo comparativo é realizado por Salton e Bucley (SALTON; BUCKLEY, 1990). Durante os anos de 2008 e 2009 o TREC dedicou uma de suas tarefas a *feedback* de relevância.

$$\overrightarrow{q_m} = \left(a \cdot \overrightarrow{q}\right) + \left(b \cdot \frac{1}{|D_r|} \cdot \sum_{\overrightarrow{D_j} \in D_r} \overrightarrow{D_j}\right) - \left(c \cdot \frac{1}{|D_{nr}|} \cdot \sum_{\overrightarrow{D_k} \in D_{nr}} \overrightarrow{D_k}\right) \tag{2.13}$$

A fórmula de Rocchio é exibida na equação 2.13. Seja $\overrightarrow{q_m}$ o resultado da consulta

modifica pela algoritmo de Rocchio aplicado na consulta original $\overrightarrow{q}$. Três parâmetros que ponderam a fórmula são configurados: a o peso da consulta original (q); b o peso dos documentos relacionados (D_r); e c peso dos documentos não relacionados (D_{nr}).

Por ser um processo que exige uma consulta inicial, o FR tem como desafio a compilação das informações de realimentação em larga escala. O problema é quando a base de documento não possui uma domínio específico podendo assim trazer ruídos a consulta do usuário. Outro desafio é não ser tão explícito a ponto de incomodar o usuário na tarefa de busca. Por isso atualmente informações de cliques e comportamento do usuário são agregados ao contexto da consulta.

2.3.2 Expansão de Consulta

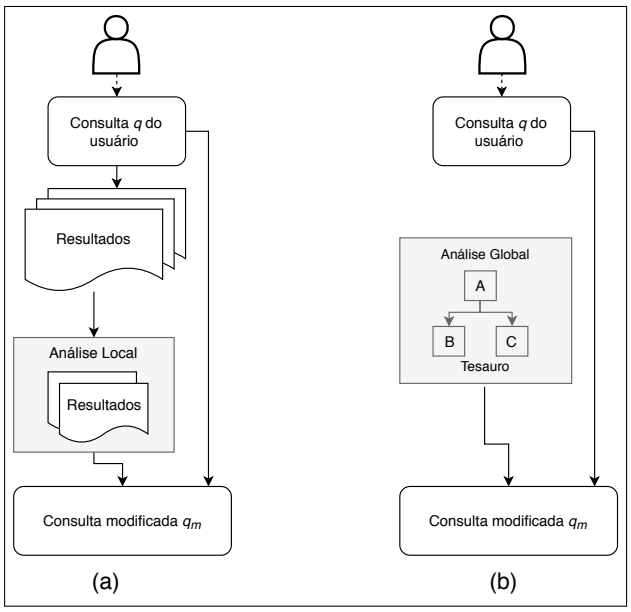
A abordagem de *feedback* de relevância tem seus desafios, porém conta com a ajuda do maior interessado nos documentos, o usuário. Algumas técnicas propõem a expansão automática da consulta sem interferência direta do usuário. São os chamados métodos implícitos. As informações necessárias para realimentação implícita podem ser coletadas a um custo baixo e sem sobrecarregar o usuário no SRI (JOACHIMS *et al.*, 2005). No entanto, as informações coletadas são mais difíceis de entender e potencialmente ruidosas. O que o sistema faz são suposições baseadas nas informações coletadas.

A principal motivação da expansão da consulta é adicionar termos significativos que ajudarão o usuário a remover a ambiguidade da linguagem natural e também expressar a necessidade de informação de forma mais detalhada que na consulta original (KROVETZ; CROFT, 1992).

Na literatura destaca-se duas abordagens principais para expansão de consulta (BAEZA-YATES; RIBEIRO-NETO, 2013): Análise Global e Análise Local.

A Figura 6 (a) exibe a Análise Local, onde os n documentos inicialmente capturados pela primeira iteração do usuário no SRI com a consulta q fornecerá um conjunto de documentos com alta relevância e estes serão utilizados para produzir termos que serão classificados e acrescentados para produzir q_m. Já a ilustração 6 (b), exemplifica a Análise Global que faz uso de um tesauro externo (seja construído por especialista ou de forma automática) para adicionar novos termos a q e produzir q_m.

Figura 6 – Expansão de Consulta Local e Global



2.3.3 Análise de Contexto Local

A expansão automática da consulta (sem interferência direta do usuário) pode ser baseada em métodos locais ou métodos globais. O trabalho de Croft (XU; CROFT, 1996) propõe usar os primeiros resultados de uma consulta para construir uma representação por coocorrência de conceitos (grupos de substantivos), por similaridade destes com a consulta, encontrar aqueles candidatos a serem agregados à expansão da consulta, ou seja, combinar análise local e global para EC.

Abaixo estão três etapas para usar a Análise de Contexto Local para expandir uma consulta q de um *corpus* (XU; CROFT, 1996):

1. Use um sistema IR padrão (INQUERY (CALLAN; CROFT; HARDING, 1992)) para recuperar as primeiras passagens com a melhor classificação. Uma passagem é uma janela de texto de tamanho fixo (300 palavras).
2. Encontrar os conceitos (grupo de substantivos mais significativos no topo da consulta original) e classificar utilizando a fórmula 2.14:

$$sim(q,c) = \prod_{k_i \in q}^{c} (\delta + \frac{\log(f(c,k_i) \times IDF_c)}{\log n})^{IDF_i}, \tag{2.14}$$

Onde k_i corresponde cada termo de q. IDF_i e IDF_c são o inverso da frequência sobre termo da consulta i e sobre o conceito c respectivamente. A função $f(c,ki)$ quantifica a correlação associativa que caracteriza a importância individual do termo c candidato

(possível termo a ser adicionado a expansão):

$$f(c,k_i) = \sum_{j=1}^{i} pf_{i,j} \times pf_{c,j} \tag{2.15}$$

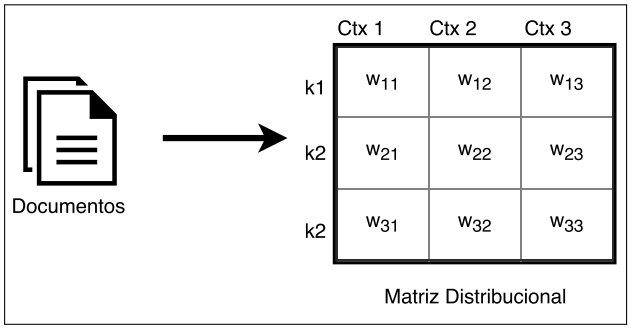
onde $pf_{i,j}$ é a frequência do termo ki na j-ésima passagem, e $pf_{c,j}$ é a frequência do conceito na j-ésima passagem.

3. Adicione a q os melhores conceitos classificados a fim de obter q'.

2.4 MODELOS SEMÂNTICOS DISTRIBUCIONAL

Esta seção brevemente apresenta Modelos de Semântica Distribucional (*Distributional Semantic Model*). Uma característica fundamental de um tesauro é a qualidade do grau de similaridade entre termos medida pela semelhança semântica. Os MSDs são construídos utilizando a hipótese distribucional de Harris (HARRIS, 1954). As palavras que possuem contextos semelhantes tendem a terem o mesmo significado. O MSD é uma representação das palavras em espaços geométricos de palavras onde vetores expressam conceitos, e sua proximidade é uma medida semântica. Isso significa que as palavras são semanticamente semelhantes se os contextos (palavras vizinhas) nos quais aparecem são semelhantes e deve levar a que suas representações sejam próximos. Construção de um MSD envolve a definição de um modelo de distribuição, formado por uma quádrupla (LOWE, 2001): vetores de palavra e dimensão; uma função que considera as coocorrências e como esses itens são representados no vetor final; uma função de similaridade definida sobre vetores; e eventualmente um mapeamento que transforma o espaço vetorial.

Figura 7 – Modelo Semântico Distribucional



Dado um *corpus* e através de um processo de treinamento não supervisionado é possível alcançar uma representação das palavras em um espaço de contexto, conforme Figura 7.

A partir dessa representação de palavras e suas operações de similaridade é possível construir um tesauro semântico. Vários trabalhos demonstram que espaços de palavras baseados em redes neurais (Word Embeddings) superam os modelos tradicionais baseados em contagem para calcular a similaridade da palavra (BARONI *et al.*, 2014; MIKOLOV *et al.*, 2013). Para este trabalho especificamente usa-se o modelo preditivo distribucional *Word2Vec* (MIKOLOV *et al.*, 2013).

Word2Vec é um método de *Word Embedding* utilizado para induzir modelos de espaço vetorial utilizando *deep learning* em redes neurais com modelos de linguagens (MIKOLOV *et al.*, 2013). Baseia-se em uma rede neural simplificada com o número de entradas proporcional ao de palavras do vocabulário. A camada escondida realiza uma projeção linear com tantos nós quanto a dimensionalidade desejada do espaço vetorial. Esse espaço de características é projetado sobre uma camada de saída hierárquica *soft-max*. A rede é treinada em cada par de exemplo de entrada-saída por vez e, para cada par, a diferença entre a saída esperada e a real da rede é calculada. Os pesos da combinação linear da rede são posteriormente ajustados para diminuir o erro usando o procedimento de *back propagation*. Este procedimento é repetido para todos os pares de dados de treinamento, muitas vezes em várias passagens sobre todo o conjunto de dados de treinamento, até que a rede converge e o erro não diminua mais (BENGIO *et al.*, 2003). Este método vem produzindo bons resultados, pois mostrou produzir representações que preservam importantes características linguísticas (MIKOLOV; YIH; ZWEIG, 2013). Na prática, estudos demostraram que uma das principais vantagens *Word2Vec* reside na sua escalabilidade (LEVY; GOLDBERG, 2014), permitindo o treinamento com até bilhões de palavras de texto de entrada em várias horas, diferenciando-se da maioria dos outros MSD.

2.5 AVALIAÇÃO DA RECUPERAÇÃO

Avaliar um sistema de RI é medir o quão bem o sistema atende a necessidade informação do usuário. O problema é que o julgamento de um resultado satisfatório cabe ao usuário, o que implica que o mesmo resultado pode ser interpretado de maneiras diferentes por usuários distintos. O procedimento comum de superar esse desafio é comparar o resultado produzido pelo SRI com os resultados sugeridos por humanos para o mesmo conjunto de consultas(BAEZA-YATES; RIBEIRO-NETO, 2013).

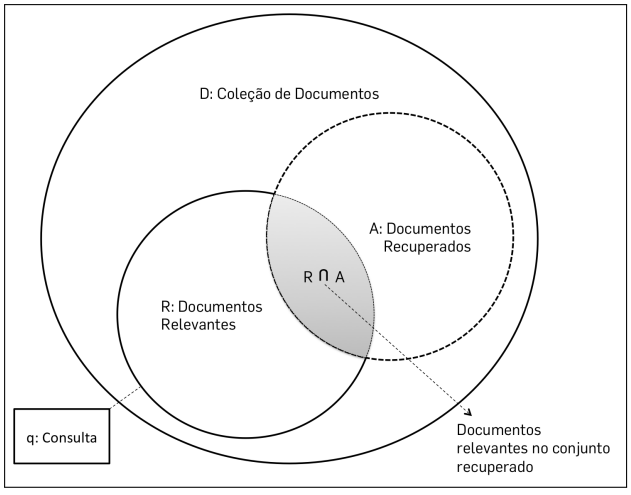
A comparação é feita com documentos de testes, chamadas de coleção de referência. Essa é a principal ferramenta usada para comparação e avaliação de SRI. A coleção é composta

de documentos, consultas (ou tópicos) e a relevância dos julgamentos (identificação da consulta e quais documentos foram julgados relevantes pela consulta). Essa organização conhecida como paradigma *Cranfield* (CLEVERDON, 1967) permitiu avaliar e comparar o par necessidade de informação-documento. Essas coleções permitem comparar diretamente os resultados obtidos por diferentes algoritmos de recuperação. Representa-se da seguinte forma:

- D conjunto de documentos
- Q conjunto de consultas
- R conjunto de avaliações (julgamento de relevância), onde $R = \{[q_m, d_j]\}, q \in Q \, e \, d \in D$, onde o julgamento é igual a 1 se relevante ou 0 para irrelevante.
- A conjunto de avaliações projetado pelo algoritimo de RI.

As avaliações produzidas em R previamente foram realizadas por especialistas que trabalharam classificando a relevância das consultas Q e os documentos D relacionados. A é o conjunto de resultados das consultas Q projetado por algum algoritmo de RI. Os conjuntos R e A são usados para comparação para avaliação do SRI.

Figura 8 – Diagrama de avaliação dos documentos recuperados na tarefa *Ad Hoc*



A eficácia é calculada medindo a capacidade dos sistemas de encontrar documentos relevantes. As duas medidas mais frequentes e básicas para medir a eficácia da recuperação de informação são o *Precision* (precisão) e o *Recall* (revocação) (SCHÜTZE; MANNING; RAGHAVAN, 2008). Considere o diagrama da Figura 8, seja agora $\mid R \mid$ somente o número de documentos relevantes para consulta q, $\mid A \mid$ apenas a quantidade de documentos recuperados por um SRI e $\mid R \cap A \mid$ o número de documentos relevantes que foram recuperados. As medidas de precisão e cobertura são definidas como segue:

- ***Precision*** é a fração de documentos recuperados que são relevantes.

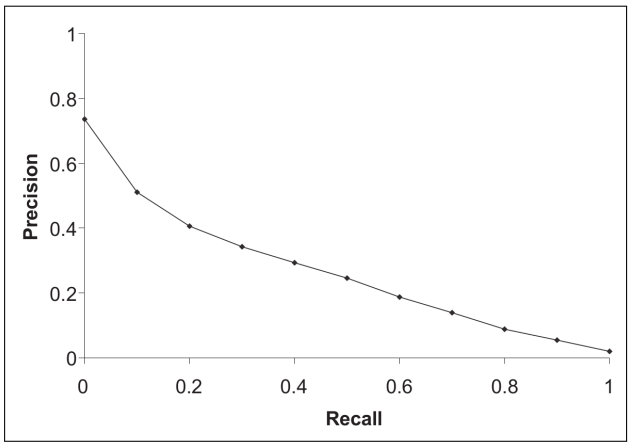
$$p = \frac{|\, R \cap A \,|}{|\, A \,|} \tag{2.16}$$

- ***Recall*** é a fração de documentos relevantes recuperados.

$$r = \frac{|\, R \cap A \,|}{|\, R \,|} \tag{2.17}$$

O *precision* e o *recall* são usados para medir o desempenho do sistema. O *recall* mede a capacidade do sistema de recuperar todos os documentos relevantes. O *precison* mede a capacidade do sistema de discriminar entre os documentos relevantes e não relevantes. A precisão e a revocação são suficientes para a medir a eficácia de um sistema. As medidas aplicadas isoladamente não requerem que o conjunto resposta esteja ranqueados, ou seja, avaliam somente a recuperação, pois são medidas baseadas em conjunto. Todavia, em um contexto de recuperação ordenado, o conjunto de documentos recuperados são naturalmente fornecidos pelos k primeiros documentos recuperados (RIJSBERGEN, 1979; SCHÜTZE; MANNING; RAGHAVAN, 2008). Para cada conjunto, os valores de *precision* e de *recall* podem ser plotados para fornecer a curva de *precisão-recall*, como a mostrada na Figura 9.

Figura 9 – Exemplo de curva de *precison-recall* médio



A cada requisição de uma consulta $q \in Q$ os valores (p,r) são calculados com 11 níveis de revocação (de 0% até 100%, a cada 10%, de documentos) e armazenados em uma tabela de *precision-recall*. Calcula-se a média para conjunto Q de (p,r) e então plota-se no gráfico os valores interpolados. A curva é comumente utilizadas para comparar a qualidade de diferentes algoritmos de recuperação. Uma maior área abaixo da curva indicam melhor eficácia do SRI. A curva pode esconder anomalias de algumas consultas, por ser resultado de uma média,

por isso algumas outras métricas sumarizadas são usadas para avaliar um SRI. Segue algumas comumente utilizadas:

1. **Média das Precisões Médias - MAP**: para um conjunto de consultas é a média das pontuações de precisão média para cada consulta.

$$\text{MAP} = \frac{\sum_{q=1}^{n} \text{AveP(q)}}{n} \tag{2.18}$$

   Onde *AveP* é a área sob a curva de *precision-recall* e *n* o total de consultas.

2. **Precisão em n (P@n)**: A precisão nos *n* documentos (P@n ou "Precisão em *n*"), que corresponde ao número de resultados relevantes na primeira página de resultados da pesquisa, mas não leva em consideração a ordem dos documentos relevantes entre os *n* do topo.

3. **R-Precisão**: é a razão (r/R) entre todos os documentos relevantes recuperados até a classificação *R*. Seja *R* o número de documentos relevantes de toda coleção e *r* o número de documentos recuperados.

4. F**-score /** F**-measure:** A média harmônica ponderada do *precision* e *recall*:

$$F = \frac{2 \cdot \text{p} \cdot \text{r}}{(\text{p} + \text{r})} \tag{2.19}$$

5. **Bpref**: calcula uma relação de preferência, ou seja, se os documentos julgados relevantes são recuperados antes daqueles julgados irrelevantes. Assim, baseia-se nas classificações relativas dos documentos julgados:

$$bpref = \frac{1}{R} \sum_{r} (1 - \frac{| \, n \text{ classificado acima de } r \, |}{min(R,N)}) \tag{2.20}$$

   onde *R* é o número de documentos julgados relevantes, *N* é o número de documentos julgados irrelevantes, *r* é um documento relevante recuperado e *n* é um membro dos primeiros *R* documentos irrelevantes recuperado.

3 TRABALHOS RELACIONADOS

Em média um usuário utiliza de duas a três palavras para descrever sua necessidade em SRI (CROFT; METZLER; STROHMAN, 2011). Essas palavras são escritas em linguagem natural e são carregadas de ambiguidade o que levaria a um modelo de recuperação a cometer erros ou omissões. O capítulo 2 discutiu-se a teoria utilizada ao problema de recuperação de informação e especificamente no aperfeiçoamento da consulta. Neste capítulo serão listadas algumas abordagens encontradas na literatura que tratam da realimentação de consultas com ACL e a construção de tesauros que usam Modelos Semânticos Distribucional (MSD), pois, é através da combinação dessas duas abordagens que o Luppar é construído.

3.1 EXPANSÃO DE CONSULTA COM ACL

A ideia de melhorar a consulta foi originalmente concebida por Rocchio (ROCCHIO, 1971). Evidências na literatura mostraram que os resultados da combinação das medidas de cobertura e precisão são melhorados em uma escala de 10% quando q é modificada para q_m (BUCKLEY; SALTON; ALLAN, 1994; LIU *et al.*, 2004; LEE; CROFT; ALLAN, 2008). As melhorias, em geral, resultam de dois procedimentos básicos: expandir a consulta adicionando novos termos e, ou, a reponderação desses termos.

Destaque para duas revisões sobre a expansão automática de consultas que detalham técnicas e abordagens em RI (CARPINETO; ROMANO, 2012; OOI *et al.*, 2015), na quais as soluções, em geral, melhoraram a eficácia do SRI resolvendo ambiguidade da consulta. Algumas abordagens são de análise global, análise local, análise de *log* do usuário ou análise de contexto. Podem ser com a participação do usuário (explícita) ou sem e automática (implícita). A revisão de Carpineto e Romano (CARPINETO; ROMANO, 2012) sugere algumas direções de pesquisas em expansão de consulta, e vale ressaltar a estratégia de combinar evidências, ou seja, mesclar diferentes métodos efetivos de refinamentos da consulta. Por exemplo, em Liu (LIU *et al.*, 2004) utiliza-se o tesauro WordNet com heurística sobre os conceitos combinando com métodos estatísticos locais e globais. Outro exemplo é Metzler e Croft (METZLER; CROFT, 2007) que combinam características linguísticas e rede de Markov para expansão.

Nessa perspectiva de mesclar soluções, o trabalho de Xu e Croft (XU; CROFT, 1996) foram pioneiros em combinar técnicas de análise local e global para expansão de consulta na qual denominaram de Análise de Contexto Local. Em uma visão geral para ACL, Xu e Croft propõem fazer uma seleção de atributos baseada na coocorrência de conceitos (grupos

de substantivos) local, considerando que os melhores termos são aqueles que coocorrem com tantos termos da consulta quanto possível dentro dos documentos mais bem classificados ou passagens de documentos. Dados utilizados para avaliação foram WEST, TREC3 e TREC4. O SRI utilizado foi INQUERY (CALLAN; CROFT; HARDING, 1992) que é baseado no modelo probabilístico. Os resultados do ACL foram comparados com as abordagens *baseline* (sem expansão), relevância local (método local) e *Phrasefinder* (método global).

Nos anos seguintes Xu e Croft (XU; CROFT, 2000) publicam outro trabalho reforçando a melhoria da recuperação de informação através da ACL. Dois novos idiomas, o espanhol e chinês, são avaliados. Diferentes tamanhos de passagens também são comparados. A hipótese de que um termo comum aos principais documentos classificados como relevantes no topo tenderá a coincidir com todos os termos da consulta nos documentos mais bem ranqueados é então detalhada com considerações formais para construção da métrica de associação entre o termo candidato e a consulta original.

A ACL passa a ter características locais e globais bem definidas. O melhor da análise local é poder utilizar os resultados do topo na qual assume-se serem os melhores resultados relacionados. Na análise global utiliza-se o conceito de contexto e estruturas frasais sobre o conjunto local (BAEZA-YATES; RIBEIRO-NETO, 2013). A medida de correlação adotada na ACL é de associação 2.15, ou seja, apenas a frequência dos termos é levada em consideração. No livro de Baeza sugere que poderia ser utilizado um fator de correlação métrico, ou seja, que considere a distância entre os termos para a função $f(c,k_i)$. Essa afirmação abre margem para que outros tipos de fatores passem a ser utilizados como formas de avanços na ACL original, como o caso de utilizar um *cluster* escalar ou qualquer outra medida que seja capaz de classificar os termos linguisticamente.

Em 2012, Wan e Wang propõem uma combinação agregada de ontologia com ACL (WAN *et al.*, 2012). O conhecimento estatístico das categorias de ontologias a que pertencem os documentos que contêm a palavra a ser consultada pode ser usado para obter o grau de ligação da palavra consultada a diferentes categorias de ontologias. Então criam a seguinte abordagem: ponderar os termos usando ACL padrão; ponderar estatisticamente os termos utilizando ontologia; e unir os conjuntos de termos candidatos. O critério de união é somar os pesos dos termos para assim classificá-los. O uso da ontologia é justificado em quanto maior a frequência de ocorrência de uma palavra consultada em documentos anexados à ontologia, maior é o grau de inserção da palavra consultada na ontologia (FAZZINGA *et al.*, 2011).

Já no trabalho de Ermakova e Mothe (ERMAKOVA; MOTHE, 2016) a combinação

é embutida ao método de ACL. Utiliza-se da hipótese que características da linguagem natural devem ajudar a decidir os melhores termos candidatos, isto é, que alguns tipos de termos devem ser melhores candidatos (por exemplo, substantivo sendo melhor que advérbios). Então desenvolveu um método que além das características da ACL também considera as informações de *Part Of Speech* (POS) para ponderar diferentemente os termos candidatos a expansão da consulta. A função que pondera a relação entre termo e a consulta se diferencia do trabalho de Croft por além de usar as estatísticas do documento (coocorrência), também usam a distância dos termos que envolvem a consulta, vide Tabela 2. Mostrou-se eficiente nos resultados em relação aos trabalhos anteriores.

Tabela 2 – Função de similaridade entre termos cadidatos e os da consulta original

	(XU; CROFT, 2000)	(WAN *et al.*, 2012)	(ERMAKOVA; MOTHE, 2016)
$f(c,k_i)$	$\sum_{j=1}^{n} pf_{i,j} \times pf_{c,j}$	$f(c,OntK_i) + f(c,k_i)$	$\frac{1}{max(1,dist(c,k_i)-2)}$

A Tabela 2 resume os trabalhos citados no que diz respeito às funções $f(c,k_i)$ usadas no algoritmo ACL que correlacionam a similaridade entre um termo c candidato e um termo k_i da consulta, conforme os trabalhos relacionados. O trabalho de Croft (2000) com a medida de correlação por associação, Wan (2012) com a união da estatística fornecida pela ontologia e Ermakova (2016) com fator de correlação métrica embutida na distância entre termos $dist(c,k_i)$. Para esta pesquisa usa-se um fator de correlação preditivo com base no operador de similaridade do *Word2vec* (MIKOLOV *et al.*, 2013).

3.2 CONSTRUÇÃO AUTOMÁTICA DE TESAURO

Um tesauro é uma lista de palavras com significados semelhantes ou afins. O algoritmo de construção do tesauro diferencia as abordagens utilizadas. Esses algoritmos podem ser caracterizados em três dimensões: pela abrangência (escopo) dos textos utilizados na construção do tesauro, pela noção de contexto adotada (quando se utiliza a hipótese de que o significado das palavras é dado pelo seu contexto de uso), e pela medida utilizada na avaliação da semelhança semântica entre as palavras.

Em relação à abrangência, um tesauro pode ser amplo, construído para todo o vocabulário e ocorrências conhecidas de uso das palavras de uma língua, ou pode ser restrito aos termos presentes em uma coleção particular de documentos. A noção de contexto refere-se a hipótese adotada sobre como se determina o significado das palavras na língua. Por exemplo, pode-se adotar a hipótese de que o significado de uma palavra pode ser inferido das palavras

que ocorrem no seu entorno, sendo que o entorno adotado pode ser uma janela determinada, a sentença, todo o parágrafo ou mesmo o documento. A medida de semelhança, por sua vez pode variar sendo probabilística ou determinística, pode considerar ou não a distância entre as palavras, ou pode ainda utilizar elementos externos tais como uma ontologia de domínio (BHOGAL; MACFARLANE; SMITH, 2007).

Um dos tesauros mais conhecidos é WordNet (MILLER, 1995) construído de forma manual e com características globais. Tem a vantagem de trazer informações léxicas o que resolvem problemas de ambiguidade em alguns casos. Já as desvantagens é que são genéricos, dessa forma não trazem ganhos em domínios específicos e trabalhosos para incluir novos termos (OOI *et al.*, 2015). Mesmo sendo difícil de crescer esse tipo de tesauro vários são os trabalhos (GONG; CHEANG; HOU, 2005; LU *et al.*, 2015; HSU; TSAI; CHEN, 2006) que utilizam essa abordagem para carregar pares de sinônimos com o objetivo de reformular as consultas.

Os tesauros construídos de forma automática são baseados na hipótese distribucional de Harris que afirma que as palavras que são usadas e ocorrem nos mesmos contextos tendem a ter significados semelhantes (HARRIS, 1954). A partir dessa hipótese construíram-se teorias e métodos para representar e quantificar a similaridade entre itens de dados linguísticos. Um modelo baseado nessa hipótese é chamado de Semântica Distribucional. Para criar uma representação, dois tipos de modelos são geralmente construídos: modelos de contagem ou modelos preditivos. Na abordagem de contagem, tradicionalmente, usam-se as estatísticas de coocorrência das palavras e assim cria-se vetores no espaço de palavras (TURNEY; PANTEL, 2010). A alta dimensionalidade dessa representação é então reduzida, sendo a densificação realizada por decomposição em valores singulares (*Single Value Decomposition* (SVD)) (LANDAUER; DUMAIS, 1997) ou por análise de componentes principais (PCA) (LEBRET; COLLOBERT, 2015). Já as abordagens preditivas utilizam as estatísticas dos termos para treinar redes neurais que criam vetores densos usados como representação dos termos (*Word Embeddings*)(MIKOLOV *et al.*, 2013). Para estas representações, em geral, são utilizadas uma das seguintes medidas de similaridade: coseno, medida de Lin (LIN, 1998) ou coeficiente de Dice (CURRAN; MOENS, 2002). Vários trabalhos demonstram que espaços de palavras baseados em redes neurais superam os modelos tradicionais baseados em contagem para calcular a similaridade da palavra (BARONI *et al.*, 2014) (MIKOLOV; YIH; ZWEIG, 2013). Especificamente o *Word2Vec*, vem motivando bons resultados, pois, mostrou produzir representações que preservam importantes características linguísticas (MIKOLOV; YIH; ZWEIG, 2013). Uma das principais vantagens *Word2Vec* reside na sua escalabilidade, permitindo o treinamento com até bilhões de palavras de texto de entrada

em várias horas, diferenciando-se da maioria dos outros MSD (LEVY; GOLDBERG, 2014).

No sentido de combinar métodos para aprimorar o processo de enriquecimento da consulta para resolver o problema de desambiguação, o Luppar utiliza-se da ACL como método para seleção de novos termos para expansão da consulta embutindo os recursos fornecidos por um tesauro semântico distribucional construído com *Word2Vec*.

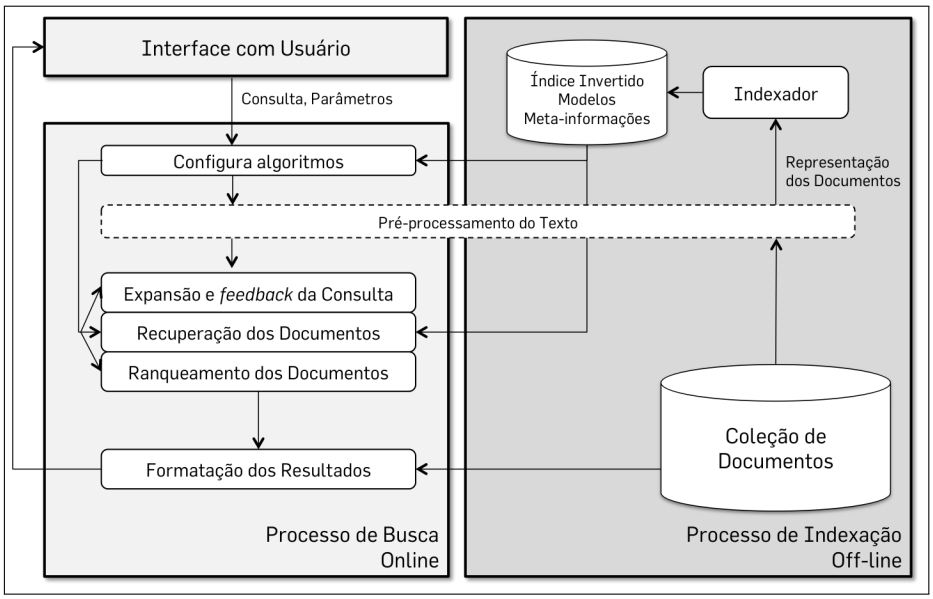
4 LUPPAR: SISTEMA DE RECUPERAÇÃO DE INFORMAÇÃO DOTADO DE ANÁLISE DE CONTEXTO LOCAL BASEADO EM MODELO SEMÂNTICO DISTRIBUCIONAL

Neste capítulo é apresentado o desenvolvimento do SRI, o qual se chamou de Luppar[1], que servirá de suporte aos demais objetivos. A abordagem proposta no capítulo 1 será embutida no SRI. A metodologia que é seguida neste trabalho divide os esforços em cinco etapas: desenvolvimento do SRI, algoritmos de recuperação, construção do tesauro MSD, expansão da consulta utilizando ACL com MSD e coleta de documentos. Segue-se descrevendo cada etapa, os modelos utilizados e artefatos produzidos.

4.1 DESENVOLVIMENTO DO SRI

Um sistema de RI pode vir atender a busca de textos não estruturados em diversos ambientes tais como os arquivos de um computador pessoal, páginas na web, documentos em redes corporativas ou em ferramentas que acumulam dados especializados. Essa diversidade traz especificidades para a definição da arquitetura. Para a pesquisa escolheu-se desenvolver um motor de busca *web* com coleções fechadas. A arquitetura em alto nível ilustrado na Figura 10.

Figura 10 – Arquitetura do sistema em alto nível.



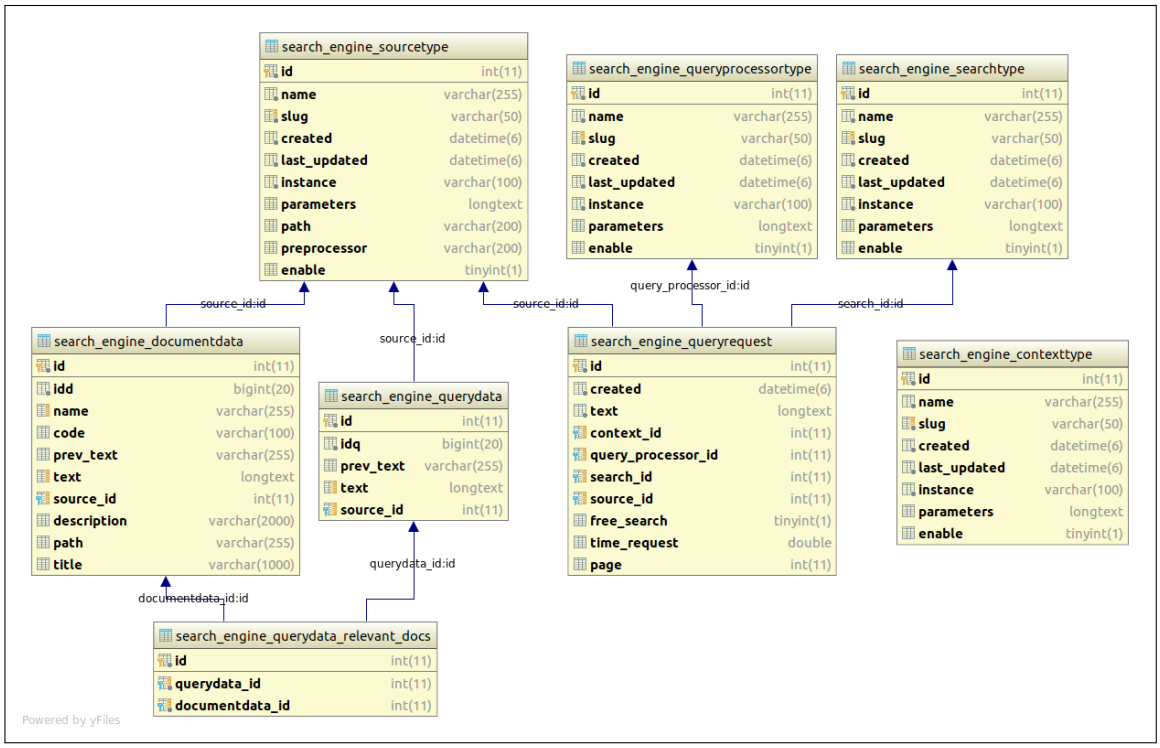
A arquitetura possui dois macro-processos: Indexação e Busca. O primeiro *off-line* e responsável por manter os recursos necessários ao funcionamento do segundo. A busca é *online*

[1] Um protótipo de Luppar pode ser acessado no endereço http://luppar.com

e compartilha do mesmo pré-processamento de texto da indexação.

Na indexação a primeira etapa é obter a coleção fechada de documentos. Essa coleção é formada por documentos D, consultas Q e avaliações A. Estes itens são armazenados fisicamente no sistema de arquivos e carregados em banco de dados de forma estruturada com identificadores e o texto em sua forma bruta, vide modelo na Figura 11.

Figura 11 – Diagrama entidade-relacionamento da aplicação Luppar



Depois dos dados armazenados são então aplicado um *pipeline* com as seguintes etapas: remoção de acentos, pontuação e espaçamento; extração das *stopwords* (palavras que não agregam valor); processo de radicalização (*stemming*) Porter (PORTER, 1980) e tratamento das variações de gêneros, categorização lexical (*tags*) e ajustes de acordo com estrutura do texto. O resultado é uma lista de termos que compõem o vocabulário da coleção (dicionário de termos). Esses termos seguirão para a representação lógica dos documentos.

Com o vocabulário formado inicia-se o processo de construção dos índices, no qual os termos, seus atributos e documentos correspondentes formarão uma estrutura de dados. Para esta pesquisa utilizamos um Índice Invertido Completo (ZOBEL; MOFFAT, 2006), mas somente com as seguintes características:

$$I = \{\langle t_1, f_t, (d, f_{d,t}, P_1 ... P_{m_{d,t}})_j \rangle, \langle t_2, f_t, (d, f_{d,t}, P_1 ... P_{m_{d,t}})_j \rangle ... \langle t_n, f_t, (d, f_{d,t}, P_1 ... P_{m_{d,t}})_j \rangle\} \quad (4.1)$$

Seja I o índice; t cada palavra no vocabulário de I; d um identificador para um documento; f_t a

contagem dos documentos que contêm t; $f_{d,t}$ a frequência de t em d; e $P_{m_{d,t}}$ a m-ésima posição de t em d. Para cada termo t é associado uma lista de tuplas do tipo $(d, f_{d,t}, P_1 ... P_{m_{d,t}})_j$ na qual j-ésimo documento. Podemos visualizar a estrutura de 4.1 é formada por um dicionário de dados, onde cada chave corresponde o termo t e cada valor é formado por um vetor de atributos agora citados. O uso do índice fornecerá acesso direto e rápido ao *bag-of-words* durante o processo de recuperação. Outras estruturas são produzidas durante essa etapa. Duas matrizes: sendo um documento por termo e outra de termo por termo, vide matriz 2.2. Todo o processo é *off-line*, ou seja, executado em *background* no sistema e independe do usuário. A estrutura é serializada em disco para um posterior carregamento em memória.

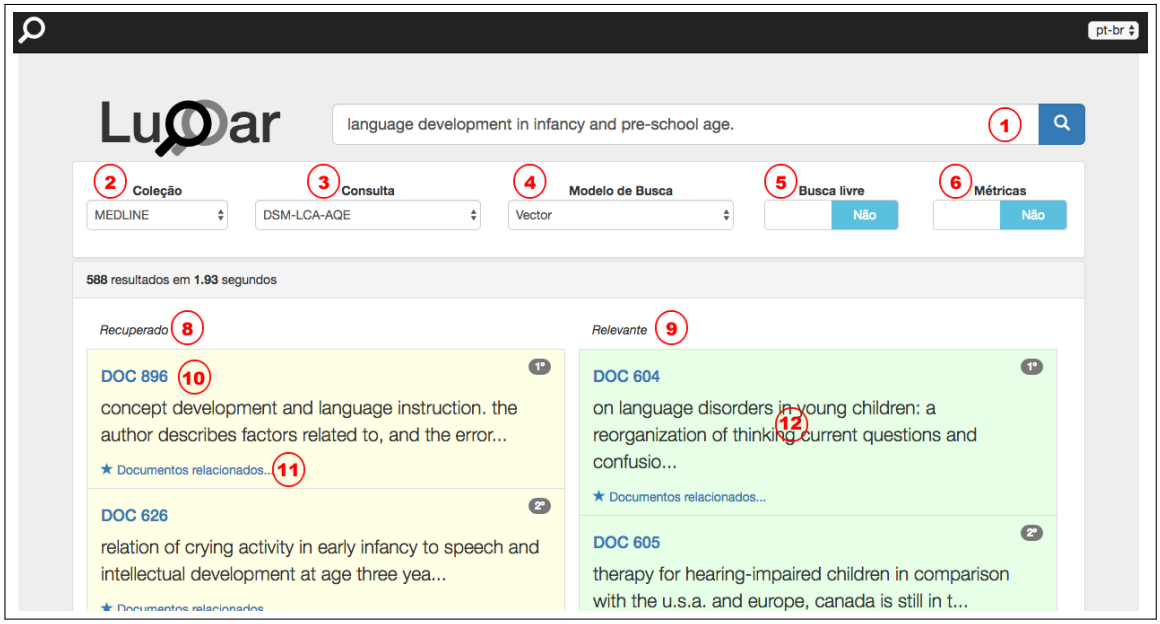
O segundo macro processo do Luppar é a etapa de Busca com uma arquitetura cliente-servidor, no qual a Interface do Sistema é o cliente e Módulo de Consulta como servidor.

4.1.1 Interface com Usuário

A Figura 12 ilustra a *interface* do sistema. O *layout* básico se assemelha aos buscadores como Google, Yahoo e Bing nos seguintes aspectos: possui um retângulo de busca para se especificar a necessidade de informação (1); possibilidade de autocompletar a frase do usuário com consulta hora conhecida (1); os resultados iniciais utilizam o padrão SERP (*Search Engine Result Page*): os resultados vêm em lista simples; paginados entre 10 a 20 itens; alinhados à esquerda na ordem dos mais relevantes; cada resultado contêm um título com *link* para o documento; e abaixo desse uma pequena sentença do documento. O que distingue o Luppar em sua *interface* é o caráter didático e uma tela voltada ao paradigma *Cranfield*. Em outras palavras a interface da aplicação conhece suas consultas e os resultados relevantes associados, assim é possível exibir os resultados relevantes, os documentos recuperados e avaliação da consulta através das métricas.

A *interface* permite a configuração em cinco níveis, Figura 12: (2) escolher a coleção de documentos; o modelo de expansão de consulta (3), o modelo de busca (recuperação e ranqueamento) (4); em (5) a opção de busca livre, no caso seja "sim"o usuário não precisa escolher uma das consultas criada por especialista através do autocompletar em (1); e em (6) a opção caso o usuário deseje visualizar os gráficos e métricas da avaliação em RI. Os resultados da consulta são exibidos em duas listas. A lista do lado esquerdo (8) corresponde os documentos que foram recuperados com a configuração da consulta escolhida pelo usuário e a segunda lista do lado direito exibe os documentos o qual se sabe que são relevantes para consulta. Este último

Figura 12 – Interface do Luppar seguindo o padrão SERP



somente é exibido se a "busca livre"(5) estiver em "não".

Outra característica do Luppar é a utilização de um *link* chamado *"documentos relacionados"* (11) abaixo dos resultados, a fim obter uma realimentação de relevância explícita por parte do usuário. Nesta etapa o sistema aplica o *Feedback* de Relevância. Essa ação exige uma consulta em duas etapas, já que se trata um refinamento para a primeira consulta ou uma forma do usuário deixar claro o caminho dos documentos que lhe interessam. Para este passo o Luppar implementa o algoritmo de Rocchio aplicados a consulta original q a fim de obter q_m, conforme equação 2.13.

4.1.2 Módulo de Consulta

A consulta inicia-se quando o texto livre é enviado ao servidor remoto com as demais preferências do usuário. Se escolhida expansão de consulta, há quatro opções disponíveis:

1. ***Baseline* (*BASE*):** o processo leva a representação da consulta q direto para o modelo de busca.
2. **Expansão por Análise de Contexto Local (*ACL*):** implementação do algoritmo de ACL clássico proposto por Xu e Croft (XU; CROFT, 1996).
3. **Expansão por Tesauro Externo WordNet (*WORDNET*):** o algoritmo usa-se do WordNet para gerar os sinônimos para cada termo da consulta e então esses sinônimos são usados na expansão (LI; GANGULY; JONES, 2016).
4. **Expansão por ACL com MSD (*ACL-MSD*)**: implementação da abordagem proposta

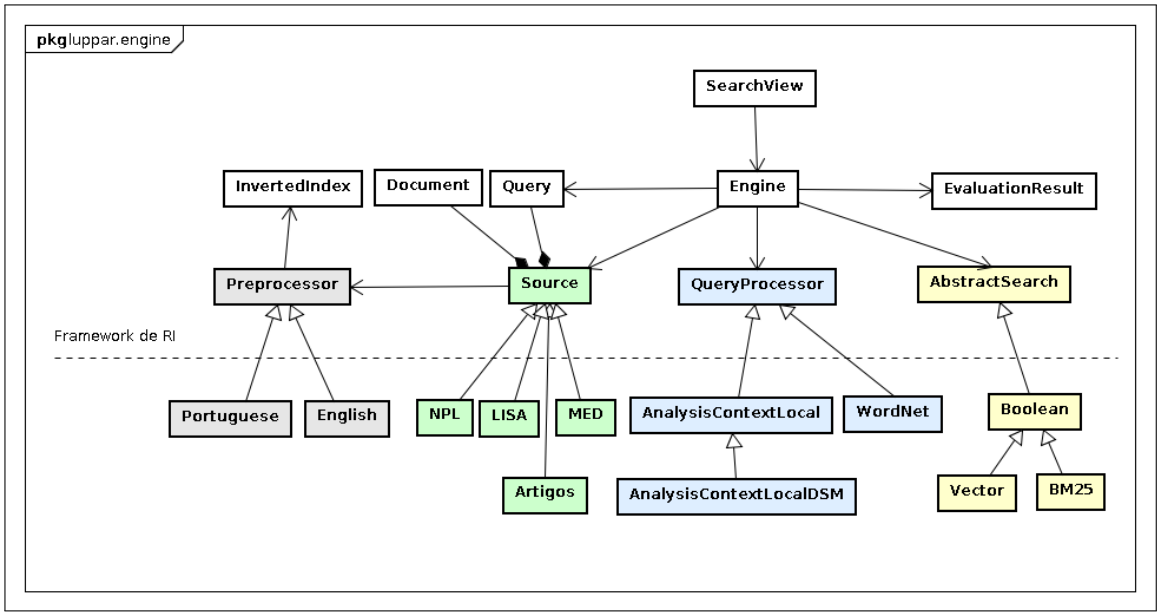
nessa pesquisa. A ACL utilizando um tesauro construído automaticamente com MSD. A seção 4.4 detalha essa abordagem.

O próximo passo é a consulta ser processada para uma representação igual ao do pré-processamento *off-line* dos documentos (indexação). A consulta ganha a representação de pseudo-documento. O índice invertido é selecionado de acordo com a coleção escolhida na *interface*. O vocabulário de referência e as matrizes são carregados em memória. Com a representação da consulta e dos documentos o Luppar executa o algoritmo de busca em RI e então retorna uma lista ordenada de documentos relevantes e a envia novamente ao usuário. Este pode refinar sua consulta escolhendo os documentos mais relevantes para iniciar novamente o ciclo de busca (*Feedback* de Relevância).

4.1.3 Aspectos da Implementação

O SRI foi desenvolvido na arquitetura cliente-servidor como são comumente os motores de busca *web*, apesar disso não é uma ferramenta de busca de páginas Web, mas de coleções fechadas e que seguem o formato *Cranfield* (vide seção 4.5). A princípio, foi necessário desenvolver um *framework* próprio de RI para atender o acoplamento de novas coleções (documentos) com diferentes algoritmos de RI. A interoperabilidade entre diversos algoritmos é um dos requisitos para manter diferentes combinações e abordagens. Essa particularidade fez-se possível devido às abstrações de código definidas no *framework*.

Figura 13 – Diagrama de classe do motor de busca do Luppar



O projeto possui uma aplicação com usuário (cliente) e um motor de busca (servidor). Este último é segmentado em um *framework* de RI e as implementações das classes abstratas do *framework*. Há quatro classes abstratas que regem o processo de RI, conforme mostrado no diagrama de classe da Figura 13:

1. *Preprocessor:* responsável pela representação do texto da coleção (índices invertidos com matrizes e dicionários de termos) e todo pré-processamento textual de acordo com idioma da coleção. O projeto implementa nos idiomas Inglês e Português.
2. *Source:* cabe a especialização dessa classe para definir de onde e como extrair as informações de consulta e documentos para forma textual. Dessa forma o projeto permite que diferentes fontes de aquivos venham de uma página *html*, um banco de dados ou arquivos em PDF e os transforme em um *corpus* reconhecido pelo sistema.
3. *QueryProcessor:* encarregado de executar os algoritmos de análise, representação e expansão da consulta. O Luppar implementa os algoritmos descritos na seção 4.1.2.
4. *AbstractSearch:* classe base para os modelos de busca em RI. Sua implementação exige, para recuperação e ranqueamento, uma função de similaridade entre consulta e documento.

Todas as implementações do projeto para operação de busca são orquestradas pela classe *Engine*. Esta conhece todas as *interfaces*, o que permite conversar entre diferentes algoritmos. Os testes e avaliações das implementações são realizados pela classe *EvaluationResult*, no qual é baseado no modelo de avaliação do TREC com resultados em gráficos e relatórios.

O Luppar está desenvolvido na linguagem de programação Python 3.6[2] e utiliza o *framework Web* Django 10.11 [3] para aplicação cliente-servidor. O banco de dados utilizado é o MySQL 5.7 [4] onde são armazenados os metadados do projeto conforme diagrama da Figura 11. Segue a Tabela com as bibliotecas utilizadas no projeto:

O projeto tem código fonte aberto e está disponibilizado no *github.com* [5].

4.2 RECUPERAÇÃO E RANQUEAMENTO

O método de recuperação em duas etapas propostas e implementado em Luppar faz uso de medidas de similaridade em dois momentos: na recuperação sem expansão inicial e na recuperação com a consulta expandida baseada nos documentos ranqueados no topo daqueles recuperados na primeira etapa. O ranqueamento é embutido ao processo e tem objetivo de

[2] https://www.python.org/
[3] https://www.djangoproject.com/
[4] https://www.mysql.com/
[5] https://github.com/fabianobie/luppar

Tabela 3 – Bibliotecas em Python utilizadas no projeto

Biblioteca	Finalidade	Link
Numpy	Programação matemática do projeto em álgebra linear.	http://www.numpy.org/
NLTK	Soluções que envolvem Processamento de Linguagem Natural na fase de pré-processamento: tokenização, radicalização, generalização, tagueamento, remoção de *stopword* e limpeza textual de forma geral.	https://www.nltk.org/
scikit-learn	Funções de Normalização (L1 e L2) dos dados. Métodos de agrupamento.	http://scikit-learn.org/
Gensim	Treinamento e geração do MSD. Implementação do Word2vec.	https://radimrehurek.com/gensim/
pdfminer	*Parsing* de texto em arquivos pdf	https://github.com/pdfminer/pdfminer.six

classificar os documentos na ordem dos mais relevantes em relação à consulta. Estes dois processos são possíveis devido aos modelos de RI.

Algoritmo 1: Pseudocódigo de Recuperação e Ranqueamento

Data: *Corpus D*, *raw_query*, *related_docs*, *params*

Result: List of documents *docs_result*

1: $query \leftarrow$ preprocessor(raw_query)
2: **if** *expand_query* **then**
3: $\quad query_m \leftarrow$ expand_query($query$)
4: $\quad query \leftarrow query_m$
5: **end if**
6: $documents \leftarrow$ boolean_search($query$, $params$)
7: $q_w \leftarrow$ tf_idf($query$)
8: **if** *related_docs* **then**
9: $\quad q_w \leftarrow$ rocchio_method(q_w,$related\_docs$)
10: **end if**
11: **for** $i \leftarrow 1$ **to** size($documents$) **do**
12: $\quad d_w \leftarrow D[documents[i]]$
13: $\quad scores[i] \leftarrow$simdq($d_w, q_w$)
14: **end for**
15: $docs\_result \leftarrow$ sort($scores, documents$)
16: **return** *docs_result*

O procedimento de recuperação do projeto segue o Algoritmo 1. É dada uma coleção D de documentos onde cada documento d_w está representado em um espaço vetorial de palavras de t termos, com os termos indexados formando um vocabulário $V = \{k_1, k_2, ...k_t\}$. Neste espaço,

cada documento é representado por um vetor de pesos das palavras $d_w = \{w_{1,i}, w_{2,i}..., w_{t,i}\}$. Seja *raw_query* o texto digitado pelo usuário em linguagem natural. Na linha 1, este passa pela etapa de preprocessamento (*preprocessor*), para tokenização, limpeza e filtragem (somente substantivos, verbos e adjetivos), a fim da representação *bag-of-word*. Agora a opção de expansão de consulta pode ser acionada (linha 3). Seguindo na linha 6, com a lista de termos da consulta é aplicado o Modelo Booleano, que por padrão está configurado na Forma Normal Disjuntiva (DNF), de modo a obter os documentos que estão associados q. Isso quer dizer que para cada termo da busca seja-a expandida ou não é reescrita a consulta na forma $q = t_1 \vee t_2 \vee ...t_n$. Com essa representação e o índice invertido consegue-se todos os documentos relacionados, mas sem nenhuma ordenação ou classificação.

Na linha 7, representa-se a consulta no mesmo espaço que os documentos recuperados através do tf_idf. Caso a consulta não tenha expansão, apenas o número de ocorrências dos termos é levado em consideração no tf_idf (vide equação 2.3), caso contrário a ponderação obtida na expansão é utilizado. A consulta q_w representada como um pseudo documento $q_w = \{w_{1,0}, w_{2,0}..., w_{t,0}\}$ sendo w_t o peso associado ao termo na consulta.

Agora com a consulta q_w e os documentos d_w recuperados é através da relação de similaridade $simdq(d_w, q_w)$, que vai depender do modelo de RI escolhido, pondera a relevância dos documentos. Os modelos quantificam a relevância dos documentos para com a consulta e assim trazem ordenados os resultados, ou seja, o ranqueamento (linha 8 a 12). O Luppar contêm os três modelos clássicos: o Modelo *Booleano*, Modelo de Vetorial e o Probabilístico. Para o Booleano (equação 2.4) o procedimento 1 encerra-se na linha 6. Dessa forma não há ranqueamento. Há opção, além da DNF, a Forma Normal Conjuntiva (CNF). Para o Vetorial foi utilizado a fundamentação algorítmica descrita em Baeza e Salton (BAEZA-YATES; RIBEIRO-NETO, 2013; SALTON; WONG; YANG, 1975) (equação 2.5) e para o modelo probabilístico foi utilizado a implementação do *Okapi BM25* (RUSSELL; NORVIG, 2010; ROBERTSON; ZARAGOZA *et al.*, 2009). Durante os cálculos de similaridade o sistema se utiliza dos índices, modelos e matrizes calculados *off-line*. Os parâmetros utilizados são os mesmos recomendados na literatura citada.

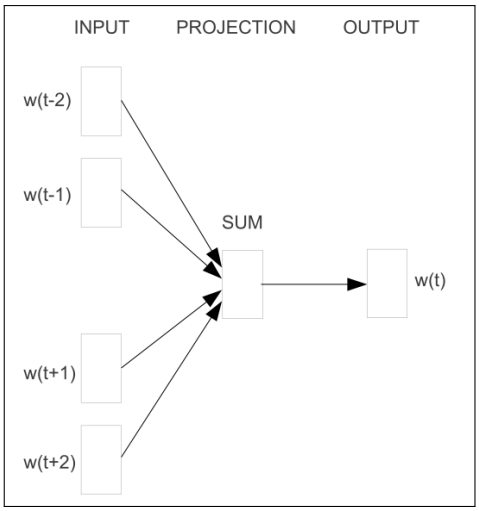
4.3 CONSTRUÇÃO DO TESAURO MSD

Dentre os processos *off-line*, destaca-se a construção do tesauro por Modelo Semântico Distribucional. Nesta pesquisa utilizou-se a abordagem preditiva com *Word embedding*,

especificamente a *Word2Vec* (MIKOLOV *et al.*, 2013). Para cada *corpus* do projeto se executa duas etapas.

A primeira etapa nessa construção foi definir as palavras alvo e os contextos em que estas aparecem. Os termos alvos são substantivos, adjetivos, advérbios e verbos, que são então filtrados. Já para os contextos, utilizou-se a relação de proximidade entre palavras na mesma sentença, ou seja, o conjunto de termos no entorno do termo alvo. Por exemplo, a seguinte sentença da coleção MED: "*methods for experimental production of and known causes* ***ofhydrocephalus*** *in animals and humans*". Seja "*ofhydrocephalus*"a palavra alvo e as palavras sublinhadas formam os contextos com janela $w = 2$.

Figura 14 – Arquitetura CBOW



A arquitetura CBOW prevê a palavra alvo $w(t)$ com base nos contextos $w(t-i)$ (MIKOLOV *et al.*, 2013)

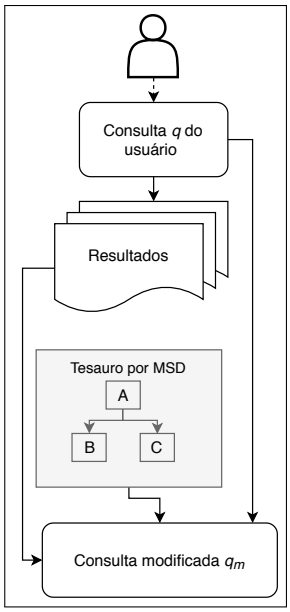
A segunda etapa é treinar uma rede neural de memória para calcular a distribuição de probabilidade sobre todas as palavras do vocabulário (MIKOLOV *et al.*, 2013) com arquitetura *CBOW*, conforme Figura 14. A saída é um modelo com vetores para cada palavra do dicionário. Nestas etapas utilizou-se a ferramenta*Word2vec* (vide Tabela 3) com arquitetura *CBOW*, janela de contexto $w = 5$ e a frequência mínima para os termos igual 5.

Com o modelo construído pode-se estabelecer a distância semântica (similaridade) entre quaisquer palavras do vocabulário através do cálculo do cosseno entre vetores de contexto do modelo. Quanto menor for a distância entre estes vetores no espaço, maior a similaridade. O tesauro então possui para cada palavra uma lista de palavras próximas ou as calcula quando necessário.

4.4 EXPANSÃO DE CONSULTA COM ACL E MSD

O método geral para expansão da consulta é representado na Figura 15. Consiste em construir e usar um dicionário de sinônimos (tesauro) para adicionar novos termos, atribuir pesos ou recalculá-los para modificar a representação original. O processo de EC consiste em utilizar os termos de q para encontrar novos termos, sem a participação do usuário, que venham resolver problemas de desambiguação e que melhor discriminem os documentos. Dado inicialmente q que passará a ter novos termos e será denominada q'. Os novos termos compõem w_j em q'. Agora a similaridade é calculada com $sim(d,q')$. A ideia é que a escolha dos novos termos seja realizada através de um tesauro. Um tesauro global construído automaticamente utilizando Modelo de Semântica Distribucional com representação *Word Embbeding* (MIKOLOV *et al.*, 2013).

Figura 15 – Diagrama de fluxo da expansão automática de consulta.



Em continuidade ao trabalho de Xu e Croft (XU; CROFT, 2000) a abordagem construída utiliza Análise de Contexto Local (ACL) combinada com Modelo de Semântica Distribucional. Esta técnica propõe usar os primeiros resultados de uma consulta para construir uma representação por coocorrência de conceitos (grupos de substantivos) e por similaridade destes com a consulta encontrar termos candidatos a serem agregados à expansão consulta, isto é, combinar análise local e global para EC. Isso é mostrado no Algoritmo 2.

Algoritmo 2: Pseudocódigo de EC proposto com ACL e MSD

Data: query q, thesaurus th

Result: modified query q_m

1: $documents \leftarrow$ search_top_ranked(q);
2: **for each** $documents$ **do**
3: $passages \leftarrow$ window($document$);
4: **for each** $passages$ **do**
5: $concepts \leftarrow$ find_concepts_in_context($passage$);
6: **end for**
7: **end for**
8: sort($concepts$);
9: **for** $i \leftarrow 1$ **to** N **do**
10: $m[i] \leftarrow$simqc($q, concepts[i], th$);
11: **end for**
12: sort(m);
13: $q_m \leftarrow q + m[1..n]$;

O método de expansão proposto e implementado em Luppar faz uso de medidas de similaridade em dois momentos no Algoritmo 2: na recuperação sem expansão inicial e na recuperação com a consulta expandida baseada nos documentos ranqueados no topo daqueles recuperados na primeira etapa.

Na linha 1, recupera-se com a consulta original os documentos mais bem ranqueados. Nesse momento a consulta possui a mesma representação dos documentos. O VSM (SALTON; WONG; YANG, 1975) foi implementado e utilizado para recuperar, ranquear e selecionar os documentos de D para o topo da consulta.

Já com os documentos do topo da consulta, nas linhas 2 e 8, recupera-se as n passagens mais bem ranqueadas usando a consulta original. Isto é conseguido quebrando os documentos inicialmente recuperado pela consulta em *passagens* (sentenças) e classificando as passagens como se fossem documentos. Nas linhas 9 a 11, para cada conceito (grupo de substantivos) nas passagens do topo dos resultados, calcula-se a similaridade $sim(q,c)$ entre toda a consulta q (e não os termos individuais da consulta) e o conceito, usando uma variante do TF-IDF. Nas linhas 12 e 13, finalmente, os m conceitos mais bem ranqueados, de acordo com $sim(q,c)$, são adicionados à consulta original q. Para cada conceito adicionado atribui-se um

peso dado por $1-0,9xi/m$, onde i é a posição do conceito no ranking de conceitos. Os termos na consulta original q podem ser enfatizados através da atribuição de um peso igual a 2 para cada um deles.

Embora a estrutura do Algoritmo 2 seja semelhante à de um ACL padrão, sua implementação é significativamente diferente. Difere em dois pontos: no conceito de janela de contexto e no cálculo da similaridade.

A Análise de Contexto Local (linha 3) usa a noção de passagem. Uma passagem é uma sentença fechada através de um sinal de pontuação. Isso resulta em uma janela de contexto de tamanho variável dependente das declarações presentes no documento que está sendo analisado, ao contrário da janela de tamanho fixo usada no ACL padrão.

O segundo ponto no qual esse algoritmo difere do ACL padrão está no cálculo da similaridade entre termos e conceitos da consulta ($simqc(q,c,th)$, linha 10). O Algoritmo 2 recebe como um de seus insumos um tesauro distribucional previamente calculado que é considerado. Quando se trata de uma coleção fechada de documentos, o tesauro é calculado para a coleção. Já em aplicações web, este tesauro pode ser global. O cálculo da similaridade é dado pela equação 4.2:

$$simqc(q,c,th) = \prod_{k_i \in q}^{c} (\delta + \frac{\log(f(c,k_i,th) \times IDF_c)}{\log n})^{IDF_i}, \tag{4.2}$$

Seja k_i corresponde a cada termo de q. IDF_i e IDF_c são o inverso da frequência sobre termo da consulta i e sobre o conceito c respectivamente. Nesta equação, δ é uma pequena constante ($0,1$ em (XU; CROFT, 1996)) para evitar zeramento da expressão em alguns casos, $f(c,k_i,th)$ é uma função que quantifica a correlação entre um conceito c e um termo da consulta considerando a distribuição no tesauro MSD th:

$$f(c,k_i,th) = word2vec.cos(th[c],th[k_i]) \tag{4.3}$$

No qual, $word2vec.cos$ é a função do *Word2vec* utilizada para medir a similaridade semântica através do coseno do vetor gerado pelo MSD. Em seguida são ranqueados os conceitos que estão mais próximos da consulta na totalidade. Os m melhores conceitos são escolhidos para serem quantificados segundo sua importância. No trabalho (ERMAKOVA; MOTHE, 2016), Ermakorva propõem penalizar os candidatos em vários aspectos (IDF, *score*, importância e POS) enquanto essa abordagem manteve a penalizar com peso 2 para as palavras da consulta original

e pesar os m conceitos com $1 - 0.9xi/m$ (XU; CROFT, 1996) e também com o IDF, tendo em vista que os outros *scores* pouco alteraram significativamente a precisão.

4.5 COLETA DE DOCUMENTOS

Uma etapa fundamental deste trabalho foi obter as coleções de documentos que seguem para o SRI. O sistema se propõe a executar a principal e mais importante tarefa de RI, a tarefa *Ad hoc*, na qual os algoritmos inclusos no sistema recebem requisições de informações e as executam sobre uma coleção de documentos predeterminada.

O projeto admite gerenciar múltiplos *corpus* de documentos. Todas as coleções são formadas por três arquivos. Um arquivo com os documentos, o segundo com as consultas e o terceiro a identificação da consulta e quais documentos foram julgados relevantes pela consulta. Estes documentos são entradas para os algoritmos propostos e sua execução produziu um quarto documento com os julgamentos automático de arquivos recuperados.

Para o Luppar foram selecionadas três bases de referência em inglês e uma originalmente em português. As três coleções de teste em inglês foram elaboradas por Ed Fox na *Virginia Polytechnic Institute and State University* (ONE, 1990). A última em português foi desenvolvida e proposta por esta pesquisa. A Tabela 4 mostra as características dessas coleções. Todas seguindo o paradigma *Cranfield*.

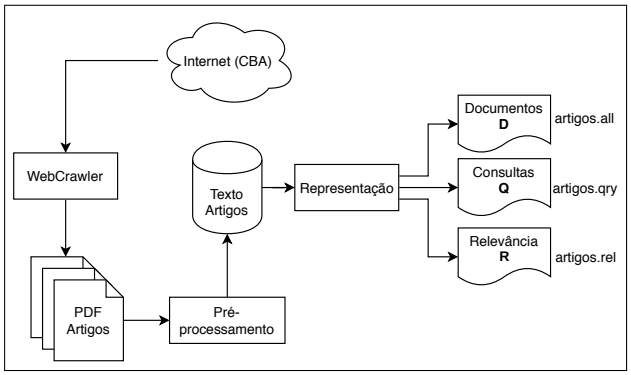
Tabela 4 – Coleções de Referência

Coleção	Assunto	Idioma	Matriz ($termos \times docs.$)	Núm. de Consulta
MED	Medicina	EN	7876 x 1033	30
LISA	Biblioteconomia	EN	11710 x 6004	35
NPL	Eng. Elétrica	EN	7861 x 11429	100
CBA	Congresso de Brasileiro de Automática	PT-BR	98081 x 2317	50

4.5.1 Coleção ARTIGOS

A coleção própria deste trabalho é formada por 2317 de 2681 artigos em português apresentados ao Congresso Brasileiro de Automática (CBA) entre os anos de 2010 a 2016. Estes estão disponíveis em *links* livre na internet. Os artigos estabelecem um domínio específico para coleção. Possuem texto livre, mas estruturalmente semelhantes por se tratar de artigos científicos. Todos os artigos estão no formato PDF. Nem todos os artigos foram selecionados para construção do *corpus*, pois deveriam atender o requisito de conter palavras-chave e ser em português.

Figura 16 – Processo automático de coleta e transformação dos artigos do CBA em um *corpus* para avaliação em RI.



A Figura 16 exibe o processo de coleta para construção da base de referência para avaliação em RI desta pesquisa. Todo o processo é automatizado através de *scripts*. Os artigos disponibilizados, no formato *PDF*, nos sites de eventos da CBA, são baixados através de rastreadores web (*Web Crawler*) e em seguida o pré-processamento é aplicado para extração de texto dos arquivos. Nessa fase a codificação de caracteres especiais (não ASCII) é ajustada e as palavras-chave são extraídas do texto. A próxima tarefa é representação da coleção de referência para o paradigma *Cranfield*. O resultado final é um arquivo D com a referência de todos os documentos, um arquivo Q com as consultas e um outro arquivo A com o julgamento de relevância ($R = Q \rightarrow D$).

O arquivo D é construído, de forma simples, atribuindo-se um identificador a cada documento da coleção para que faça referência quando necessário obter o texto correspondente. Já Q e R requerem um procedimento sistemático, pois o processo é automático e uma suposição básica do paradigma *Cranfield* é que os julgamentos de relevância são completos (CLEVERDON, 1991), ou seja, que cada documento é julgado por cada tópico (consulta) e estes são realizados por julgamento humano especializado.

Seja Q um conjunto de consultas q_i, i um identificador da consulta e k_{ij} os termos que formam q, seja j o identificador do documento. Cada q_i é elaborado e associado aos documentos segundo sua relevância para compor R. Esta pesquisa propõem a seguinte heurística para construir Q e R simultaneamente:

1. **Selecionar os candidatos** k_{ij}: para cada artigo selecionado existe um conjunto de palavras-chave associados. Estes são extraídos e divididos em termos k para formarem candidatos a compor uma consulta q_i. O resultado dessa etapa é a união de todos os k dos artigos, o vocabulário K.

2. **Representação das palavras-chave**: com todos os k cria se uma matriz TF-IDF $M = KxD$.
3. **Combinar os k_{ij}**: o objetivo dessa etapa é selecionar e combinar os n candidatos a k_{ij} de forma compor $q_i = \{k_{ij_1}, k_{ij_2}, \ldots, k_{ij_n}\}$. O critério de seleção escolhido foi utilizar os agrupamentos (*cluster*) gerados através do processo de Indexação Semântica Latente (LSI) (DEERWESTER, 1988). Técnica que geralmente é utilizada para categorização de texto. Neste caso as palavras-chave seriam as categorias. O LSI consegue agrupar os k que são próximos por contextos (hipótese distribucional (HARRIS, 1954)) e consequentemente por documentos. O resultado dessa etapa é formação de grupos de palavras-chave (*cluster*) que atendem a um maior número de documentos semelhantes, ou seja, a própria consulta q_i.
4. **Compor julgamento de relevância ($R = Q \rightarrow D$)**: para cada consulta formada q_i existem n palavras-chave k_{ij} e para cada existem documentos de D associados. A união desses conjuntos formam nosso conjunto R.
5. **Ordenar o julgamento de relevância**: Neste passo já sabe-se que R possue todos os documentos relevantes, porém a ordem é desconhecida. O critério utilizado para classificar é contabilizar os documentos que abrangem um maior número de palavras-chave. Isso pode ser obtido somando os n vetores de $kij \in q_i$ da consulta e ordenar forma decrescente. Os maiores j equivalem os documentos mais relevante a q.

Para está pesquisa um total de 50 consultas foram geradas uniformemente com n variando de 3 e 8 palavras-chave. O critério de seleção para composição está fortemente ligado ao baixo grau de disjunção exigido para ser coerente uma consulta. Abordagens como a combinação aleatória ou agrupamentos por vizinho mais próximo (*KNN*) foram utilizadas, mas não superaram o LSI nesse critério de disjunção dos documentos. Apesar de a classificação do julgamento ser automática não desvia do paradigma *Cranfield* já que as palavras-chave são escolhidas por especialistas (autores dos artigos) e estas compõem a intenção de informação associados aos documentos.

Usar coleções de referência permite comparar diferentes SRI e suas variantes nas funções de ranqueamento. Além de garantir repetibilidade dos experimentos. É importante notar que as bases seguem formatos diferentes de acordo com necessidade de avaliação. Para este trabalho a eficácia da consulta (*recall*) e como consequência um ranqueamento efetivo (*precision*) é suficiente, por isso, a escolha das bases com foco na relevância e as coleções de pequeno porte.

5 RESULTADOS E DISCUSSÃO

O Luppar é um SRI completo composto por um conjunto de algoritmos que trabalham juntos para solucionar o problema de recuperação de informação de forma a trazer os documentos que sejam relevantes e com uma boa precisão. Esses algoritmos foram experimentados em diversas combinações com objetivo compreender os efeitos dessas combinações com seus parâmetros e representações. Algumas das implementações são esforços de trabalhos anteriores e outros como contribuição desta pesquisa. Este capítulo avalia a capacidade do SRI Luppar em recuperar documentos e em especial sua contribuição em Análise de Contexto Local baseada em Modelo Semântico Distribucional (ACL-MSD) discutida no capítulo anterior e a coleção de referência em português, denominada de ARTIGOS, construída de forma semi-automática. Os dados, os parâmetros utilizados, resultados quantitativos e uma análise dos procedimentos serão apresentados.

5.1 DADOS E MÉTRICAS DE AVALIAÇÃO

A avaliação fez uso de quatro conjuntos de dados. Três coleções são no idioma inglês e uma em português. As três coleções teste, em inglês, já comumente usadas como referência para avaliação em recuperação de informação, foram elaboradas por Ed Fox na *Virginia Polytechnic Institute and State University* (ONE, 1990). A quarta, em português, foi construída sobre os artigos publicados no Congresso Brasileiro de Automática entre os anos de 2010 a 2016. A Tabela 5 mostra as características dessas coleções.

Coleção	Assunto	Idioma	Núm. de Termos	Núm. de Documentos	Núm. de Tópicos	Tamanho Documento (termos)	Tamanho Tópico. (termos)	Tamanho Tóp. x Doc (%)
MED	Medicina	Inglês	9622	1033	30	167.2	23.8	14.2
LISA	Biblioteconomia	Inglês	13706	6004	35	97.5	66.1	67.7
NPL	Eng. Elétrica	Inglês	7861	11429	100	41.9	10.9	26
CBA	Congresso Brasileiro de Automática (CBA)	Português	98081	2317	50	5092.63	7.1	0.14

Tabela 5 – Coleções de referência utilizadas nos testes e avaliação do Luppar

Todas as coleções são formadas por três arquivos. Um arquivo com os documentos, o segundo com as consultas e o terceiro a identificação da consulta e quais documentos foram julgados relevantes pela consulta. Estes documentos são entradas para os algoritmos implementados e sua execução produziu um quarto documento com o julgamento automático de arquivos recuperados (conjunto resposta). Este último arquivo comparado com o de julgamento permite

calcular as métricas tradicionais de RI e comparar o par necessidade de informação-documento e medir a eficácia do SRI.

Para avaliar o Luppar fez-se necessário utilizar todas as coleções de teste e submetê-las a tarefa *ad hoc* de RI, ou seja, a busca do par informação-documento. Para cada coleção o conjunto de consultas (tópicos) são submetidas ao SRI e então coletados os resultados para em seguida aplicar as métricas de avaliação da recuperação. Dessa forma, especificamente avaliou-se o desempenho da expansão de consulta ACL utilizando MSD. Três métodos de trabalhos relacionados foram implementados para comparação. A expansão de consulta utilizando a *WordNet* (MILLER, 1995; LI; GANGULY; JONES, 2016), isto é, um tesauro externo global, a própria ACL com características locais (XU; CROFT, 1996; XU; CROFT, 2000), mas sem empregar o MSD e a consulta original, sem expansão, tomada como *baseline*.

As métricas utilizadas para avaliação são as mesmas da conferência TREC (HASHEMI *et al.*, 2016). Inclusive, o mesmo software chamado de *trec_eval* em sua última versão 9.0 [1]. Das onze métricas produzidas, quatro foram escolhidas de modo a quantificar a qualidade do SRI e seus algoritmos. Seguem as métricas:

- **MAP:** precisão média sobre todas as consultas. Esta métrica sumariza o quanto é assertivo o algoritmo em recuperar os documentos.
- **Bpref:** calcula uma relação de preferência, ou seja, se os documentos julgados relevantes são recuperados antes daqueles julgados irrelevantes.
- **Reciprocal Rank:** precisão média em relação aos primeiros resultados. Esta métrica avalia o quanto os resultados corretos estejam no topo do ranking.
- **A curva Recall-Precision:** 11 pontos interpolando a precisão média (em 0%, 10%,..., 100% do *recall*) que permite desenhar a curva de cobertura e precisão permitindo perceber que à medida que os documentos mais relevantes são recuperados (o *recall* aumenta) e enquanto documentos irrelevantes são recuperados (a precisão diminui).

As equações das métricas supracitadas são detalhadas na seção 2.5. Na próxima seção os resultados exibidos correspondem as métricas de análise em cada coleção inteira da Tabela 5.

[1] https://github.com/usnistgov/trec_eval

Tabela 6 – Parametrização dos algoritmos durante os experimentos

Algoritmo	Função	Parâmetros
Preprocessor	Pré-processamento e representação	use_stop_words=True Selecionar palavras (Substantivos, verbos, advérbio e adjetivo): reduce=True Representação: R = TF.IDF Normalização: norm='L1' Radicalização: Porter Stemmer (Inglês) e RSLP Stemmer (Português)
VSM	Recuperação e Ranqueamento	Seleção dos documento: mode='OR' Representação da consulta: R = TF
BM25	Recuperação e Ranqueamento	Seleção dos documento: mode='OR', Configuração do BM25: k1=2.0, k3=1.0, b=0.75 Representação da consulta: R = TF.IDF
WordNet	Expansão de consulta	Linguagem = Português e inglês Quantidade palavras: limit = 2 por termo
ACL	Expansão de consulta	Janela de palavras para passagens: W=300 Quantidade de documentos do topo do ranking: N=10 Número de passagens: 50 Quantidade de conceitos selecionados: m=5
ACL-DSM	Expansão de consulta	Janela de palavras para passagens: W=Selecionado em tempo de execução Quantidade de documentos do topo do ranking: N=10 Número de passagens: 50 Quantidade de conceitos selecionados: m=5
word2vec	Gerar tesauro MSD	Janela de palavras: W=5 Tamanho do vetor: size=300 Algoritmo de treinamento: CBOW sg=0 Quantidade mínima de palavra: min_count=2 Quantidade de *threads*: workers=4

5.2 RESULTADOS

Os ensaios ocorreram em sequência combinando 4 fontes de dados (MED, LISA, NPL e ARTIGOS), 2 modelos de busca (VSM e BM25), 4 opções de expansão de consulta (*Baseline*, *Wordnet*, ACL e ACL-MSD) e 215 consultas, o que totaliza 1.720 execuções com gráficos e métricas calculados.

A Tabela 6 apresenta os parâmetros utilizados nos diferentes algoritmos implementados em cada execução para esta pesquisa. A parametrização foi escolhida baseado na literatura e com possibilidades exequíveis para as coleções.

Os ensaios foram realizados em um sistema operacional MacOSX (*El Capitan*) em um notebook MacBook Pro (13-inch, Mid 2012) com processador *Intel Core* i5 de 2,5 GHz com memória RAM de 8 GB 1333 MHz DDR3. Todos os testes foram implementados na linguagem *Python* 3.6 e utilizaram as bibliotecas da Tabela 3.

As Tabelas 7, 9, 8 e 10 apresentam os resultados dos experimentos segundo a análise do *trec_eval*. Cada tabela mostra quatro colunas de resultados: *Baseline*, *WordNet*, ACL, e

AC-MSD. A coluna *baseline* refere-se à recuperação de informações por uma consulta sem expansão. Os resultados na coluna *WordNet* referem-se a expansão de consulta com o tesauro *WordNet* (LI; GANGULY; JONES, 2016). A coluna ACL registra os resultados onde a expansão da consulta é baseada na Análise de Contexto Local (XU; CROFT, 2000). Finalmente, a coluna ACL-MSD aplica a representação semântica distribucional (*Word Embedding*) combinada com Análise de Contexto Local para construir um dicionário de sinônimos e usa este na expansão da consulta. Todos obtidos com modelos VSM e BM25.

Tabela 7 – Resultados para coleção MED

Modelos	Métricas	*Baseline*	*WordNet*	ACL	ACL-MSD
VSM	map	0,5142	0,4949	0,5255	0,5348
	bpref	0,8985	0,9318	0,9418	0,9406
	recip_rank	0,8537	0,7726	0,8889	0,8889
BM25	map	0,5033	0,4873	0,5262	0,5459
	bpref	0,8985	0,9318	0,9660	0,9712
	recip_rank	0,8992	0,8294	0,8253	0,8944

Tabela 8 – Resultados para coleção LISA

Modelos	Métricas	*Baseline*	*WordNet*	ACL	ACL-MSD
VSM	map	0,2641	0,2034	0,2475	0,2602
	bpref	0,9981	1,0	1,0	0,9981
	recip_rank	0,5184	0,4618	0,5006	0,5038
BM25	map	0,3495	0,2520	0,3577	0,3627
	bpref	0,9981	1,0	1,0	0,9981
	recip_rank	0,6459	0.5085	0,6400	0,6693

Tabela 9 – Resultados para coleção NPL

Modelos	Métricas	*Baseline*	*WordNet*	ACL	ACL-MSD
VSM	map	0,1886	0,1428	0,1968	0,2282
	bpref	0,9767	0,9886	0,9878	0,9333
	recip_rank	0,4437	0,3583	0,5014	0,4267
BM25	map	0,2124	0,1756	0,2640	0,2580
	bpref	0,9766	0,9819	0,9891	0,9815
	recip_rank	0,5987	0,5432	0,6142	0,6373

Note que os resultados para ACL-MSD são consistentemente superiores para as quatro bases, para os dois modelos e competitivo nos demais indicies de desempenho. O resultado da métrica *map* demonstra que o sistema alcança melhoria na qualidade da busca de forma geral com a expansão da consulta. Em comparação ao *baseline*, o LISA, é um único que não teve

Tabela 10 – Resultados para coleção ARTIGOS

Modelos	Métricas	*Baseline*	*WordNet*	ACL	ACL-MSD
VSM	map	0,2985	0,2778	0,3014	0,3111
	bpref	0,8011	0,9872	0,9264	0,9982
	recip_rank	0,7573	0,7046	0,8198	0,8273
BM25	map	0,3024	0,1756	0,2928	0,3170
	bpref	0,8011	0,9819	0,9264	0,9988
	recip_rank	0,8230	0,5432	0,8087	0,7862

Figura 17 – *Precision x Recall* para coleção MED

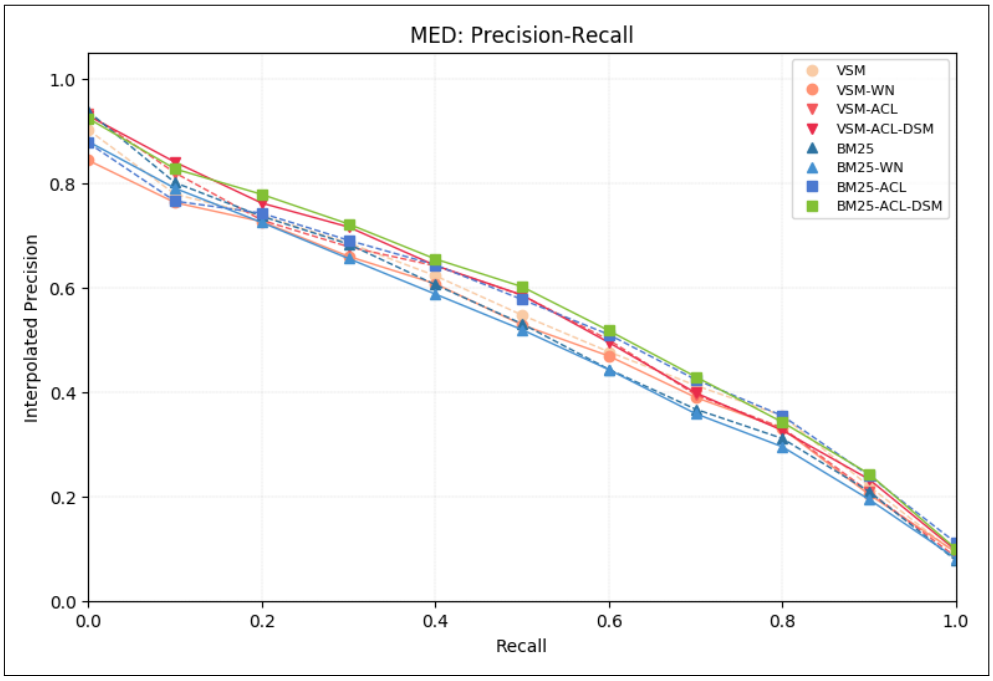


Tabela 11 – Desempenho da Análise do Contexto Local com MSD na coleção ARTIGOS

	Precision - 50 Tópicos de ARTIGOS					
Modelo	**VSM**			**BM25**		
Recall	ACL	ACL-MSD	Ganho(%)	ACL	ACL-MSD	Ganho(%)
0	87,49	86,99	-0,57	85,61	84,88	-0,85
10	56,63	56,82	+0,33	57,35	61,27	+6,83
20	47,55	48,02	+0,98	46,70	50,20	+7,49
30	40,48	41,67	+2,93	40,16	44,16	+9,96
40	34,82	36,45	+4,68	32,74	37,04	+13,13
50	28,94	29,46	+1,79	27,77	30,59	+10,15
60	22,85	23,89	+4,55	22,89	24,53	+7,16
70	17,45	18,56	+6,36	16,76	17,48	+4,29
80	12,27	13,65	+11,24	11,45	12,55	+9,60
90	7,55	8,55	+13,24	7,26	7,77	+7,02
100	1,35	4,19	+210,37	1,28	4,18	+226,56
Média	32,49	33,48	**+4,5514**	31,82	34,06	**+7,4946**

Figura 18 – *Precision x Recall* para coleção NPL

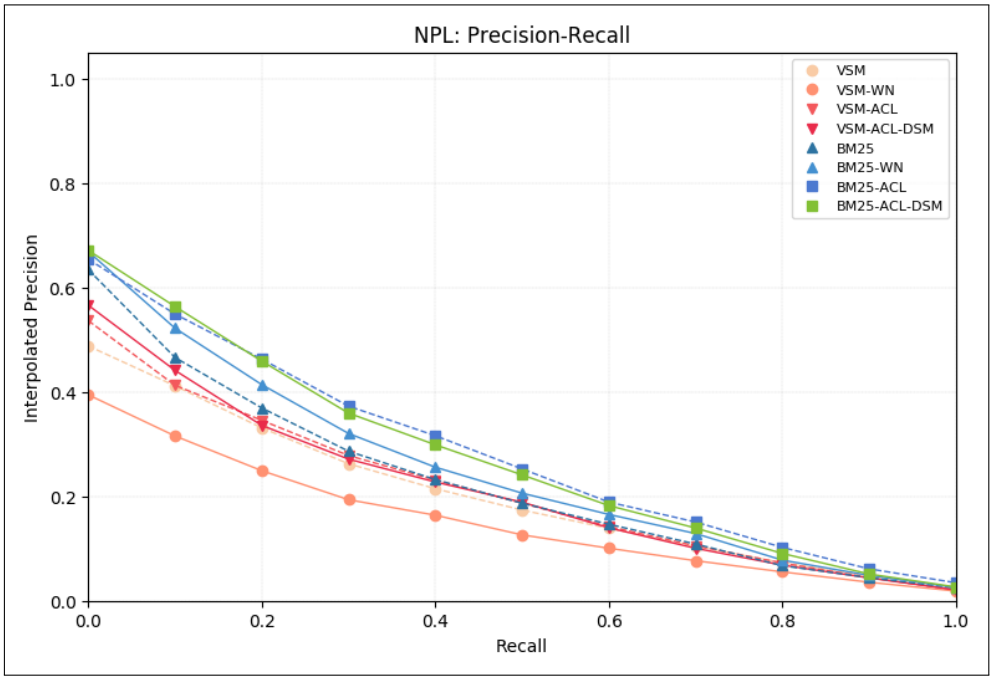


Figura 19 – *Precision x Recall* para coleção LISA

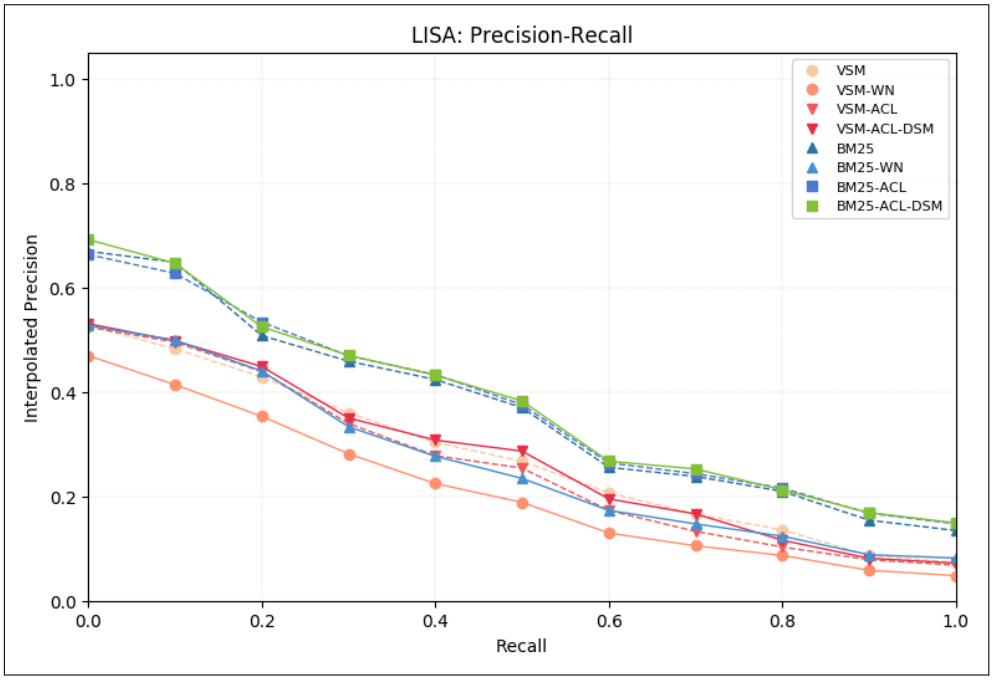


Figura 20 – *Precision x Recall* para coleção ARTIGOS

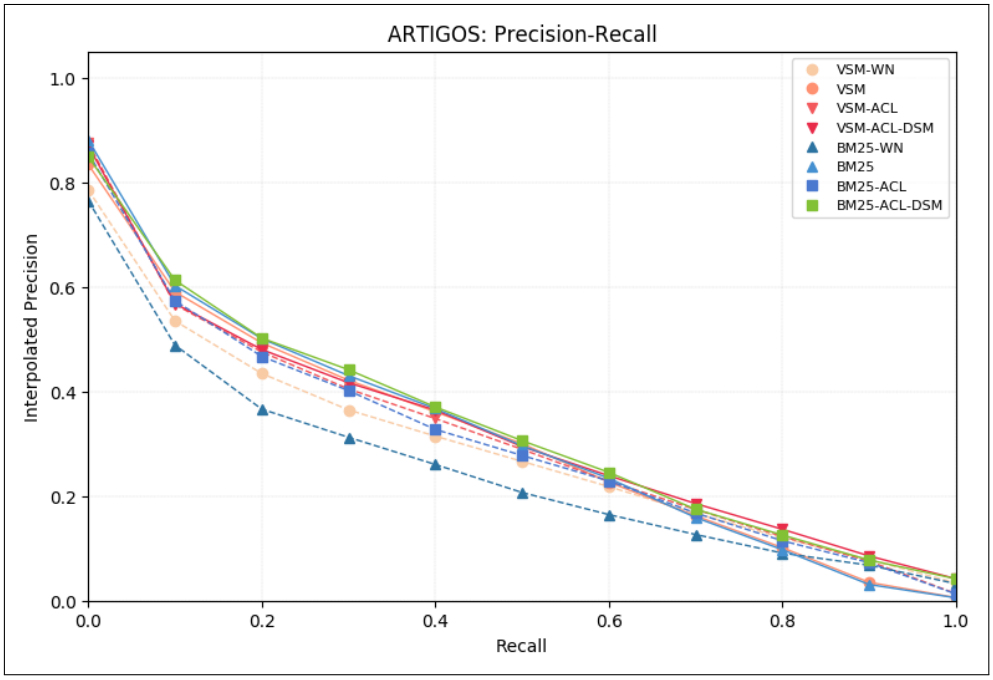
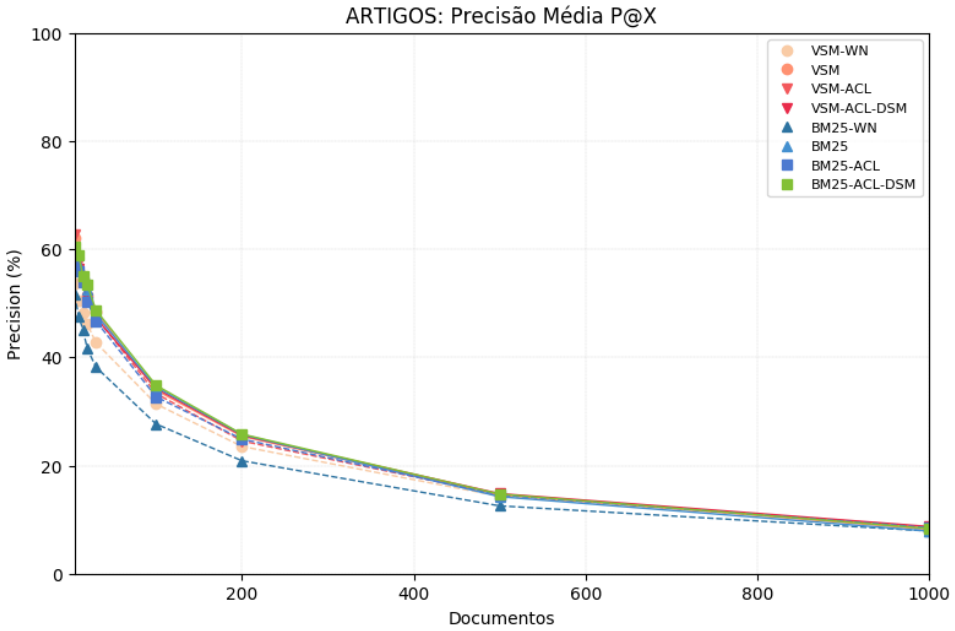


Figura 21 – Precisão P@n para coleção ARTIGOS



ganhos na recuperação com expansão de consulta e isto é explicado pelo fato do tamanho dos tópicos serem 67% do tamanho dos documentos (vide Tabela 5) o que dificilmente novos termos conseguirão especificar ainda mais a consulta.

As curvas *recall-precision* nas Figuras 17, 18, 19, e 20, uma das principais métricas de comparação de desempenho por apresentarem uma comparação não pontual, confirmam esses resultados. Note ainda que *map* é uma medida aproximada da área sob a curva *recall-precision* (RP-AUC) não interpolada e confirmam essas conclusões.

Tabela 12 – Alguns exemplos de consultas e os respectivos termos resultantes da EC

Modelo VSM			Termos Expandidos		
Coleção	Consulta	Termos	WordNet	ACL	ACL-MSD
MED	infantile autism	infantil, autism		twin, subtest, receptor, item, nosolog	childhood, communiti, scale, children, school
MED	neoplasm immunology	neoplasm, immunolog	tumor, tumour	polyoma, mice, tumour, thymectomi, syngen	pathogenesi, immun, syngen, induct, polyoma
MED	blood or urinary steroids in human breast or prostatic neoplasms	blood, urinari, steroid, human, breast, prostat, neoplasm	chest, man, homo, steroid, human_be, tumor	cancer, examin, patient, activ, observ	pregnanediol, benign, postmenopaus, estrogen, estradiol
ARTIGOS	transformada wavelet, controlador pi, grasp	transformada_wavelet, controlador_pi ,grasp		janel, deviation, ressalta-s,mva, top'	heurs, metaheurs, meta-heurs, metaheuris, heuris
ARTIGOS	controlador pid, controle digital, controle por modos deslizantes, controle fuzzy	control_pid, control_digit, control_por_mod_desliz, control_fuzzy	inspeca, limit, verificaca	tca, pawlak, mod_desliz, logic_fuzzy, superfici	sistem_de_control, control_robust, projet_de_control, lqr, pid
ARTIGOS	processamento de imagens, sistemas não lineares, lmis	red_intelig, diagnost_de_falh, process_de_imag, robo_movel	insucess, process, robo	monitor, mund, cenari, comunicaca, destin	sistem_intelig, visa_computac, reconhec_de_padro, classificaca_de_padro, inov

A Tabela 11 exibe a comparação dos resultados entre ACL e ACL-MSD obtidos com a recuperação nos modelos VSM e BM25. O resultado está organizado com o *precision* interpolado para os 11 níveis padrão de *recall* para a média das 50 consultas da coleção ARTIGOS. Na última linha é exibido a média das médias. A coluna ganho exibe em porcentagem o ganho de *precision* do ACL-MSD em relação ao ACL. Os valores são exibidos de forma gráfica na Figura 20.

5.3 DISCUSSÃO

As métricas utilizadas nessa pesquisa avaliam a qualidade dos resultados e não o desempenho do SRI, como, por exemplo, tempo de processamento das consultas. O Luppar, por possuir caráter científico e didático, exibe essas métricas na tela de resultados da busca conforme Figura 22.

É válido observar algumas características das bases durante a expansão de consulta conforme Tabela 5 e os resultados. Os dados MED são documentos curtos, mas com consultas de precisão alta mesmo sem expansão de consultas e com tópicos curtos. O LISA são documentos com tamanhos curtos, com precisão baixa e com consultas longas que vão de oposto a necessidade de expandir consulta. O NPL com tamanho curto dos documentos, mas com uma quantidade bem maior de documentos, com precisão mediana e com muitas consultas variando entre longas e

Figura 22 – Tela de resultados com métricas exibida pelo Luppar depois de uma consulta.

curtas. Os ARTIGOS já são documentos longos e com consultas curtas em relação os documentos. Nestes quatro cenários verificamos ACL-MSD produz melhores termos extras para consulta.

Observe alguns exemplos dos resultados da expansão de consulta na Tabela 12 onde o fato de utilizar um dicionário global, como *WordNet*, em coleções fechadas (como é o caso do Luppar) muitas vezes não superou o *baseline* nos gráficos e métricas, pois, os sinônimos selecionados foram genéricos para a coleção ou os termos da consulta é desconhecido pelo dicionário. No primeiro caso o resultado se torna inferior ao *baseline*, como indicado nas Figuras 19 e 18. Reafirmando que a busca contextual local é melhor nestes casos.

Note que o ACL em seu funcionamento clássico foi um eficiente método de expansão de consulta. Como já demostrado na literatura por utilizar o melhor da análise local sobre os resultados do topo, na qual assume-se serem os melhores resultados relacionados, e análise global utilizando-se do conceito de contexto sobre o conjunto local. Nos gráficos de *precision - recall* evidênciamos que ACL-MSD segue os passos da cobertura do ACL, mas consegue ser melhor na precisão em vários pontos. O ponto fundamental dessa mudança na abordagem é a seleção realizada por similaridade de um tesauro predito MSD.

Finalmente, analisando os resultados do processo de recuperação sem e com expan-

são para coleção ARTIGOS, verificamos que suas consultas geradas de forma semi-automática estão de forma equilibrada associadas a precisão do processo de recuperação, pois conforme Figura 21, os documentos relevantes estão entre os primeiros resultados e com até 60% de precisão o que possibilita que novos algoritmos possam utilizar essa coleção como referência em português para recuperação e expansão de consulta.

Em uma comparação apenas entre ACL e ACL-MSD para com a coleção ARTIGOS, utilizou-se o teste estatístico *student t-test*. Como percebido na Tabela 11, as duas abordagens possuem comportamento semelhante, porém, ao analisarmos a taxa de melhoria em relação ao ACL é de 4,5% e 7,5% na recuperação VSM e BM25 respectivamente e estatisticamente significativa, pois *p-value* de 0,001, que seria significativo em um nível de $\alpha = 0,05$. Portanto, para esses dados, o teste t nos permite rejeitar a hipótese nula e concluir que o algoritmo ACL-MSD foi mais eficaz que ACL clássico.

6 CONCLUSÃO

Luppar é um Sistema de Recuperação de Informação (SRI) concebido e implementado com vista ao uso corporativo, ou seja, para coleções fechadas de documentos (não para web). A aplicação tira proveito disso incluindo um tesauro semântico baseado em *word embedding* construído apenas com os documentos das coleções alvos. Essa abordagem evita que consultas sejam expandidas com termos que, embora sejam significativos na língua, são inexistentes nas coleções em foco. *Word embedding* restrito ao corpus combinado com Análise de Contexto Local (ACL-MSD) completam a proposta em Luppar.

O trabalho utilizou critérios e métodos da conferência TREC para avaliar a proposta. Os resultados das Tabelas 7, 8, 9 e 10 mostram que os métodos de RI de Luppar são satisfatórios e compatíveis com o estado da arte. Os experimentos foram construídos de forma a revelar o ganho de eficácia da combinação ACL-MSD em relação a cada método individual, ACL ou MSD, aplicados isoladamente. O desempenho da consulta sem expansão foi incluída com método *baseline* para controle dos experimentos.

Além da abordagem ACL-MSD desenvolvida por Luppar outras contribuições são roteirizadas neste trabalho: o desenvolvimento de um SRI de forma estruturada e exequível. O código fonte é disponibilizado em seu sítio (*http://luppar.com*); Um *framework* de RI com arquitetura para expansão de novos métodos, idiomas e cálculos de métricas de eficácia; Uma metodologia de geração de coleções de teste para RI (documentos e consultas) a partir de artigos científicos, utilizando as palavras chaves como *ground truth*, foi desenvolvida e testada com resultados satisfatórios. O que gerou uma coleção de referência em português disponível para pesquisas futuras em RI.

Embora tenha amplo nicho de aplicação, a maior limitação de Luppar é a sua aplicação restrita a coleções fechadas de documentos. A principal linha de continuidade desta pesquisa é reescrever Luppar para Web. Também é necessário testar em coleções mais amplas. Luppar para busca na web é um projeto em andamento.

REFERÊNCIAS

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Recuperação de Informação - 2ed: Conceitos e Tecnologia das Máquinas de Busca**. [S.l.]: Bookman Editora, 2013. ISBN 9788582600498.

BARONI, M.; BERNARDI, R.; ZAMPARELLI, R. *et al*. Frege in space: A program for compositional distributional semantics. **Linguistic Issues in language technology**, USA, v. 9, p. 241–346, 2014.

BENGIO, Y.; DUCHARME, R.; VINCENT, P.; JAUVIN, C. A neural probabilistic language model. **Journal of machine learning research**, v. 3, n. Feb, p. 1137–1155, 2003.

BHOGAL, J.; MACFARLANE, A.; SMITH, P. A review of ontology based query expansion. **Information processing & management**, Elsevier, v. 43, n. 4, p. 866–886, 2007.

BUCKLEY, C.; SALTON, G.; ALLAN, J. The effect of adding relevance information in a relevance feedback environment. In: SPRINGER. **SIGIR'94**. [S.l.], 1994. p. 292–300.

CALLAN, J. P.; CROFT, W. B.; HARDING, S. M. The inquery retrieval system. In: SPRINGER. **Database and expert systems applications**. [S.l.], 1992. p. 78–83.

CARPINETO, C.; ROMANO, G. A survey of automatic query expansion in information retrieval. **ACM Computing Surveys (CSUR)**, ACM, v. 44, n. 1, p. 1, 2012.

CLEVERDON, C. The cranfield tests on index language devices. In: MCB UP LTD. **Aslib proceedings**. [S.l.], 1967. v. 19, n. 6, p. 173–194.

CLEVERDON, C. W. The significance of the cranfield tests on index languages. In: ACM. **Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.], 1991. p. 3–12.

CROFT, B.; METZLER, D.; STROHMAN, T. **Search Engines: Information Retrieval in Practice**. [S.l.]: Pearson Education, 2011. ISBN 9780133001594.

CROFT, W.; METZLER, D.; STROHMAN, T. **Search Engines: Information Retrieval in Practice**. [S.l.]: Pearson, 2010. ISBN 9780131364899.

CURRAN, J. R.; MOENS, M. Improvements in automatic thesaurus extraction. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition-Volume 9**. [S.l.], 2002. p. 59–66.

DEERWESTER, S. Improving information retrieval with latent semantic indexing. 1988.

ERMAKOVA, L.; MOTHE, J. Query expansion by local context analysis. In: **Conference francophone en Recherche d'Information et Applications (CORIA 2016)**. [S.l.: s.n.], 2016. p. pp–235.

FAZZINGA, B.; GIANFORME, G.; GOTTLOB, G.; LUKASIEWICZ, T. Semantic web search based on ontological conjunctive queries. **Web Semantics: Science, Services and Agents on the World Wide Web**, Elsevier, v. 9, n. 4, p. 453–473, 2011.

GOLDSCHMIDT, R.; PASSOS, E. **Data Mining**. [S.l.]: Elsevier Brasil, 2015. ISBN 9788535278231.

GONG, Z.; CHEANG, C. W.; HOU, U. L. Web query expansion by wordnet. In: SPRINGER. **International Conference on Database and Expert Systems Applications**. [S.l.], 2005. p. 166–175.

HARPER, D. J.; RIJSBERGEN, C. J. V. An evaluation of feedback in document retrieval using co-occurrence data. **Journal of documentation**, MCB UP Ltd, v. 34, n. 3, p. 189–216, 1978.

HARRIS, Z. S. Distributional structure. **Word**, Taylor & Francis, v. 10, n. 2-3, p. 146–162, 1954.

HASHEMI, S. H.; CLARKE, C. L.; KAMPS, J.; KISELEVA, J.; VOORHEES, E. M. Overview of the trec 2016 contextual suggestion track. In: **Proceedings of TREC**. [S.l.: s.n.], 2016. v. 2016.

HSU, M.-H.; TSAI, M.-F.; CHEN, H.-H. Query expansion with conceptnet and wordnet: An intrinsic comparison. In: SPRINGER. **Asia Information Retrieval Symposium**. [S.l.], 2006. p. 1–13.

JOACHIMS, T.; GRANKA, L.; PAN, B.; HEMBROOKE, H.; GAY, G. Accurately interpreting clickthrough data as implicit feedback. Citeseer, 2005.

KROVETZ, R.; CROFT, W. B. Lexical ambiguity and information retrieval. **ACM Transactions on Information Systems (TOIS)**, ACM, v. 10, n. 2, p. 115–141, 1992.

LANDAUER, T. K.; DUMAIS, S. T. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. **Psychological review**, American Psychological Association, v. 104, n. 2, p. 211, 1997.

LEBRET, R.; COLLOBERT, R. Rehabilitation of count-based models for word vector representations. In: SPRINGER. **International Conference on Intelligent Text Processing and Computational Linguistics**. [S.l.], 2015. p. 417–429.

LEE, K. S.; CROFT, W. B.; ALLAN, J. A cluster-based resampling method for pseudo-relevance feedback. In: ACM. **Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.], 2008. p. 235–242.

LEVY, O.; GOLDBERG, Y. Linguistic regularities in sparse and explicit word representations. In: **Proceedings of the eighteenth conference on computational natural language learning**. [S.l.: s.n.], 2014. p. 171–180.

LI, W.; GANGULY, D.; JONES, G. J. Using wordnet for query expansion: Adapt@ fire 2016 microblog track. In: **FIRE (Working Notes)**. [S.l.: s.n.], 2016. p. 62–65.

LIN, D. Automatic retrieval and clustering of similar words. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 17th international conference on Computational linguistics-Volume 2**. [S.l.], 1998. p. 768–774.

LIU, H.; MOTODA, H. **Computational methods of feature selection**. [S.l.]: CRC Press, 2007.

LIU, S.; LIU, F.; YU, C.; MENG, W. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In: ACM. **Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.], 2004. p. 266–272.

LOWE, W. Towards a theory of semantic space. In: **Proceedings of the Annual Meeting of the Cognitive Science Society**. [S.l.: s.n.], 2001. v. 23.

LU, M.; SUN, X.; WANG, S.; LO, D.; DUAN, Y. Query expansion via wordnet for effective code search. In: IEEE. **Software Analysis, Evolution and Reengineering (SANER), 2015 IEEE 22nd International Conference on**. [S.l.], 2015. p. 545–549.

METZLER, D.; CROFT, W. B. Latent concept expansion using markov random fields. In: ACM. **Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.], 2007. p. 311–318.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

MIKOLOV, T.; YIH, W.-t.; ZWEIG, G. Linguistic regularities in continuous space word representations. In: **Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S.l.: s.n.], 2013. p. 746–751.

MILLER, G. A. Wordnet: a lexical database for english. **Communications of the ACM**, ACM, v. 38, n. 11, p. 39–41, 1995.

ONE, V. D. Cd-rom from virginia polytechnic institute and state university. **Blacksburg, VA**, 1990.

OOI, J.; MA, X.; QIN, H.; LIEW, S. C. A survey of query expansion, query suggestion and query refinement techniques. In: IEEE. **Software Engineering and Computer Systems (ICSECS), 2015 4th International Conference on**. [S.l.], 2015. p. 112–117.

PORTER, M. F. An algorithm for suffix stripping. **Program**, MCB UP Ltd, v. 14, n. 3, p. 130–137, 1980.

QIAN, G.; SURAL, S.; GU, Y.; PRAMANIK, S. Similarity between euclidean and cosine angle distance for nearest neighbor queries. In: ACM. **Proceedings of the 2004 ACM symposium on Applied computing**. [S.l.], 2004. p. 1232–1237.

RIJSBERGEN, C. J. V. **Information Retrieval**. 2nd. ed. Newton, MA, USA: Butterworth-Heinemann, 1979. ISBN 0408709294.

ROBERTSON, S.; ZARAGOZA, H. *et al.* The probabilistic relevance framework: Bm25 and beyond. **Foundations and Trends® in Information Retrieval**, Now Publishers, Inc., v. 3, n. 4, p. 333–389, 2009.

ROBERTSON, S. E.; JONES, K. S. Relevance weighting of search terms. **Journal of the Association for Information Science and Technology**, Wiley Online Library, v. 27, n. 3, p. 129–146, 1976.

ROCCHIO, J. J. Relevance feedback in information retrieval. **The SMART retrieval system: experiments in automatic document processing**, Prentice-Hall Inc., p. 313–323, 1971.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. [S.l.]: Prentice Hall, 2010. (Prentice Hall Series in Artifi). ISBN 9780136042594.

SALTON, G. The smart retrieval system—experiments in automatic document processing. Prentice-Hall, Inc., 1971.

SALTON, G.; BUCKLEY, C. Improving retrieval performance by relevance feedback. **Journal of the American society for information science**, Wiley Online Library, v. 41, n. 4, p. 288–297, 1990.

SALTON, G.; WONG, A.; YANG, C.-S. A vector space model for automatic indexing. **Communications of the ACM**, ACM, v. 18, n. 11, p. 613–620, 1975.

SALTON, G.; YANG, C.-S. On the specification of term values in automatic indexing. **Journal of documentation**, MCB UP Ltd, v. 29, n. 4, p. 351–372, 1973.

SANDERSON, M.; CROFT, W. B. The history of information retrieval research. **Proceedings of the IEEE**, IEEE, v. 100, n. Special Centennial Issue, p. 1444–1451, 2012.

SCHÜTZE, H.; MANNING, C. D.; RAGHAVAN, P. **Introduction to information retrieval**. [S.l.]: Cambridge University Press, 2008. v. 39.

SINGHAL, A. Modern information retrieval: A brief overview. **IEEE Data Eng. Bull.**, v. 24, n. 4, p. 35–43, 2001.

TURNEY, P. D.; PANTEL, P. From frequency to meaning: Vector space models of semantics. **Journal of artificial intelligence research**, v. 37, p. 141–188, 2010.

WAN, J.; WANG, W.; YI, J.; CHU, C.; SONG, K. Query expansion approach based on ontology and local context analysis. **Research Journal of Applied Sciences, Engineering and Technology**, Maxwell Science Publishing, v. 4, n. 16, p. 2839–2843, 2012.

XU, J.; CROFT, W. B. Query expansion using local and global document analysis. In: ACM. **Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.], 1996. p. 4–11.

XU, J.; CROFT, W. B. Improving the effectiveness of information retrieval with local context analysis. **ACM Transactions on Information Systems (TOIS)**, ACM, v. 18, n. 1, p. 79–112, 2000.

ZOBEL, J.; MOFFAT, A. Inverted files for text search engines. **ACM computing surveys (CSUR)**, ACM, v. 38, n. 2, p. 6, 2006.