UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Curso de Bacharelado em Ciência da Computação

Trabalho de Conclusão de Curso

Currículos Lattes: Expansão Automática de Termos baseada em Ontologia

Glauco Roberto Munsberg dos Santos

Pelotas, 2015

Glauco Roberto Munsberg dos Santos

Currículos Lattes: Expansão Automática de Termos baseada em Ontologia

Trabalho de Conclusão de Curso apresentado ao Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação

Orientador: Prof. Dr. Ricardo Matsumura Araujo
Coorientadora: Prof[a]. Dr. Daniela Francisco Brauner

Pelotas, 2015

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

S237c Santos, Glauco Roberto Munsberg dos

Currículos Lattes : expansão automática de termos baseada em ontologia / Glauco Roberto Munsberg dos Santos ; Ricardo Araújo Matsumura, orientador ; Daniela Francisco Brauner, coorientadora. — Pelotas, 2015.

81 f. : il.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) — Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2015.

1. Currículos Lattes. 2. Recuperação de informação. 3. Expansão de termos. 4. Expansão automática de consulta. 5. Ontologia. I. Matsumura, Ricardo Araújo, orient. II. Brauner, Daniela Francisco, coorient. III. Título.

CDD : 005

Elaborada por Aline Herbstrith Batista CRB: 10/1737

Glauco Roberto Munsberg dos Santos

Currículos Lattes: Expansão Automática de Termos baseada em Ontologia

Trabalho de Conclusão de Curso aprovado, como requisito parcial, para obtenção do grau de Bacharel em Ciência da Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 07 de Dezembro de 2015

Banca Examinadora:

Prof. Ricardo Matsumura de Araújo – Orientador

Prof. Alexandre Rademaker

Profa. Ana Marilza Pernas Fleischmann

Profa. Tatiana Aires Tavares

Dedico aos sábios e insanos ensinamentos que me transformaram e a todos que em mim ecoam de alguma forma...

AGRADECIMENTOS

A compreensão de minha própria existência naturalmente iniciou-se com a pergunta *"Quem sou eu?"* e esta pergunta me perseguiu por muito tempo até ter maturidade o suficiente para entender o "penso, logo existo" de Descartés. É um grande exercício perceber que sou resultado das minhas experiências e da forma pela qual interpreto o mundo.
Então se sou a materialidade dessas experiênicas e do meu modo particular de ver o mundo, a próxima pergunta que me fiz sintetiza a minha vontade de compreender o que me move: afinal *"Estou fazendo o que me apaixona?"*.
Ajustadas as velas para compreender esse novo viés e hoje aos 27 anos sou grato a cada segundo que tive até agora e sem exceções gostaria de agradecer a **todos** que se fizeram presentes até este momento:
Agradeço aos meus pais, sou imensamente grato por ser filho de vocês, por receber esse amor incondicional e compreensão. Sem eles não seria possível estar aqui hoje escrevendo esses agradecimentos. Também aqueles que partiram entre eles amigos e entes: vocês *existem* em mim, sou fruto da sinergia que houve entre nós.
Aos amigos... Ah aos amigos! Obrigado por todos os momentos de conselhos, discussões, lágrimas e sorrisos. Agradeço pelos muitos momentos de embates de viés filosóficos que acredito que tenham chegado sempre ao mesmo lugar: A essa amizade que tanto cultivo com cada um de vocês. Aos amigos *renegados*: Vocês são partes dessa **paixão** que construi nesses últimos 5 anos.
Alma mater UFPel você me proporcionou vivências que estarão eternamente riscadas em meus pensamentos e ações, exercer a discência em seu colo me fez um homem mais sábio... Obrigado por me acolher em todos os aspectos. Prof. Ricardo e Profª. Daniela obrigado por dividir comigo a sabedoria de vocês. Esse trabalho tem meu suor e muito do que aprendi com vocês.

“Se você não tem uma visão de futuro, está condenado a viver eternamente a repetição de seu passado.” — A.R. BERNARD

RESUMO

SANTOS, Glauco Roberto Munsberg dos. **Currículos Lattes: Expansão Automática de Termos baseada em Ontologia**. 2015. 82 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2015.

A Plataforma Lattes, criada e mantida pelo CNPq, integra as bases de dados de currículos, grupos de pesquisa e instituições, em um único sistema de informações e é um importante meio pelo qual pesquisadores, professores e alunos vêm reunindo informações sobre a suas formações e trabalhos produzidos durante décadas. Assim, hoje o Lattes é referência nacional no formato de currículo, divulgação da produção científica e tecnológica brasileira. Porém, a ferramenta de pesquisa atual do Lattes exige grande compreensão da busca que se espera, e de certa forma, uma objetividade ao realizar uma pesquisa. Com isso, propomos um mecanismo de busca que permita uma melhor experiência de uso, aumentando a relevância dos resultados retornados a partir de uma busca fornecida pelo usuário. De um lado, percebe-se que há um alto grau de informalidade nos termos buscados mais frequentemente pelos usuários, enquanto os pesquisadores tendem a usar termos técnicos específicos para descrever seus trabalhos em seus currículos. Como as ferramentas tradicionais de recuperação de informação utilizam apenas os termos que são mencionados nos currículos para indexar a informação, os usuários precisam ter conhecimento desses termos para recuperar currículos relevantes em suas consultas. Como forma de melhorar esses resultados, propomos aqui o desenvolvimento de um mecanismo de busca com expansão de termos apoiado por uma base de conhecimento. O objetivo é ampliar os resultados da busca fornecendo assim uma melhor experiência de uso. Os resultados obtidos mostraram que houve um ganho significativo na aproximação do vocabulário entre o utilizado pela comunidade e pelas publicações indexadas. Também foi observando que 23,1% das consultas realizadas contaram com uma expansão e que destas 64,9% foram clicadas pelo usuário. Isso demonstra, que para esse conjunto de currículos, houve uma relevância significativa para o motor de busca o uso de expansão de termos proposto aqui.

Palavras-chave: Currículos Lattes, Recuperação de Informação, Expansão de Termos, Expansão Automática de Consulta, Ontologia, WordNet, Feeling.

ABSTRACT

SANTOS, Glauco Roberto Munsberg dos. **Curriculum Lattes: Automatic Expansion Terms based on Ontology**. 2015. 82 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2015.

The Lattes Platform, created and maintained by CNPq, integrates the database of resumes, research groups and institutions into a single information system and is an important means by which researchers, teachers and students have been gathering information on their training and works produced for decades. Today the Lattes is a national reference in resume format and dissemination of Brazilian scientific literature and technology . But the current research tool Lattes requires great understanding of search that is expected, and in some ways, an objectivity when performing a search.Thus, we propose a search engine that allows for a better user experience, increasing the relevance of the results returned from a search provided by the user. On the one hand, it is clear that there is a high degree of informality in terms most often searched by users, while researchers tend to use specific technical terms to describe their work on their resumes. As the traditional tools of information retrieval uses only the terms that are mentioned in the curriculum to index the information, users need to be aware of these terms to retrieve relevant curricula in your queries. As a means to improve these results, we propose here the development of a search engine in terms of expansion supported by a knowledge base. The objective is to expand the search results thus providing a better user experience.The results showed that there was a important gain in approach between the vocabulary used by the community and the indexed publications. It was also observed that 23.1% of consultations drew an expansion and that these consultations 64.9% of them were clicked by the user demonstrates that for this set of curricula, there was a significant relevance to the search engine use expansion terms proposed here.

Keywords: Currículos Lattes, Information Retrieval, Expansion Terms, Automatic Query Expansion, Ontology, WordNet, Freeling.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SRI	Sistema de Recuperação de informação
RI	Recuperação de informação
EAC	Expansão Automática de Consulta
PL	Plataforma Lattes
TCC	Trabalho de Conclusão de Curso
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CIT	Centro de Inovação Tecnológica
UFPel	Universidade Federal de Pelotas
SGBD	Sistema de Gerenciamento de Banco de Dados
LOD	Linked Opened Data
NLP	Natural Language Processing (Processamento Natural de Linguagem)

SUMÁRIO

1 INTRODUÇÃO

Na Ciência da Computação a Recuperação de Informação (RI) é uma área abrangente que centraliza seus esforços em fornecer ao usuário uma forma fácil de extrair, de um montante maior de informações, as que sejam relevantes. Essa demanda pela classificação e resgate da informação nos remete ao motivo de sua necessidade.

Com o crescente volume de informação gerado pela sociedade, nasceu a necessidade de extrair rapidamente informações de grandes volumes de dados. Naturalmente, as bibliotecas foram as primeiras a se deparar com esse problema, visto que em uma catalogação pode haver inúmeros campos de informações e a busca manual em um catálogo extenso pode ser demorado, oneroso e ineficiente.

Com o propósito de auxiliar essa recuperação de informação, surgiram os sistemas de recuperações de informações (SRI). O objetivo principal de um sistema de RI é recuperar os documentos relevantes à necessidade de informações do usuário e, ao mesmo tempo, recuperar o menor número possível de documentos irrelevantes.(BAEZA-YATES R., 2013)

1.1 O Problema

O problema central desse trabalho está em como permitir que a comunidade encontre competências de uma universidade, ou de um grupo de acadêmicos, através de uma busca intuitiva em uma base de currículos do grupo. Atualmente a Plataforma Lattes[1] (PL), mantida pelo CNPq[2] (Conselho Nacional de Pesquisa e Tecnologia), oferece uma base interessante para busca de currículos de pesquisadores, já que tem como finalidade interligar diversas bases de dados como a de currículos, de grupos de pesquisa e de Instituições através de um único sistema. Hoje, a base da PL conta com mais de 3 milhões de currículos cadastrados[3]. A busca[4] e a recuperação dessas informações tornaram-se um processo trabalhoso ao usuário, visto que a plataforma

[1] http://lattes.cnpq.edu.br
[2] http://www.cnpq.br
[3] http://estatico.cnpq.br/painelLattes/
[4] http://buscatextual.cnpq.br/buscatextual/busca.do?metodo=apresentar

implementa mecanismos tradicionais de busca por termos identificados nos currículos dos pesquisadores. O usuário precisa ter conhecimento especializado sobre os termos a serem utilizados em suas buscas (SOUZA MEIRELES, 2014, p. 80). Sendo assim, uma busca por um conceito ou área pode resgatar apenas 27% do montante esperado para aquela busca (SOUZA MEIRELES, 2014, p. 62).

Esta problemática também foi identificada pela demanda da CIT[5] (Coordenadoria de Inovação Tecnológica) da UFPel[6] que instigou o curso de Computação no CDTec[7] em trabalhar para criar meiopara dar visibilidade dos conhecimentos da universidade. Com o objetivo de promover uma melhora na busca do conteúdo dos currículos e também permitir a visibilidade dos conhecimentos propoem-se aqui a continuidade do trabalho de SOUZA MEIRELES (2014). Porém, optou-se por partir do início, sem aproveitar a implementação proposta por ele, com o objetivo de construir um motor de busca que aumentasse a acurácia dos resultados sobre os currículos dos pesquisadores da Instituição.

Não obstante, há o problema central desse trabalho que é em como prover a um SRI de currículos Lattes um meio que aproxime os termos usados pela comunidade com aqueles usados pelos pesquisadores. Visto que existe um grande descompasso nos tipos de termos utilizados pela comunidade e pelos pesquisadores, já que a comunidade utiliza termos mais informais na pesquisa, enquanto que os pesquisadores utilizam jargões técnico-científicos para descrever seus trabalhos. Atualmente, a ferramenta de busca provida pelo CNPq, aparentemente, não possui mecanismos de expansão de termos ou de consulta. Ademais esses mecanismos tendem a tornar a busca mais flexível a luz do usuário.

Neste contexto, esse trabalho propõem um estudo e uma pesquisa exploratória em uma base de conhecimento com o objetivo de identificar soluções. Como será visto no decorrer deste obra, o trabalho enfoca-se na construção de métricas que permitam expandir os termos relevantes encontrados nos currículos. Os termos originais e os termos expandidos devem possuir pesos que permitam a classificação dos resultados pelo SRI.

E esse mecanismo é pensado no modelo mental consolidado de busca para usuários: Em que o usuário seja capaz de inserir os termos chaves e com um processo escondido em uma “caixa preta”, o motor de busca seja capaz de realizar o mecanismo proposto acima e retornar um resultado satisfatório para o usuário, listando o resultado através desta classificação (*ranking*) (BÜTTCHER; CLARKE; CORMACK, 2010, p.6).

[5] http://wp.ufpel.edu.br/prppg/equipe-inovacao
[6] http://www.ufpel.edu.br
[7] http://cdtec.ufpel.edu.br

1.2 Metodologia

Para este TCC foi pensado um modelo de desenvolvimento baseado em experimentações e avaliações passando primeiramente pela coleta, catalogação dos fatos e objetos para esse trabalho. Nesse contexto, o primeiro momento foi dedicado a compreensão do problema, sua magnitude e a revisão bibliográfica com foco em como resolver a expansão de termos através do uso de ontologias.

Posteriormente, realizou-se uma análise para compreender quais campos de um documento são de interesse em uma possível expansão dos termos. Foi realizado um estudo exploratório para identificar uma base de conhecimento que pudesse subsidiar a aproximação de termos usados entre pesquisadores e a comunidade. Além disso, foi necessário identificar um formato de representação dos termos expandidos, que permita conectar os conceitos da base de conhecimento aos dados dos pesquisadores.

Pensando na solução, percebemos o problema de que haveria a necessidade de criação de uma métrica para a relevância dos campos a serem expandidos e sua profundidade de expansão.

Revisando os trabalhos relacionados, pode-se então compreender os principais problemas da área de RI, bem como, o modo pelo qual deve-se extrair o sentido e palavras sinônimas da ontologia ou base de conhecimento escolhida (SHEKARPOUR et al., 2013, p.4). Com esse embasamento teórico partiu-se para o processo conceitual da metodologia proposta e do SRI e assim completa-se a primeira fase do processo.

A segunda fase dedicou-se na construção da arquitetura resultante da modelagem conceitual da primeira fase, junto as tecnologias e ferramentas que melhor se adequaram ao problema, partiu-se para o desenvolvimento da ferramenta. Fases seguintes (teste, análise e conclusões) foram adicionado de forma incremental nesse TCC de acordo com o término dessas etapas.

1.3 Objetivos

Espera-se através desse TCC atingir o objetivo geral e para isso usou-se dos objetivos específicos para conduzir a tal feito. Sendo assim, nesta seção expõem-se abaixo cada um desses objetivos e, de forma ordenada, como pretende-se resolver o problema.

1.3.1 Objetivo Geral

O objetivo central deste trabalho consiste em implementar e avaliar um sistema de expansão de termos para os currículos da PL. Sendo proposto aqui uma ferramenta capaz de expandir termos relevantes encontrados dentro dos currículos.

1.3.2 Objetivos Específicos

Para se obter o objetivo geral desse TCC naturalmente é necessário passar por uma série de etapas e processos dos quais são marcos fundamentais para a construção de uma solução. Para esse êxito foram propostos os seguintes objetivos específicos:

- Investigar nos currículos da PL quais campos contém os termos relevantes para a expansão;
- Investigar e definir a métrica para a relevância e peso das palavras indexadas usadas no SRI;
- Gerar uma interface gráfica que permita o usuário compreenda e interaja com o sistema;
- Avaliar as métricas propostas com o intuito de compreender o impacto das expansões de termos sobre a consulta;

1.4 Organização do Trabalho

O trabalho está estruturado em 5 (cinco) capítulos com o objetivo de orientar melhor a compreensão do TCC. O primeiro capítulo aborda de forma introdutória o tema do trabalho, bem como a motivação e objetivo. Na sequência, o segundo capítulo apresenta quais são os trabalhos relacionados, os SRIs e a estrutura dos mesmo e a fundamentação teórica da área que apoia esse TCC, é introduzido o conceito de ontologias e bases de conhecimento, que junto ao SRI, são pilares de sustentação desse trabalho. A última sessão deste capítulo é dedicado as tecnologias utilizadas no apoio da concepção da ferramenta.

O terceiro capítulo está dedicado a concepção da ferramenta, a estrutura da informação, a compreensão das métricas propostas tanto na expansão de termos quanto no processo adotado em cada etapa que o compõem e por fim a implementação da ferramenta proposta como um todo. No quarto capítulo são apresentados os resultados da aplicação desta ferramenta e comparativos pertinentes, bem como a avaliação dos resultados obtidos em experimentos realizados sobre o SRI. Já no quinto capítulo são apresentadas as considerações finais do trabalho, contribuições e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICAS

Neste capítulo abordaremos de forma aprofundada as duas áreas que dão embasamento teórico para a realização desse trabalho, bem como, também abordaremos na terceira sessão todas as tecnologias e também os trabalhos relacionados que serviram de base para este trabalho.

2.1 SRI

Essa seção dedica-se a compreensão e fundamentação teórica em relação aos SRI. Aqui serão apresentados os principais conceitos da área de RI para a construção de um SRI. Para isso, as etapas que são pertinentes a esse trabalho estão divididas nas seis sessões abaixo descritas.

2.1.1 Arquitetura de um SRI

A construção da arquitetura de um SRI está baseada em dois requisitos básicos de software: *Eficácia* e *Eficiência*. Sendo que, para o primeiro requisito, quando abordado na área de RI, preocupa-se em prover um mecanismo capaz de recuperar o conjunto mais significativo de documentos para uma determinada consulta do usuário, isso implica na qualidade sobre o sistema. Já o segundo requisito, eficiência, é então esperado que o SRI processe a consulta do usuário o mais rápido possível, implicando assim, no tempo de resposta do sistema(CROFT; METZLER; STROHMAN, 2010).

A principal dificuldade para atingir a eficácia está em saber não só como extrair a informação dos arquivos, mas também em como utilizá-la para decidir o quanto ela de fato é *relevante*. Esse é o principal ponto em RI. Salienta-se ainda que a “relevância” é um julgamento pessoal que está intimamente ligada a tarefa a ser resolvida e o seu contexto. Assim, a relevância pode temporalmente ser modificada, ou seja, um documento que hoje pode ser útil e relevante para um determinado usuário, amanhã o mesmo documento pode não ter a mesma relevância.

Compreendendo que deve-se construir um SRI que seja tanto eficiênte como eficaz e segundo BAEZA-YATES R. (2013) e CROFT; METZLER; STROHMAN

(2010) geralmente os mesmos são arquitetados sobre cinco componentes: a Coleta, Transformação de Dados, Indexação, Ranqueamento e Consulta (Veja Figura 1), a soma deles corresponde por todo o ciclo de indexação ao ranqueamento.

Contudo, segundo (BAEZA-YATES R., 2013) a melhor abordagem para explicar o funcionamento de um SRI é através da arquitetura de um sistema simples e genérico com três grandes módulos, como ilustra a Figura **??**. O primeiro módulo é responsável pelo Processo de Coleta de informação a partir de algum repositório, na imagem a Web é ilustrada como fonte. O segundo, começa pela etapa de armazenamento da informação local, passando por um indexador que gera a estrutura de *índice invertido* e por vez o processo de recuperação (consulta).

O próximo módulo, o terceiro, é responsável pela busca e conta com a etapa da análise da consulta, da recuperação e de ranqueamento da informação e por fim retorna um conjunto de respostas para o usuário. Abordaremos nas próximas sessões de forma detalhada como executa-se os processos ilustrados pela Figura 1.

Notemos que a primeira etapa na construção de um SRI é o processo de coleta de informação e é aplicado quando não se há uma coleção própria de documento CROFT; METZLER; STROHMAN (2010, p.35). Para esse trabalho não foi necessário contar com este processo, uma vez que os dados foram fornecidos pela instituição.

2.1.2 Processamento de Texto

Após a extração da informação que desejamos pesquisar, o próximo passo é decidir o que precisa ser modificado ou reestruturado de alguma forma para simplificar o processo de busca. Esse processo de mudança, geralmente é chamado de transformação de texto ou processamento de texto e, será abordado nessa seção. A principal tarefa dessa etapa é converter diferentes formas de palavras em algo mais consistente, chamado de termo de indexação. O Termo de indexação é a representação do conteúdo do documento que é utilizado pela busca (BAEZA-YATES R., 2013, p.26),(CROFT; METZLER; STROHMAN, 2010, p.27).

Essa etapa é importante por permitir uma maior normalização do conteúdo extraído pelos coletores antes de ser indexado. A indexação e a busca do texto exato extraído do documento pode causar problemas para a busca. Suponhamos que o usuário busque por “ciência da computação”, e no documento que está escrito “Ciência da Computação” essa busca não seria combinada com o documento, pois, há uma diferença de maiúsculas e minúsculas nas palavras. Felizmente, a maioria das ferramentas de busca já possuem a capacidade de ignorar o maiúsculo e minúsculo das palavras, mas evidencia o quanto a etapa do processamento do texto é importante(BÜTTCHER; CLARKE; CORMACK, 2010, p.85).

As etapas mais comuns a serem realizadas serão abordadas aqui e são elas: a análise sintática, a eliminação de *stopwords* e a radicalização das palavras (*stem-*

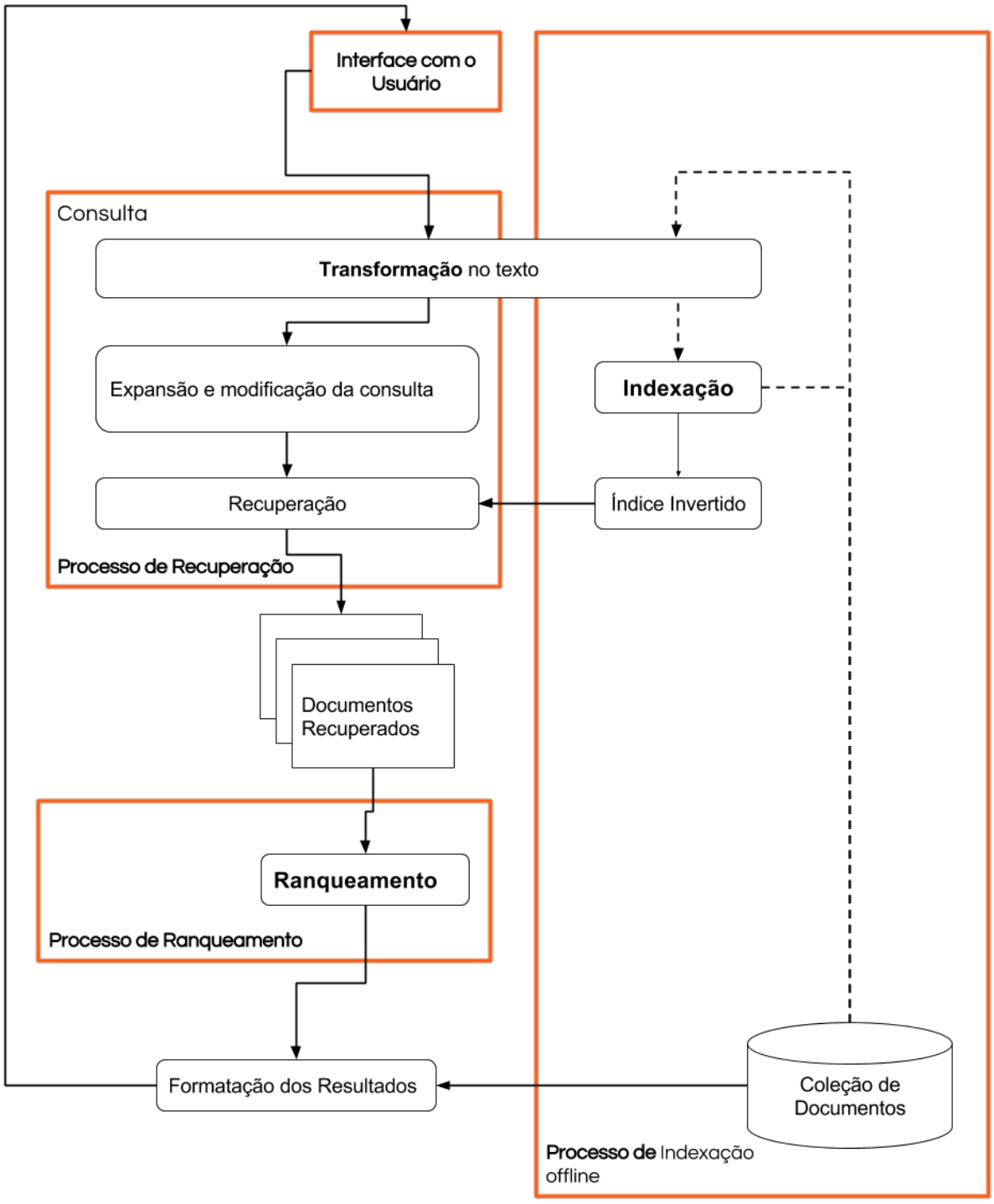

Figura 1: Os processos de indexação, recuperação e ranqueamento de uma coleção de documentos. Fonte: (BAEZA-YATES R., 2013, p.9)

ming). Ao fim dessas etapas chega-se a um conjunto menor do documento, conforme mencionado anteriormente.

2.1.2.1 Tokenização

O processo de Tokenização concentra-se na transformação de uma sequência de caracteres de um determinado documento em *Tokens*. Essa etapa realiza o processamento da sequência de símbolos textuais do documento, reconhecendo assim a sua estrutura. Fazer a quebra de texto em símbolos é um importante passo no processo de transformação e deve ser aplicado tanto antes da indexação como para a entrada da busca a fim de realizar uma combinação perfeita dos elementos (CROFT; METZLER; STROHMAN, 2010, p.87).

A primeira vista, o processo de criação de termos está constituída apenas em dividir palavras pelos espaços que há entre elas e também pelo processo de remoção de caixa alta e baixa das palavras. Porém, há ainda uma série de questões a serem tomadas a respeito de dígitos, hifens, marcas de pontuação, por exemplo. Tradicionalmente, números não são bons indexadores, dado que fora de seu contexto podem produzir resultados vagos, porém, sequências de dígitos são sabiamente importantes e não podem ser desconsiderados por serem bons índices. Outra difícil tomada de decisão enfrentada pelo analisador léxico são os hifens dado que “estado-da-arte” quebrado produziria “estado”,“da”,“arte” o que semanticamente teria um sentido e relevância diferentes na busca da palavra como um todo se a mesma tivesse sido indexada sem a quebra por hífem(BAEZA-YATES R., 2013, 212).

Tanto para BAEZA-YATES R. (2013) como para CROFT; METZLER; STROHMAN (2010) essas operações textuais são de simples aplicação, mas a tomada de decisão a cada um desses desafios produz um impacto profundo sobre o tempo de recuperação do texto. Contudo, não há uma aplicação transversal que seja a solução para esse problema.

2.1.2.2 StopWords: Remoção de palavras de parada

Palavras com alto índice de frequência em documentos são inúteis ao processo de recuperação. Tais palavras são chamadas de *Stopwords* ou palavras de parada e são normalmente removidas dos termos de índice em potencial. Nesse grupo têm como fortes candidatos palavras dos grupos gramaticais: artigos, proposições, conjunções e, podendo ser estendido conforme necessidade. Por exemplo, uma coleção em que documentos, contam com alta frequência da palavra ”currículo”é pertinente que esta seja considerada palavra de parada dado que influenciaria a quantidade de palavras indexadas e também não teria alta relevância (BAEZA-YATES R., 2013, p.213).

Como resultado da remoção das palavras de paradas temos uma diminuição do tamanho geral de termos a serem indexados. Porém, em documentos que contenham

a frase “ser ou não ser”, com a remoção das palavras de parada, o que restaria seria apenas “ser”. Isso dificultaria a recuperação do documento dado que apenas seria combinada uma única palavra a busca. Porém a saída segundo CROFT; METZLER; STROHMAN (2010) é a construção de um conjunto enxuto de *stopwords* baseado na análise estatística sobra o coleção, assim a lista de palavras está intimamente ligada a coleção.

2.1.2.3 Stemming

Plurais e afixos[1] são exemplos claros de que variações podem evitar um união entre o termo usado na busca com os termos presentes no documento. Os usuários se expressão através de uma única variante da palavra, porém, suas derivações também estão contidas indiretamente no desejo de recuperação do usuário. Para contornar esse problema parcialmente, é possível aplicar a técnica de *Stemming* e de lematização que indexa não a palavra como um todo, mas pelo seu *stem* ou *lema*.

Assim é adotado o processo de *stemming* que é um processo que reduz ao mesmo *stem* (parte fundamental semelhante ao radical) palavras que se diferenciam basicamente pela flexa?o. Por exemplo a redução quando aplicada sobre o termo “swimming” e “swam” retornará o mesmo *steam* neste caso provavelmente “swim” e posteriormente indexado pelo motor de busca a fim de aumentar as chances de combinação entre ambas as palavra BÜTTCHER; CLARKE; CORMACK (2010, p.87).

Ainda para CROFT; METZLER; STROHMAN (2010, p.92) há duas formas básicas de se aplicar o processo de *stemming*: Através de dicionários ou de algoritmos. O primeiro método, em busca de dicionário, parece uma técnica simples, para a sua aplicação é necessário apenas um dicionário com todas as palavras da língua em questão e seus respectivos radicais. Porém, parece ser um processo oneroso caso não haja o dicionário com seus radicais pré-compilado.

Já os algoritmos de *stemming* destaca-se nas bibliografias o Algoritmo *Porter Stemmer*[2], ele foi originalmente construído para a língua inglesa e está baseado em uma série de regras e condições aplicadas sequencialmente para esta língua. A adaptação e aplicação do algoritmo ao idioma de base português destacamos os trabalhos de MARTIN PORTER (2015) e ORENGO V. M.; HUYCK (2011).

BAEZA-YATES R. (2013) nos trás ainda dois métodos que são a variação de sucessores e N-Gramas[3], ambos descritos abaixo:

Variedade de Sucessores: Baseia-se na delimitação de morfemas, para isso usa-se conhecimentos da construção linguística em questão. Notoriamente mais complexo que o algoritmos de afixos como Porter Stemmer.

[1] Prefixos e Sufixos

[2] http://tartarus.org/martin/PorterStemmer

[3] n-gramas são sequências de uma ou mais palavras

N-Gramas: Baseia-se na identificação de diagramas e trigramas, esse processo aproxima-se mais ao *clustering* do que processos de *steamming*.

Há diversos estudos que tentam provar ou refutar a eficácia do processo para a melhoria da indexação. Um estudo em particular foi realizado por BAEZA-YATES; FRAKES (1992) com 8 experimentos diferentes no qual não chegaram a uma conclusão definitiva. Como resultado de tais indecisões muitos dos atuais motores de busca evitam esse processo.

2.1.3 Indexação de documento

O objetivo desta seção é demonstrar como obter rapidamente o resultado através da estrutura escolhida para a indexação dos documentos da coleção. Estamos abordando aqui a arquitetura sobre o viés da *eficiência*, em que pode ser utilizado uma série de estruturas com o objetivo de encontrar um item pelo seus atributos, para solucionar esse problema podemos usar tabelas hash ou até mesmo estruturas mais complexas como Árvores B .

Quando observamos a busca em uma biblioteca com mais de 100 mil documentos e realizar a busca internamente em cada um dos documentos nos faz compreender que essa é uma etapa computacionalmente custosa e impactaria diretamente na velocidade. Assim, resultaria em uma *eficiência* menor no resgate da informação, apesar de manter a *eficácia* visto que o processo mais lento ainda retornaria o esperado (CROFT; METZLER; STROHMAN, 2010, p. 125) e (BAEZA-YATES R., 2013, p.339-341).

Na maioria dos casos um índice deve ser utilizado para maximizar a eficiência. Para BAEZA-YATES R. (2013, p. 340) “índices são estruturas de dados construídas a partir do texto para acelerar a busca”, apesar da própria construção e manutenção dessa estrutura ser considerada mais complexa do que a própria execução de uma busca sequencial. Ainda segundo o autor, é a única forma de obter o resultado da busca em um tempo aceitável quando pensado sobre um volume significante de documentos na coleção que se está analisando.

Dentre as técnicas de indexação a estrutura de *Índice Invertido*[4] é utilizada largamente pelos motores de busca. O Índice Invertido trata-se de um leque de algoritmos que em si compartilham a mesma filosofia. São estruturas compostas por dois elementos: O vocabulário e as ocorrências. O primeiro, é o conjunto de todas as palavras que estão presentes ao menos uma vez no texto. Já o segundo, as ocorrências são o apontamento para todos os documentos nos quais essa palavra têm ocorrência (CROFT; METZLER; STROHMAN, 2010, p.129-130).

[4]Também chamado de “Inverted Index” e ”arquivo invertido”(BAEZA-YATES R., 2013, p.342) e (CROFT; METZLER; STROHMAN, 2010, p.129)

Uma forma simples de representar essa estrutura segundo BAEZA-YATES R. (2013, p.343) é: onde um documento guarda-se a matriz do vocábulo com o número de ocorrência pelo qual esta ocorre e uma lista invertida de ocorrências. A ilustração pode ser vista na Figura 2 com mais detalhes e sua aplicação contextual com o SRI na Figura 1.

Vocábulo	F_i
to	2
do	3
be	4
or	1

d_1	d_2	d_3	d_4
4	2	-	-
2	-	3	3
2	2	2	2
-	1	-	-

```
[1,4],[2,2]
[1,2],[3,3],[4,3]
[1,2],[2,2],[3,2]
[2,2]
```

Figura 2: Índice invertido básico e matriz de termos por documentos para a coleção. Fonte: (BAEZA-YATES R., 2013)

Porém, um ponto negativo sobre essa estrutura é o consumo de espaço necessário para armazenar, proporcionalmente ao número de documentos multiplicado pelo tamanho do vocábulo (BAEZA-YATES R., 2013, p.343). Em CROFT; METZLER; STROHMAN (2010, p.140-155) e BÜTTCHER; CLARKE; CORMACK (2010, p.204) são apresentados algumas abordagens para a compressão desta informação que pode obter uma redução entre 25-50%. Porém, fica evidente a simplicidade representada pela estrutura do Índice Invertido, uma vez que é necessário apenas um acesso à matriz para determinar se um documento contém ou não uma determinada palavra.

2.1.4 Ranqueamento

Esta trata-se de uma das tarefas mais fundamentais de um SRI, pois o processo de ranqueamento tem como objetivo realizar a atribuição de pesos aos documentos da coleção retornados em uma determinada busca. Esse não é um processo trivial, dado que está baseado em um julgamento de relevância. Não se trata de uma tarefa fácil, pois parte do pressuposto da compreensão do que é realmente interessante (*relevante*) para o usuário. Uma série de algoritmos vêm sendo desenvolvidos com o objetivo de descrever o mais natural possível como deve ser a relevância na recuperação de documentos. Porém, dois importantes aspectos sobre relevância devem ser observados antes de discutirmos os modelos: Primeiro, é distinguir a diferença entre relevância típica e relevância para o usuário: por exemplo, um documento é tipicamente de interesse se ele contém um tópico de interesse da consulta. Já a relevância para o usuário são todos os demais julgamentos adicionados pelas perspectiva do usuário: idade do documento, linguagem do documento e outros. O segundo aspecto importante para a relevância trata-se da resposta a ser dada e será baseada em consultas

binárias ou multivaloradas: o primeiro, obviamente, apenas mede se determinado documento é interessante ou não. Já o segundo compreende que o julgamento para a relevância deve permitir classificar em mais possibilidades como por exemplo Relevante, Não Relevante e Inseguro (CROFT; METZLER; STROHMAN, 2010, 234).

Na literatura (BÜTTCHER; CLARKE; CORMACK, 2010, p.258-271), (BAEZA-YATES R., 2013, p.26-101) e (CROFT; METZLER; STROHMAN, 2010, p.233-291) existe uma gama modelos matemáticos com o objetivo de prover a melhor relevância. Dado que este trabalho aborda os currículos da PL, levou-se em consideração os algoritmos dedicados a recuperação de textos. A partir deste escopo apresentamos aqui os modelos clássicos de ranqueamento para texto: o modelo Booleano, o modelo baseado em ponderação e por último o modelo Vetorial.

2.1.4.1 Modelo Booleano

Este modelo é baseado na teoria dos conjuntos e na álgebra booleana e é conhecido pela sua simplicidade. No modelo Booleano uma consulta q é uma expressão booleana convenciona sobre termos de indexação, em que os elementos da matriz de termos por documento são representados por 1 quando o termo está presente no documento e por 0 para indicar que o mesmo não está presente. Considerando $c(q)$ como qualquer um dos componentes conjuntivos da consulta e um dado documento d_j, sendo $c(d_j)$ seu componente conjuntivo de documento correspondente então, a similaridade entre o documento e a consulta q é definida a seguir na Figura 3.

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

Figura 3: Modelo Booleano: Sendo $c(q)$ é o componente conjuntivo da consulta e $c(d_j)$ o componente conjuntivo do documento.

Caso $sim(d_j, q) = 1$, então o documento d_j é considerado relevante para q , caso contrário não é considerado importante dado a predição. A abordagem booleana apesar de ser simples não permite em consideração a satisfação parcial para a consulta. Isso torna-se um inconveniente para maioria dos usuários (BAEZA-YATES R., 2013, 26-27).

2.1.4.2 Ponderação pela Frequência dos Termos (TF)

Com a observação dos problemas enfrentados pelo modelo booleano, a abordagem da Frequência dos Termos(TF) realiza a extração de propriedade estatísticas dos textos dos documentos. A ponderação TF, baseia-se na suposição de que o valor (ou peso) de um termo k_i que ocorre em um documento d_j é simplesmente proporcional à

frequência do termo $f_{i,j}$. Isto é , quanto mais frequentemente um termo k_i ocorrer no texto do documento d_j maior será a sua frequência de termo $TF_{i,j}$.

$$TF_{i,j} = \begin{cases} 1 + log_2 f_{i,j} \text{ se } f_{i,j} > 0 \\ 0 \text{ caso contrário} \end{cases}$$

Figura 4: Ponderação TF: $f_{i,j}$ é a frequência do termo k_i no documento d_j

Baseado na observação de que termos com altas frequências são importantes para descrever os tópicos-chave de um documento, a qual leva diretamente à seguinte formulação da ponderação TF que é dada por $TF_{i,j}$ = $f_{i,j}$. Uma variante da ponderação TF utilizada na literatura é onde o algoritmo utiliza a base 2. Esse forma logarítmica é preferível para ponderação TF porque torna os pesos diretamente comparáveis aos pesos IDF, que também são expressos com uma função logarítmica.

2.1.4.3 Ponderação pela Frequência Inversa de Documentos (IDF)

A Frequência Inversa de Termo (IDF) trás uma importante melhoria ao SRI por basear-se na relação entre *exaustividade* e *especificidade*, que em síntese pode ser descrita como: Quanto maior a descrição deste documento, a especificidade destes termos tende a ficar menor. Esta propriedade pode ser exercitada quando observamos uma coleção nos quais todos os documentos contêm um determinado termo. Se todos os documentos possui o determinado termo então dizemos que o termo tem especificidade mínima e torna-se inútil para a extração, pois a recuperação retornaria todos os documentos. Isto leva a ideia da ponderação de termos por especificidade. Com base nos conceitos de especificidade e exaustividade, se chegou a um modelo de ponderação onde o valor do peso do termo é zero se ele for encontrado em todos os documentos, e caso ele apareça em poucos documentos esse valor do termo tende a aumentar. O algoritmo para esta análise é dado por:

1) Verificar a ocorrência do termo k_i para cada documento d_j da coleção n_i ;

2) Calcular a frequência relativa inversa de cada termo (N/n_i);

3) Por fim, aplicar a função log_2 na frequência relativa inversa de cada termo.

Como pode ser observado na Figura 5 que representa a fórmula quando n_i se aproxima de N, temos que IDF_i se aproxima de zero.

2.1.4.4 Ponderação de Termos TF-IDF

Este é o modelo de ponderação mais popular, sendo ele a combinação dos fatores TF e IDF. Logo, podemos definir que $w_{i,j}$ referente ao peso do termo associado ao

$$IDF_i = log_2 \frac{N}{n_i}$$

Figura 5: Poderação IDF: N é o número total de documentos e n_i é o número de documentos em que o termo k_i aparece

par (k_j, d_j), define-se a ponderação do TF-IDF como é mostrado na Figura 6. Embora simples, os pesos TF-IDF são bastantes eficazes para coleções genéricas, isto é, para atribuir pesos aos termos de uma coleção de documentos sobre a qual não temos nenhuma informação (BÜTTCHER; CLARKE; CORMACK, 2010, p.57).

$$TF_{i,j} = \begin{cases} (1+\log_2 f_{i,j}) \times \log_2 \frac{N}{n_i} & \text{se } f_{i,j}>0 \\ 0 & \text{caso contrário} \end{cases}$$

Figura 6: Fórmula da ponderação TF-IDF

2.1.4.5 Modelo Vetorial

Para BAEZA-YATES R. (2013, p.45) o modelo vetorial reconhece que a recuperação booleana é bastante limitada e propõe o modelo vetorial como um quadro no qual casamentos parciais são possíveis. Isto é feito por meio de atribuições *não binárias* aos termos indexados das consultas usando para computar a relevância um *grau de similaridade*. Assim, o Modelo vetorial ordena os documentos recuperados de forma decrescente e o principal efeito disso é que documentos ranqueados formam uma resposta que melhor satisfaz o usuário e mais precisa que o Modelo Booleana. No modelo vetorial os termos de indexação são representados por vetores unitários em um espaço com t dimensões, no qual t é o número de termos de indexações. Neste modelo, o cálculo para obtenção do grau de similaridade de um documento d_j em relação a uma consulta q é dado sob a forma da correlação entre os vetores $\vec{d_j}$ e $\vec{q}$, os quais são formados pelas t dimensões dos termos que os compõem: $\vec{d_j} = (w_{1,j}, w_{2,j}, ..., w_{t,j})$ e $q = (w_{1,q}, w_{2,q}, ..., w_{t,q})$. O peso $w_{i,j}$ associado ao par termo documento (k_i, d_j), e o peso $w_{i,q}$ associado ao par termo-consulta $(k_{i,q})$, são não negativos, não binários, e são dados pela ponderação TF-IDF. A correlação entre os vetores $\vec{d}$ e $\vec{q}$ pode ser quantificada pelo cosseno do ângulo entre esses dois vetores.

No modelo vetorial o $sim(d_j, q)$ (Figura 1) difere do booleano, já que este retorna valores 0,1, e o vetorial retorna valores entre 0 e 1. O modelo vetorial também consegue ordenar os documentos de acordo com esse grau de similaridade em relação à consulta, dessa forma um documento pode ser recuperado mesmo que ele satisfaça a consulta apenas parcialmente, ordenando os documentos recuperados de forma decrescente de acordo com esse grau de similaridade. Tanto para BAEZA-YATES R.

Formula	
$sim(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{\left\|\vec{d_j}\right\| X \left\|\vec{q}\right\|} = \frac{\sum_{i=1}^{t} w_{i,j} X \sum_{i=1}^{t} w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} X \sqrt{\sum_{i=1}^{t} w_{i,j}^2}}$	
Componente	**Descrição**
$\left\|\vec{d_j}\right\| e \left\|\vec{q}\right\|$	São as normas dos vetores do documento e da consulta.
$\vec{d_j} \cdot \vec{q}$	É o produto interno dos dois vetores.
$\left\|\vec{q}\right\|$	Este fator não afeta o ranqueamento (isto é, a ordenação dos documentos) porque ele é o mesmo para todos os documentos.
$\left\|\vec{d_j}\right\|$	Este fator faz a normalização pelo tamanho do documento.

Tabela 1: Descrição da função de similaridade do Modelo Vetorial

(2013, p.47) quanto para CROFT; METZLER; STROHMAN (2010, p.239-241) as principais vantagens do modelo vetorial são:

1) Seu esquema de ponderação de termos melhora a qualidade de recuperação;

2) Sua estratégia de casamento parcial permite a recuperação de documentos que aproximem as condições de consulta;

3) A fórmula do cosseno ordena os documentos de acordo com a similaridade com os termos da consulta;

4) A normalização pelo tamanho dos documentos está naturalmente embutida no modelo.

O Modelo vetorial é uma boa estratégia de ranqueamento para coleções genérias por fornecer resultados reanqueados que dificilmente podem ser melhorados. Mesmo já havendo diversos modelos alternativos de ranqueamento o consenso parece ser que, com coleções genéricas, o Modelo Vetorial é um bom e sólido método.

2.1.5 Consulta

Há diferentes modelos de consultas que podem ser implementadas pelos SRI, entre eles: o modelo de *Full-Text*, o modelo baseado em hipertexto e o de palavras chaves que é amplamente usado nas máquinas de busca da Web. Para este trabalho optou-se pelo modelo de palavras-chaves. A consulta é composta por uma ou mais palavras-chaves e os documentos contendo-as são buscados.

Entretanto, as consultas por palavras apesar de retornar os documentos que as possuem, não são capazes de ordenar segundo o contexto. Para isso muitos sistemas complementam as consultas de uma única palavra com a habilidade em procurar palavras em um dado contexto, isto é, perto de outras palavras. Os modelos de *Frase* e *Proximidade* são ambos modelos que partem do pressuposto que: quanto maior a aproximação das palavras maior é a probabilidade de relevância do documento daqueles que as têm (as palavras) separadas (maior distância entre elas). O modelo Frase é

o mais rígido dos dois, em que a consulta como um todo é apenas uma palavra a ser consultado no índice. Assim “melhor consulta” poderia casar com algum documento que contenha a sequência “... melhor consulta ...”. Outro modelo é o de Aproximidade. Este é um modelo menos rígido e parte de uma sequência de palavras isoladas ou frases junto com uma distância máxima permitida entre elas. Para o exemplo anterior “melhor consulta”, se fosse repassado para o modelo de aproximidade com peso 4, as palavras “melhor” e “consulta” poderiam conter no máximo 4 palavras entre elas. Qualquer quantidade menor ou igual ao valor 4 satisfaz a restrição (BAEZA-YATES R., 2013, p.251). De forma opcional, alguns SRIs ainda possibilitam a expansão de busca (CROFT; METZLER; STROHMAN, 2010, p. 199-207) e (BAEZA-YATES R., 2013, p.178-182) genericamente organizado por:

1) Adição de Sinônimia

2) Análise local (documentos recuperados para uma consulta)

3) Análise global (coleção de documentos);

4) Tesauro

Este trabalho aborda a expansão de termos não em tempo de consulta, mas posteriormente ao Processamento do Texto e anteriormente ao Processo de Indexação.

2.2 Linked Open Data

Esta seção tem como objetivo apresentar a fundamentação teórica sobre o funcionamento da *Linked Open Data*, estrutura e composição de *Ontologias* e abordar o funcionamento das linguagens de representação que servem de apoio as ontologias. O objetivo é encontrar no *Linked Data*, uma forma para representar o conhecimento expandido da base de currículos, de forma a conectar com as demais bases de conhecimento disponíveis abertamente.

A web, como a conhecemos, nasceu com o objetivo de ser uma rede global para publicação e interligação de documentos. Hoje, evoluiu para um ambiente, no qual, os usuários são colaboradores ativos na sua construção (Web 2.0). Apesar dela ser interligável, intangível e abstrata, a web está tão presente de forma que é difícil perceber a divisão tênue entre a vida real e a vida virtual (DIAS; SANTOS, 2013). O principal desafio hoje é fazer com que as máquinas[5] possam compreender o significado do conteúdo da Web com o intuito de prover serviços inteligentes às pessoas. Dessa forma, surgiu o conceito de Web Semântica (Web 3.0 ou Web dos Dados), cuja “espinha dorsal” está na estruturação dos dados, a partir do uso de artefatos tecnológicos e ferramentas.

[5]representadas por agentes de softwares inteligentes

A *Linked Open Data*(LOD) é um esforço conjunto em construir modelos de descrição de dados usando protocolos de transferência de dados (HTTP) para publicar estruturas na internet que promovam a integração de diferentes fontes de dados, efetivamente, permitindo a um fonte de dados ser ”ligada’ a outra fonte de dados. A iniciativa começa no ano de 2006 quando Tim Berners-Lee publica os princípios que regeriam o que conhecemos sobre a web estruturada. Junto a isso, também postulou o princípio de como os editores destes dados deveriam começar a perceber a ”*Data Web*[6]”. Enquanto os *links* na Web dos hipertextos estão conectando documentos escritos em HTML, a Web dos Dados utiliza relações semânticas para conectar conceitos ou objetos (coisas) descritos em RDF. As URIs identificam um objeto ou conceito na Web de Dados.

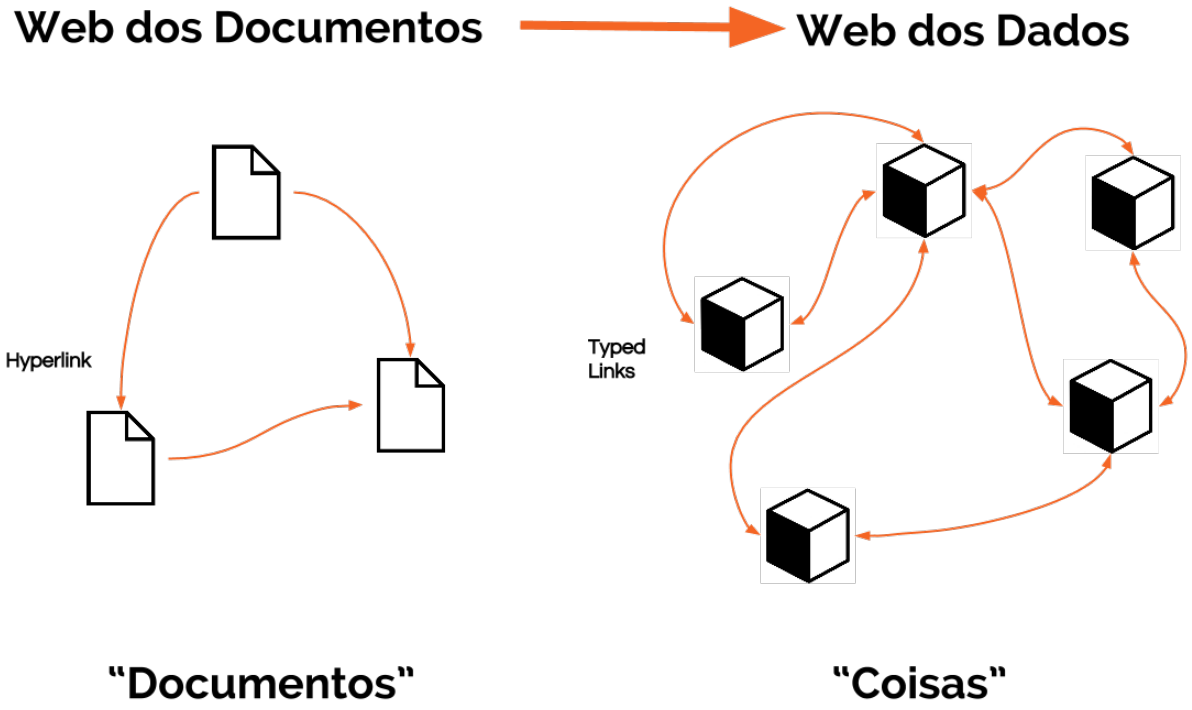

Figura 7: Web dos Dados: A conexão da-se através de conceitos que permite que uma base X reutilize conceitos definidos em base Y

Por tanto, o que estamos construindo é a semântica sobre o princípio dos documentos da web. Com isso, os dados podem ser acessados usando a arquitetura Web (URI), já consolidada, permitindo com que estes se relacionem tal como os documentos (veja Figura 7). Este processo, diferente do *hyperlink*, cria uma plataforma comum que oferece um meio de reutilização de dados além dos seus domínios.

[6]“Data Web” ou “Web de Dados” é como Berners-Lee referece-se a web estruturada

2.2.1 Ontologias e Bases de Conhecimento

Ontologia é esse meio pelo qual se especifica essas conceitualizações na Web de Dados, ou seja, é uma descrição de conceitos e relacionamentos que existem entre eles (conforme apresentado na Figura 7). A origem do nome é o ramo da metafísica[7] que estuda os tipos de coisas que existem no mundo. A palavra é derivada do grego *ontos* (ser) e *logos*(palavra) . Entretanto, seu termo de origem é a palavra aristotélica "categoria", termo utilizado no sentido de classificação (ALMEIDA; BAX, 2003).

Para BORST (1997) uma ontologia pode ser definida como uma especificação *formal* e *explicita* de uma *conceitualização compartilhada*, onde a especificação formal quer dizer algo que é legível para os computadores, explicita os conceitos, as propriedades, as relações, as funções, as restrições e as axiomas explicitamente definidos. A conceitualização representa um modelo abstrato de algum fenômeno do mundo real e compartilhada significa conhecimento consensual.

Como pode ser visto na Figura 8, a ontologia formaliza a situação de uma aluna matriculada no curso "Tecnologia da Web Semântica". Para este exemplo temos como classes: *Semester*, *Enrollment*, *Course* e *Studant*. Já as instâncias são responsáveis pela individualização e são neste exemplo: "Jéssica Helena", "Web Semântica", "matrícula 1"e "semestre 2015/1".

Quando uma ontologia é específica de um domínio, chamamos de ontologia de domínio. Este tipo de ontologia é também conhecida como bases de conhecimento. Hoje, há um grande volume destas ontologias de domínio fazendo parte da rede de dados abertos, como pode ser visto na Figura 9.

A *LOD Cloud*[8] é possível conhecer diversas bases de conhecimento com diversos domínios. Destacamos duas bases de conhecimentos que nos foi possibilitado conhecer através desta rede e que influenciam este trabalho: A WordNet que é um base de conhecimento semântica sobre linguas e a DBpedia que tem com objetivo extrair informação estruturada da Wikipedia.

2.2.2 Linguagens para Representação

Compreendido que a Web dos Dados utiliza as conexões para ligar conceitos descritos em domínios diferentes e vendo na seção anterior que uma ontologia é composta de diversos elementos que auxiliam na compreensão da conceitualização do domínio tratado. Ainda é necessário compreender como formalizar estes elementos (junto ao metadado) e permitir que o conhecimento seja reutilizado. Para isto, é preciso utilizar linguagens de construção que possam ser interpretadas por sistemas ou

[7] É a ciência do ser como ser, ou dos princípios e das causas do ser e de seus atributos essenciais (ALMEIDA; BAX, 2003)

[8] *Linking Open Data cloud diagram* é um diagrama das bases de conhecimento abertas que pode ser acessada pelo link http://lod-cloud.net/

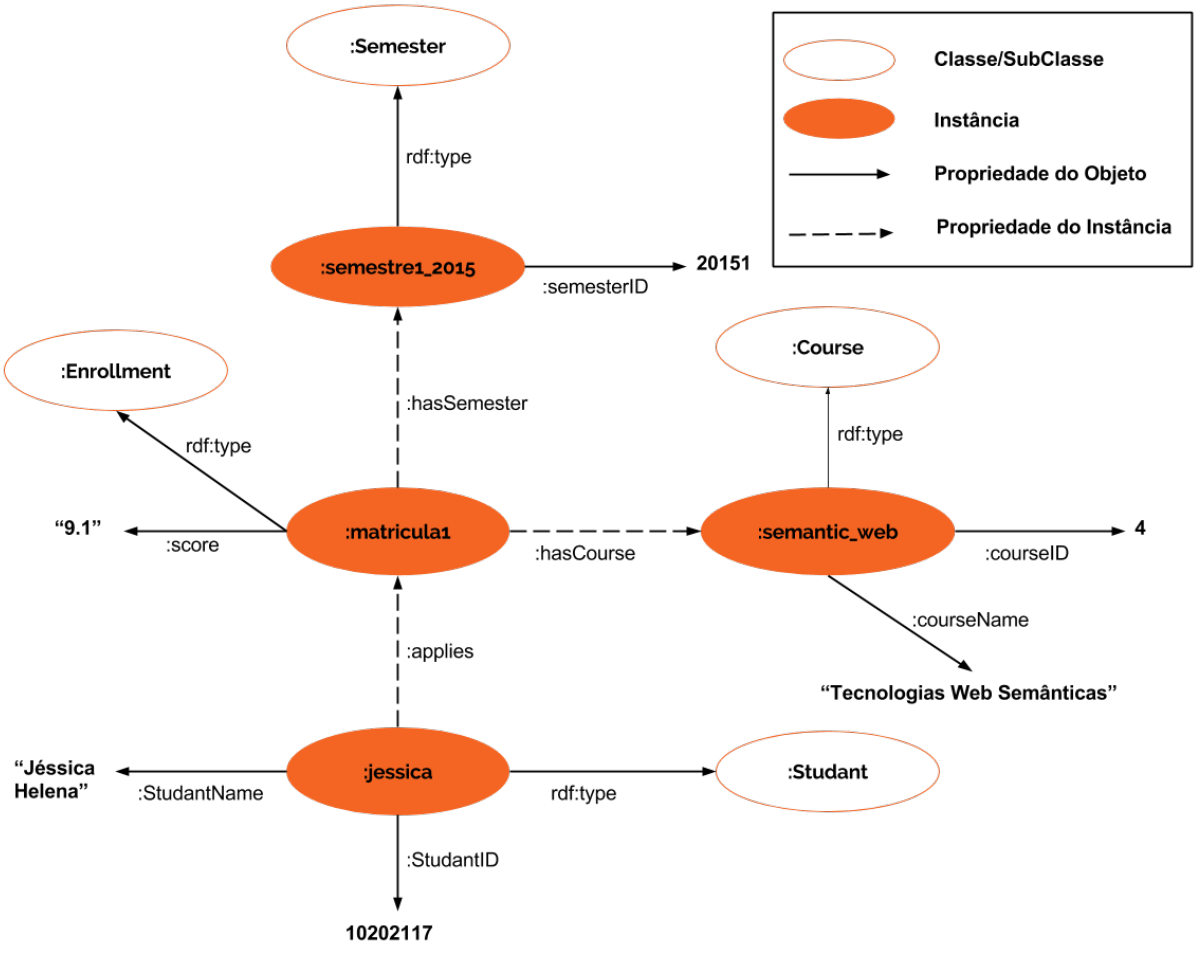

Figura 8: Ontologia: aplicação hipotética para descrição de uma disciplina em um curso

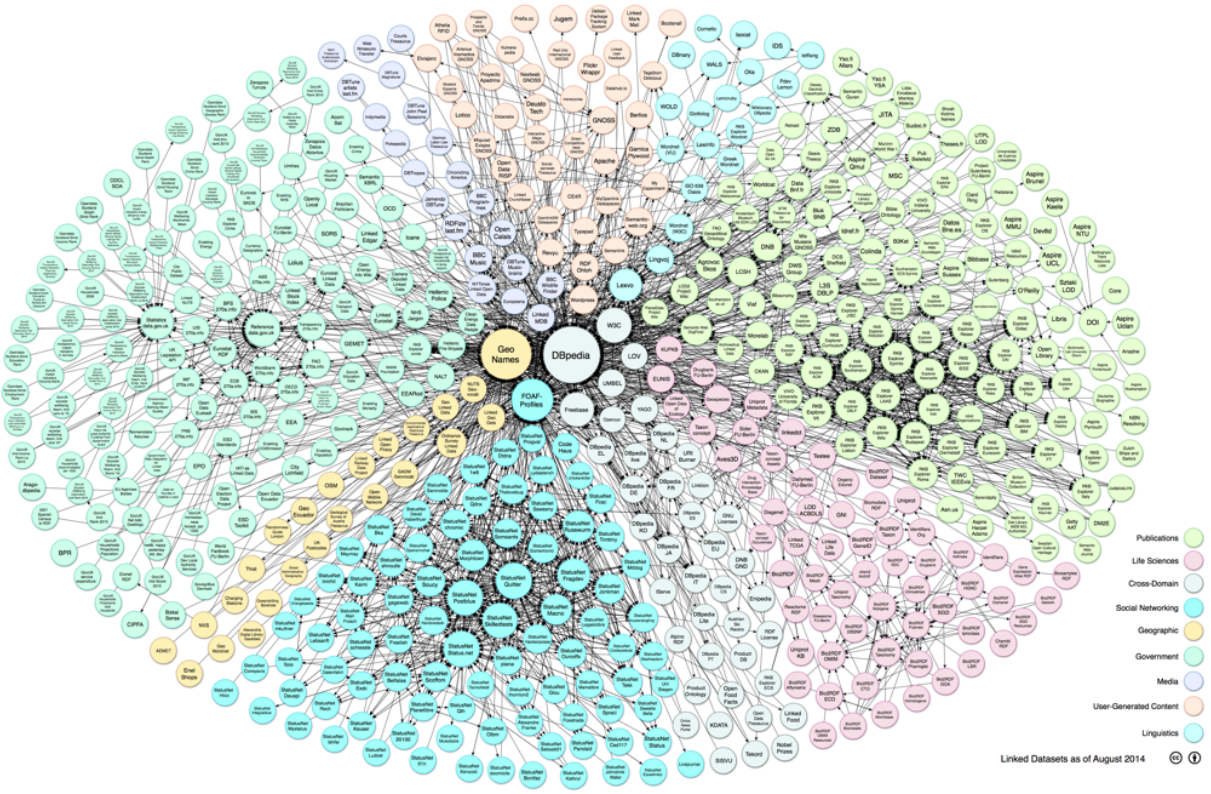

Figura 9: Núvem de relação da *Linked Open Data*. Fonte: LOD Cloud

por pessoas (DIAS; SANTOS, 2013).

Neste contexto, várias linguagens surgiram com o propósito de codificar as ontologias. Isto culmina em um impasse: havendo uma infinidade de sistemas construídos com tecnologias diversas, devemos encontrar uma linguagem que transpasse a todos e, sendo assim, reconhecida por todas estas tecnologias, é claro não trata-se de uma tarefa fácil. Além, é claro que esta linguagem escolhida deve ser capaz de ser aplicada a toda e qualquer tipo de conhecimento ou domínio. Portanto, a diversidade de propostas é inevitável.

A princípio a linguagem HTML foi considerada, porém, apesar de ser uma linguagem de marcação ela não se enquadra por duas limitações chaves: falta de estrutura e impossibilidade de validação de informações exibidas (FENSEL et al., 2011).

2.2.2.1 XML: Extensible Markup Language

O XML é uma linguagem de marcação extensível que permite dividir o documento em partes identificáveis. Lançada em 1998 pelo consórcio W3C[9] a linguagem de marcação tinha como objetivo suprir algumas demandas que o próprio HTML não conseguia. O fator que realmente diferencia XML de outras linguagens deste mesmo tipo é sua capacidade extensiva, pois provê um formato de dados para estruturação de documentos sem necessidade de uso de um vocabulário específico (DIAS; SANTOS,

[9]World-Wide Web Consortium

Sujeito	Predicado	Objeto
<http://www.w3.org/People/EM/contact#me>	<http://www.w3.org/2000/10/swap/pim/contact#fullName>	"Marcos Somer"
<http://www.w3.org/People/EM/contact#me>	<http://www.w3.org/2000/10/swap/pim/contact#mailbox>	mailto:marcos.s(at)examplle
<http://www.w3.org/People/EM/contact#me>	<http://www.w3.org/2000/10/swap/pim/contact#personalTitle>	"Dr."
<http://www.w3.org/People/EM/contact#me>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/2000/10/swap/pim/contact#Person>

Tabela 2: Representação de tripla RDF

2013).

2.2.2.2 RDF: Resource Description Framework

A linguagem RDF oferece um modelo de representação simples e flexível, que permite a interpretação semântica do conhecimento, com a utilização de conectivos lógicos de negação, disjunção e conjunção. Ela é composta por duas partes: RDF e RDF-Schema. A primeira define como descrever recursos através de suas propriedades e valores, enquanto a segunda define propriedades específicas e restringindo sua utilização.

```
<Class ID="Female">
    <subClassOf resource="#Animal"/>
    <disjointWith resource="#Male"/>
</Class>
```

Figura 10: Especificação em RDF/XML

Sendo assim, o RDF é uma arquitetura de metadados cujo maior objetivo é definir um mecanismo de descrição de documentos que não esteja vinculado a nenhum domínio de conhecimento específico resolvendo o problema levantado no item 2.2.2. Os conceitos que fundamentam RDF são os vocábulos baseado em URIs. Conceitualmente, o modelo gráfico é dado pelo sujeito, predicado e objeto, que juntos formam uma tripla (Veja Tabela 2). Sendo o objeto um recurso da web que está sendo descrito, e deve ser identificado através da sua URI. O predicado indica uma propriedade, isto é, um aspecto, uma característica, atributo ou relação atribuída ao recurso pela sentença. Sujeito é a parte que identifica o objeto da declaração (FENSEL et al., 2011, p. 92).

2.2.2.3 OWL: Ontology Web Language

A *Ontology Web Language*(OWL) serve como uma linguagem para ser utilizada quando as informações contidas em documentos web, precisam ser processadas por aplicações em situações em que seu conteúdo não apenas será visível para humanos. Pode ser usada para representar explicitamente o significado de termos em vocabulários e os relacionamentos entre os termos. Esta é um aperfeiçoamento das linguagens XML, RDF e RDF(S) com um maior poder de expressão (FENSEL et al.,

2011, p. 33).

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <!-- OWL Class Definition Example -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/plants#planttype">
    <rdfs:label>The plant type</rdfs:label>
    <rdfs:comment>The class of plant types.</rdfs:comment>
  </owl:Class>
</rdf:RDF>
```

Figura 11: Especificação em OWL

2.3 Tecnologias

Para a concepção deste trabalho foram utilizados um conjunto de tecnologias com o propósito de auxiliar o desenvolvimento. A escolha delas deu-se por estarem já referenciadas nas bibliografias, em trabalhos relacionadas ou como recurso oferecido dada a familiarização com estas tecnologias.

Elas agrupam-se em dois conjuntos, em que o primeiro conjunto está a linguagem de programação Python[10] utilizado nas etapas anteriores, a etapa de Indexação e Ranqueamento propostas no capítulo seguinte e a linguagem de Programação Ruby[11] junto ao *framework* Ruby On Rails[12] que servem de subsídio para a interface Quantum e o SGBD não relacional MongoDB[13] para a armazenamento dos currículos que permeia todos as etapas do processo.

No segundo conjunto estão as tecnologias essenciais para a idealização e concepção do processo aqui apresentado a seguir e estão dispostas na Figura 12 para a visualização de como estas interagem com as etapas para solucionar o problema central deste trabalho.

2.3.1 SLattes: Semantic Lattes

O Slattes relaciona-se com este trabalho por promover uma base de conhecimento semântica e ele é uma transformação XSLT[14] que processa arquivos de Currículos Lattes e gera o dado RDF sobre a Ontologia VIVO-ISF. O processo é dado da seguinte forma: um interpretador como o **xsltproc**[15] chama um arquivo de transformação XSLT para ser aplicado em um arquivo XML e a partir desse processo é gerado um terceiro documento que pode ser um arquivo de texto, XML, PDF, HMTL ou RDF por exemplo.

A Slattes é parte de um projeto da FGV[16] que visa a integração e extração

[10] http://www.python.org
[11] http://www.ruby-lang.org/
[12] http://rubyonrails.org
[13] https://www.mongodb.org
[14] eXtensible Stylesheet Language for Transformation
[15] http://xmlsoft.org/XSLT/xsltproc2.html
[16] Fundação Getúlio Vargas

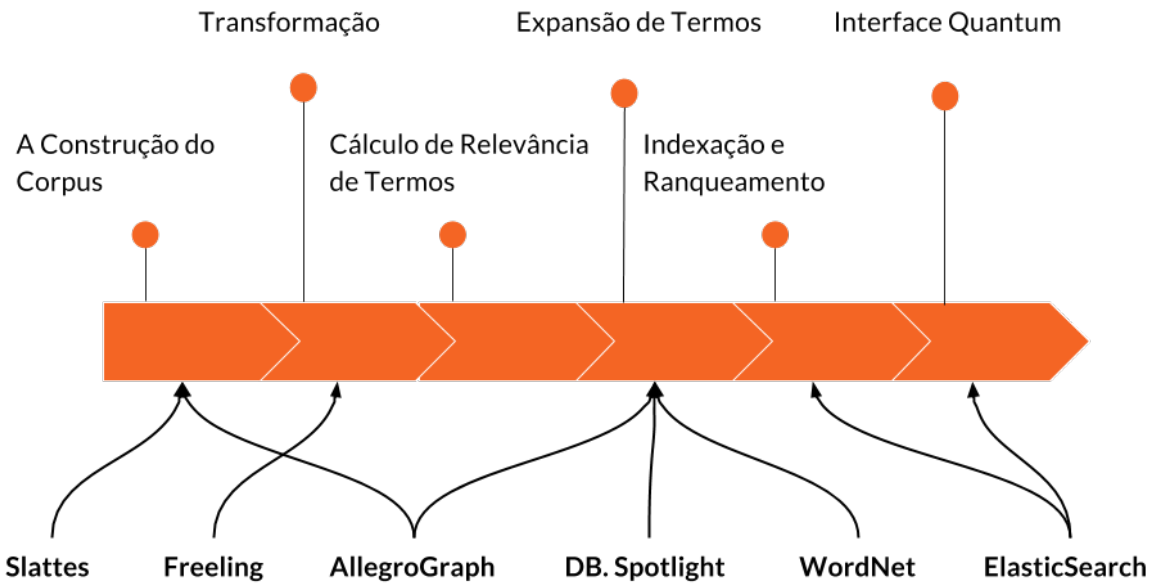

Figura 12: Tecnologias envolvidas em cada uma das etapas da metodologia adotada

de informações de publicações, interesses e todas as atividades acadêmicas dos pesquisadores brasileiros (RADEMAKER; HAEUSLER, 2013). O Slattes é uma transformação específica para ser aplicada aos arquivos XML, exportados pela PL, com o intenção de gerar informação baseado na Ontologia VIVO-ISF[17]. Como parte dos esforços do projeto, o Slattes nasceu como meio para promover uma transformação para a Plataforma VIVO onde este mecanismo promove um interface via navegador que permite ser possível buscar informações de forma mais simples sobre artigos, projetos de pesquisa, cursos acadêmicos entre outros. Como também prover mecanismo de avaliação, para a administração da fundação, do retorno e performace dos investimentos da a instituição em projetos de pesquisa.

2.3.2 AllegroGraph

AllegroGraph RDF Store[18] é um sistema para carregamento, armazenamento e busca em dados RDF. Ele inclui uma interface de busca SPARQL e um *reasoner* RDFS[19] que permite ser usado como processador de RDFS. Ele é desenvolvido em Common Lisp com clientes em Java e Python. Apesar de ser uma solução comercial há uma licença gratuita para até 5 milhões de triplas.

Sobre a ferramenta de armazenamento há uma série de serviços, como por exemplo de indexação e otimização, que servem de base a serviços de protocolo de troca de mensagens especializados (Direct Server, HTML Server, SPARQL Protocol) dos

[17] https://wiki.duraspace.org/display/VIVO/VIVO-ISF+Ontology
[18] http://www.franz.com/products/allegrograph/
[19] Ferramentas que executam tarefas de raciocínio baseado em RDFS

quais são meios de comunicação para outras linguagens de programação.

O AllegroGraph é um importante repositório de RDFs focado em alto desempenho e sua escolha deu-se, entre outras características, por possuir um cliente na linguagem Python (Figura 13) na qual foi desenvolvida a ferramenta de expansão aqui proposta.

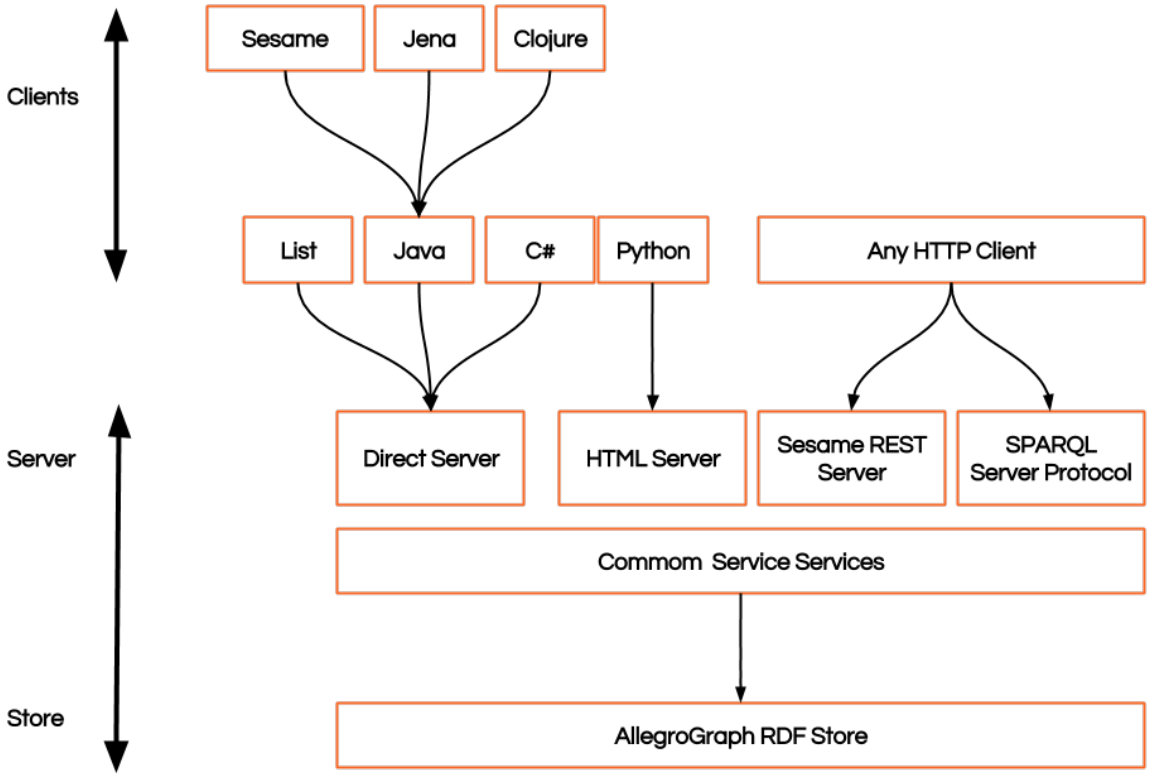

Figura 13: Estrutura de Armazenamento de RDFs no AllegroGraph. Fonte: Allegrograph

2.3.3 WordNet

As Wordnets são também chamadas de Ontologias Lexicais dada que as relações de hiperonímia e hiponímia podem ser vistas como categorias de especialização entre os conceitos. Utilizada com inúmeros pesquisadores da área de Processamento de Linguagem Natural a WordNet é um importante mecanismo utilizado para diversas atividades entre elas inclusão de desambiguação de sentido em palavras, sistemas de informação, classificação de textual entre outros. Ela também faz parte dos bancos de conhecimento lexicais que têm como objetivo armazenar e relacionar recursos lexicais semântico (FELLBAUM, 2009) e muitas vezes é visto como a combinação de um dicionário e um tesauro[20].

[20]Tesauro é uma lista de palavras com significados semelhantes

Classe	Definição
Substantivos	
Hiperonímia	**Y** é um hiperônimo de **X** se todo **X** é parte de **Y** (“Canino” é um hiperônimo de “Cachorro”)
Hipônima	**Y** é uma hipônima de **X** se todo **Y** é um tipo de **X** (“Cachorro” é um hipônima de “Canino”)
Meronímia	**Y** é meroníma de **X** se **Y** é parte de **X** (“Janela” é meroníma de “Prédio”)
Holonimia	**Y** é uma holonimía de **X** se **X** é parte de **Y** (“Prédio” é holonimía de “Janela”)
Termo Coordenada	**Y** é um termo coordenado de **X** se **X** e **Y** compartilham um hipernônimo. (“Lobo” é um termo coordenado de “Cachorro” e “Cachorro” é um termo coordenado de “Lobo”)
Verbos	
Hiperonímia	O verbo **Y** é uma hiperônima do verbo **X** se a ação **X** é parte de **Y** (“Perceber” é uma hiperônima de “Escutar”)
Troponímia	O verbo **Y** é topónimo de **X** se a ação **Y** está fazendo **X** de alguma maneira (“Balbuciar” é topónimo de “Falar”)
Implicação Lógica	O verbo **Y** is implica em **X** se fazendo **X** você deve estar fazendo **Y** (“Dormir” implica em “Roncar”)
Termo Coordenado	Ambos verbos que compartilham em comum uma hiperônima (“balbuciar” e “gritar”)

Tabela 3: Classes de Significados Semânticos

As palavras na wordnet se relacionam através das categorias lexicais: Substantivos, Verbo, Adjetivo e Advérbio. Já as palavras da mesma categoria que possuem algum grau de sinonima são agrupados em *synsets*. Palavras da mesma categoria, que são mais ou menos lexical, sinónimos, são agrupados em *synsets*. Note que cada *synset* contém todas as lexicalizações de um único conceito e constitui um nó da rede. Um exemplo prático são as expressões “carro” e “automóvel”, por exemplo, estão ambas incluídas no mesmo *synset*, já que são lexicalizações do mesmo conceito.

Observemos que na expressão “Acabou a **linha** de costura” e na expressão “Qual linha de ônibus devo pegar?” a palavra “linha” possui polissemia onde um significante, neste caso a palavra “linha”, serve de base para mais de um plano de conteúdo, neste caso o significado, que na primeira expressão a linha está ligada ao de fio enquanto no segundo a palavra linha está ligada ao significado de trajetória.

Para MARRAFA et al. (2005) diferentemente dos dicionários tradicionais que o significado lexical é definido por paráfrases para relacionar as palavras, nas wordnets o sentido emerge das relações lexicais e conceptuais que se estabelecem entre elas. Há também a representação semântica entre os *synsets* pertencentes a classes de Substantivos e Verbos, onde as principais relações são descritos na Tabela 3 .

2.3.3.1 openWordnet-PT

A escolha do uso de WordNet da-se pelo poder de expressão na relação entre palavras que serve então de base para a expansão de termos proposta por este trabalho.

Mais especificamente para esse trabalho está se usando a wordnet openWordnet-PT (PAIVA; RADEMAKER; MELO, 2012) desenvolvida inicialmente na FGV com colaboradores e que pode ser acessada[21] ou baixada[22] e instanciada em uma repositório de RDF como o AllegroGraph (Item 2.3.2) e a partir disso realizar consultas utilizando SPARQL.

2.3.4 *Freeling*: Serviços de Análise de Linguagem Natural

Freeling é uma biblioteca *open-source* que provê serviços básicos de Processmaento de Linguagem Natural entre outras funções para desenvolvedores de aplicações de NLP[23]. A sua escolha deu-se por prover o *sense* necessários para a etapa 3.4.1 (A etapa de casamento entre termo e sentido será melhor descrito mais a adiante no item 2.3.4.5). Também observamos que esta ferramenta automatiza a análise morfológica empregada na etapa 3.2 de forma satisfatória.

O *Freeling* foi primeiramente apresentado no evento LREC'04 como uma pacote de ferramentas de análise sobre a Licença GNU Lesser General Public License e provia análises morfológica e *PoS-Tagging* em espanhol, Inglês e Catalão (CARRERAS et al., 2004). Na sua versão 1.3 houve diversos acréscimos sobre a melhoria de análise e um módulo de classificação de Entidade Nome e anotação sintática do WordNet como desambiguador de sentido semântico. Atualmente o *Freeling* encontra-se na versão 3.1 com suporte a 11 línguas entre elas o Inglês, Catalão, Francês, Italiano, Português, Espanhol e Russo.

Também é uma ferramenta que permite o acesso a 18 serviços de análise entre eles o de Tokenização, Quebra de Sentença, Deteção de Número, Quantidade e Data, Dicionário Morfológico, Regras de Afixos, Deteção de Entidade-Nome, *PoS tagging*, Anotação WordNet e Desambiguação UKB de *sense* (PADRÓ; STANILOVSKY, 2012). Além de incluir como base a anotação dos dicionários baseados no WordNet e usa a árvore de relação de sinonímia entre as palavras, quando disponível, para a língua em questão.

Todas estas característica acima citas são de vital importância para a atomização de passos realizados na etapa 3.2. Este trabalho vale-se dos serviços abaixo descritos.

2.3.4.1 Módulo de Identificação de Símbolos

Este módulo também chamado de [24] é o primeiro pelo qual passa o texto e tem como objetivo transformá-lo em um vetor de objetos. Posteriormente esse vetor é passado por regras de tokenização através de expressões regulares com objetivo de

[21] http://wnpt.brlcloud.com/wn/
[22] https://github.com/own-pt/openWordnet-PT
[23] Natural language processing
[24] Tokenizer Module

extrair o *token*. As regras são divididas em três sessões: Macros, Expressões Regulares e Abreviações. A primeira permite ao usuário definir as macros, que por exemplo definem o dicionário alfanumérico. Definidas as macros é possível, sobre elas criar Regras e Expressões Regulares e por fim a seção de abreviações define abreviações comuns que não devem ser separadas do ponto seguinte (por exemplo etc., sra.).

2.3.4.2 Módulo de Divisão

Splitter Module ou Módulo de Divisão recebe o vetor de objeto-texto produzidos pelo Módulo *Tokenizer* e isola-os até detectar um limite de frase. Em segunda, uma lista de objetos de sentença é retornado. Para isso são usados marcadores ($<$SentenceStart$>$e $<$SentenceEnd$>$) para identificar elementos que iniciam e terminam possivelmente uma sentença.

2.3.4.3 Módulo de Análise Morfológica

Módulo de Análise Morfológica ou *Morphological Analyzer* de fato trata-se de um meta-módulo por não realizar nenhum processamento por si só. Mas uma abstração para a chamada de 8 sub-módulos de acordo com a necessidade. A Tabela 4 trás a definição de cada uma delas.

Ao final da análise morfológica os lemas extraídos pelo processo acima mencionado, passa então para os próximos passos onde será realizado a vinculação de possíveis *PoS-Tag* e encontro do *sense* adequado.

2.3.4.4 Módulo de Atribuição por Probabilidade

Este módulo[25] finaliza a cadeia de análise morfológica e tem como objetivo duas funções: Caso a análise dos passos do módulo anterior não seja positiva, este módulo tenta a atribuição através de probabilidade para as palavras de algum das morfologias da Tabela 4. A segunda função deste módulo é então tenta encontrar quais são os prováveis Pos-Tag da palavra baseada na terminação da palavra.

2.3.4.5 Módulo de Rotulagem de Sentido

Este módulo é chamado de *Sense Labelling Module* e está responsável pelo encontro do lema de cada análise no dicionário de sentido e enriquece a análise com a lista de todos os sentidos encontrados. Para isso o *Freeling* permite a configuração o dicionário de sentido para a utilização e as regras de mapeamento para ser utilizadas para adaptar as Pos-Tag para aqueles utilizados no dicionário sentido.

[25]"Módulo de Atribuição por Probabilidade" é uma tradução livre pelo autor para o nome "Probability Assignment and Unknown Word Guesser Module" encontrado na documentação da ferramenta

Módulo	Descrição da Função
Number Detection	O módulo é baseado em um autômato de estado finito que reconhecem a validade da expressão numérica e detecção de números de acordo com a linguagem, por exemplo “vinte e cinco” e “25” são mareados para o mesmo lema.
Punctuation Detection	Basicamente detecta todas as pontuações que ocorrem no texto e cria uma tag lema para cada símbolo de pontuação
User Map	Permite ao usuário customizar lemas, criando os seus próprios, por exemplo “<.*>XMLTAG Fz” é uma regra que pode ser adicionada para reconhecer marcadores de texto e atribuir o lema Fz a esses tokens.
Data Detection	Assim como o módulo “Number Detection” é um autômato de estado finito que tem por objetivo detectar datas de acordo com a lingua analisada. Padrões como DD-MM-AAA são detectados.
Dictionary Search	Ele possui duas funções: encontrar formas de palavras para os seus lemas e PoS Tags e adicionar e remover regras de afixos para encontrar palavras não canonicais.
Multiword Recognition	Módulo responsável pela detecção de n-gramas (palavras de múltiplas lemas) gerando um n-grama por tinha como por exemplo “a_buenas_horas a_buenas_horas RG A”
Named Entity Recongnition	Composto de dois módulos independentes chamados de acordo com o arquivo de configuração, o primeiro detecta por sequência de palavras capitalizadas, como por exemplo o tri-grama “Banco do Brasil”. O segundo trata-se do Bioner ,um detector de nome de entidades baseado em aprendizado de máquina.
Quantity Recognition	Módulo com função de reconhecer quantidades como por exemplo: Raio, porcentagem, medidas físicas, correntes magnéticas e é fortemente dependente do módulo Number Detection dado que quando um número não é detectado a quantidade não é reconhecida.

Tabela 4: Freeling: 8 sub-módulos da análise morfológica. Fonte: (PADRÓ; STANILOVSKY, 2012)

2.3.4.6 Módulo de Desambiguação de Sentido

Este módulo apenas é chamado quando se faz necessário a desambiguação de palavras. Isto quando a execução a ítem 2.3.4.5 atribui mais de um sentido possível a palavra. Este módulo é uma implementação do algoritmo conhecido como UKB[26], ele conta com uma rede de relação semântica para desambiguar os sentidos mais prováveis para as palavras em um determinado texto. Ele vale-se do algoritmo *Page-rank* para desambiguação de sentido de palavras.

O algoritmo em questão funciona da seguinte forma: Um construtor de conhecimento linguístico é formado pelo conjunto de conceitos e relações entre as palavras, e um dicionário, isto é, uma lista de palavras (tipicamente, lemas palavra) cada um deles ligado a pelo menos um conceito de LKB. Dada tal LKB, nós construímos um grafo não direcionado $G = (V, E)$, onde nós representam conceitos LKB (*vi*), e cada relação entre conceitos *vi* e *vj* é representado por um vértice unidirecional $e_{(i,j)}$. A partir desse grafo é então extraído o sub-grafo G_d de interesse cuja vértices são as relações que se dá entre as palavras do grupo de palavras de interesse.

Esse então é o sub-grafo de uma palavra, o processo repeti-se para todos os conceitos de todas as palavras no contexto e ao final temos um grafo resultante da soma destes processos. Ao final o que tem-se interesse está nos menores caminhos entre a palavra em questão com todas as demais do contexto. Uma vez que o gráfico G_d é construído, calcula-se o algoritmo PageRank tradicional sobre ele. A intuição por trás deste passo é que os vértices representam os conceitos correção será mais relevante em G_d do que o resto dos possíveis conceitos das palavras de contexto, o qual deve ter menos relações, em média, e ser mais isoladas.

2.3.5 DBpedia Spotlight API

Sendo um dos mais conhecidos projetos da iniciativa *Linking Open Data*, a DBPedia é um esforço colaborativo realizado para a construção de uma ferramenta web que permita a extração de informação estruturada da Wikipedia, convertendo-as em RDF e tornando-as disponível na web. E assim permitindo a realização de buscas semânticas sobre as propriedades e relações a base de dados da Wikipedia.

O Spotlight API DBpedia, por fim, trata-se é uma ferramenta API RESTful para mencionar anotações de recursos DBpedia em textos, ele recebe como entrada texto puro e gera texto anotado. Sua escolha e relação com este trabalho dar-se por esta característica de permitir que de textos de títulos de artigos possam ser extraídos entidades semânticas. Fornecendo então um meio para a ligação de fontes de informação não-estruturados para a nuvem *Open Linked Data*. Assim a DBpedia Spotlight realiza a extração de entidade, incluindo a detecção e resolução de nomes entidade

[26]disponível em http://ixa2.si.ehu.es/ukb/

(MENDES et al., 2011). Hoje a ferramenta é disponível gratuitamente, encontra-se disponível para acesso[27] e execução através de um *browser*, entretanto também disponível para ser baixada[28] e executada de forma autônoma.

O processo adotado pela ferramenta para a transformação do texto em texto anotado passa por três etapas: A primeira etapa (*Spotting*) é o encontro de possíveis candidatos para a anotação, este processo é então o encontro de possíveis entidades. Este processo dar-se através do algoritmo LingPipe[29] que é baseado no algortimo Aho-Corasick dado que este permite a localização de palavras chaves em textos a partir de um tempo linear.

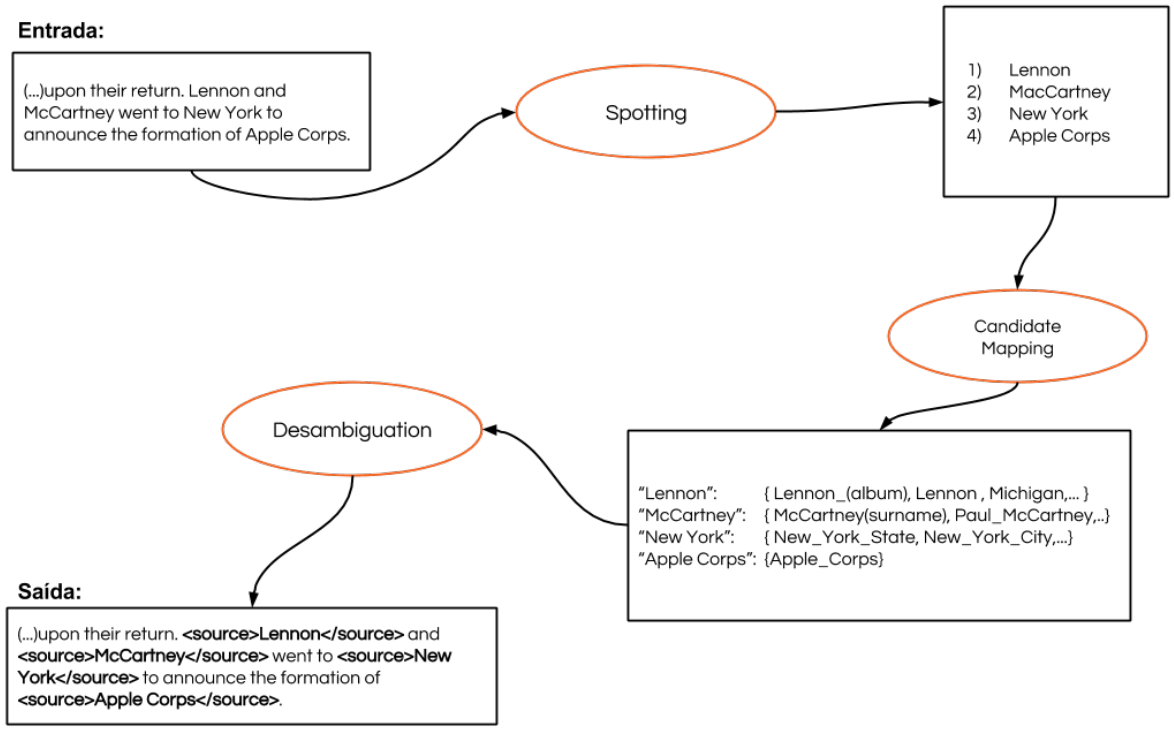

Figura 14: Spotlight: Transformação de texto plano para texto anotado

O segundo passo (*Candidate Mapping*) então é, a partir das palavras chaves retornada pelo algoritmo do passo um, quais são os possíveis *sources* encontrados na DBpedia para aquele candidato. A escolha dos possíveis *sources* se dá então pelas fontes que possuem o mesmo nome que o possível candidato a anotação, *sources* que apesar de não ter os mesmos nomes, possuem alias e formas de fala como o candidato a anotação ou *sources* de uma página de desambiguação da wikipedia.

Por fim a terceira passo (*Desambiguation*) é a escolha do *source* a ser anotado ao texto. Esta etapa, caso o candidato a anotação receba mais de um possível *source*,

[27] http://dbpedia-spotlight.github.io/demo/
[28] https://github.com/dbpedia-spotlight/dbpedia-spotlight/releases
[29] http://alias-i.com/lingpipe/

este passa por um processo de decisão baseado em qual dos *sources* possui o melhor *score*[30] de contexto com o texto. A ferramenta também permite configurar a confidência de desambiguação pelo qual o processo acima descrito é realizado.

Sendo o parâmetro de confiança, variante na escala de 0 a 1, da anotação realizada por DBpedia Spotlight este parâmetro tem como propósito apontar a pertinência do tópico a ambiguidade contextual. A definição de um limiar de confiança elevado pela DBPedia Spotlight evita anotações incorretas. A ferramenta estima esse parâmetro em um conjunto de desenvolvimento de 100.000 amostras Wikipédia. A lógica é que um valor de confiança de 0,7 irá eliminar 70% dos casos de teste incorretamente desambiguados.

2.3.6 ElasticSearch

O *ElasticSearch*[31] é um importante servidor de busca hoje no mercado, sua primeira versão foi lançada em 2010. O servidor de busca é baseado no Apache Lucene que naturalmente importa suas propriedade de serviço de busca distribuído. O serviço provido por ele é guiado através de uma API RESTful que recebe requisições através do protocolo HTTP, no formato JSON. Atualmente o servidor de busca está disponibilizada sobre os termos da Licença Apache KUC; ROGOZINSKI (2013).

Algumas características que nos fizeram escolher tal ferramenta como substituto a uma implementação são elencadas abaixo:

É distribuído gratuitamente sobre a licença Apache 2;

Arquitetura distribuída: Permitindo distribuir os documentos a medida que for sendo necessário se ter mais capacidade, basta adicionar mais nós e deixar que o *cluster* reorganize-se para aproveitar melhor o hardware extra;

Escalabilidade: Os *clusters* são flexíveis, detectando e removendo nós com falhas, reorganizando-se para garantir que os dados estejam seguros e acessíveis;

Suporte ao modelo utilizados no ítem 3.5;

Quase qualquer ação pode ser executada através da sua simples API RESTful, utilizando JSON via HTTP. Além disso possui API de integração com as mais variadas linguagens de programação.

2.4 Trabalhos Relacionados

Nesta seção são apresentados os trabalhos relacionados. O primeiro é uma ferramenta que está ligadas diretamente a etapa de extração dos dados do Lattes: Script

[30] O *score* é calculado baseado TF-ICF que traduz a relevância do termo no contexto da DBpedia (MENDES et al., 2011, p. 6)

[31] https://www.elastic.co/products/elasticsearch

Lattes. Em seguida, são apresentados outros dois que atuam como SRI para Lattes. Por fim, mais dois sobre a utilização de expansão de termos usando ontologias para a melhoria de buscas.

O scriptLattes[32] é uma ferramenta aberta[33] que permite, através de requisições web, baixar em formato HTML (HyperText Markup Language) os currículos lattes, bem como a partir disso gerar automaticamente outras páginas HTML com as listas de produções, vários gráficos de co-autoria entre membros do grupo de interesse e pode ser visto como uma forma de extrair informações sobre os trabalhos de docentes sendo assim uma alternativa a coleta de dados dado que neste trabalho os arquivos foram obtidos com a própria instituição.

Outro trabalho relevante para este TCC é o de SOUZA MEIRELES (2014) onde ele propõem um SRI para a PL chamado de AcademicS, no qual, o motor de busca proposto por ele, apresenta resultados promissores, porém, ao final o autor levanta problema de ranqueamento com o objetivo de melhorar a qualidade (relevância) dos currículos (SOUZA MEIRELES, 2014, p. 86). Este relaciona-se diretamente dado que o trabalho aqui, apenas de não aproveitar a posposta do autor, tenta solucionar o problema de relevância aproximando o vocabulário dos currículos com a dos usuários do motor de busca.

O próximo trata-se de uma referência na área do WordNet FELLBAUM (2009) em que duas partes do conjunto da obra se destacam e relacionam-se intimamente com este trabalho, o primeiro deles é o da LEACOCK; CHODOROW (1998, p. 265) intitulado como “Combining Local Context and WordNet Similarity for Word Sense Indentification” que demonstra meios para a obtenção da combinação local de contexto com a identificação de WordNet Sense mais adequada.

A segunda parte que se destaca desta obra e que se relaciona com este TCC trata-se do trabalho de (VOORHEESS, 1998, p. 285) “Using WordNet for Text Retrieval” em que a autora aponta duas abordagens para o processo de automatização com o objetivo de melhorar a precisão da seleção de *synsets* para a expansão de termos. Ambos os trabalhos de FELLBAUM (2009) relacionam-se ao permitir a esta ferramenta encontrar um meio pelo qual pode-se expandir os termos.

Ao analisar os quatros trabalhos acima mencionados fica evidente a importância e necessidade de que há em extrair e coletar informações da PL. O trabalho de SOUZA MEIRELES (2014) lança uma grande luz sobre o desafio de obter uma melhor experiência na extração destas informação e também uma alternativa a motor de busca, hoje oferecido pelo PL, que onera o usuário e nem sempre retorna um resultado satisfatório. Ao final do trabalho o autor ressalta que melhorias podem ser realizadas a respeito de como obter um aumento na qualidade (relevância) dos currículos que

[32] http://scriptlattes.sourceforge.net

[33] A ferramenta encontra-se sobre a licença GNU General Public License

possuem a informação na qual o usuário necessita (SOUZA MEIRELES, 2014, p. 86). Ao analisar o trabalho de VOORHEESS (1998) e LEACOCK; CHODOROW (1998) fica evidente que ambos podem ser fontes de conhecimento para a solução do problema proposto neste TCC.

3 CONCEPÇÃO DA FERRAMENTA DE EXPANSÃO DE TERMOS

Após a revisão teórica proporcionada pelo capítulo anterior, este é dedicado a concepção da Ferramenta de Expansão de Termos e também faz parte deste capítulo a modelagem da interface do SRI utilizado pelo usuário. A abordagem selecionadas para este trabalho são fundamentadas na base conceitual e no escopo deste trabalho. O processo então é descrito abaixo através das sessões. Esta abordagem dá-se para que se compreenda o processo aqui proposto pela ferramenta e o funcionamento das tecnologias possam ser analisadas separadamente no capítulo acima.

Como pode ser visto da Figura 15 o processo como um todo está dividido em 6 etapas, onde a primeira refere-se a construção do *corpus*[1], o segundo passo é então a transformação da informação da base de informação, a terceira está incumbida de calcular qual termo possui relevância para que na próxima etapa, 4º passo, sejam expandido os termos. Já o 5º dedica-se a descrever a indexação e ranqueamento sugerido e por fim o 6º passo está centrado a construção da interface pelo qual o usuário interage com o SR.

3.1 A constituição do *corpus*

A etapa de coleta trata-se então do momento de constituição do *copus*[2], ou seja, da coleção de documentos (currículos) que serão empregados nesse trabalho. Esta é uma etapa opcional, caso não haja acesso direto aos documentos, e no contexto deste trabalho os documentos forma disponibilizados pela UFPel.

A coleta ocorreu em Novembro de 2015 e foram adquiridos 1995 currículos Lattes de docentes e técnicos-administrativos da instituição UFPel. Estes arquivos foram então exportados pela API da CNPq no formato XML, este formato de arquivo é largamente reconhecido pela sua estrutura em auxiliar o mapeamento e estruturação

[1] plural corpora, corpus é o conjunto de textos estruturados

[2] O termo *Corpus* usado na área de recuperação de informação, nasce da noção de *Corpus Linguístico* que é o conjunto de textos escritos e registros orais em uma determinada língua e que serve como base de análise

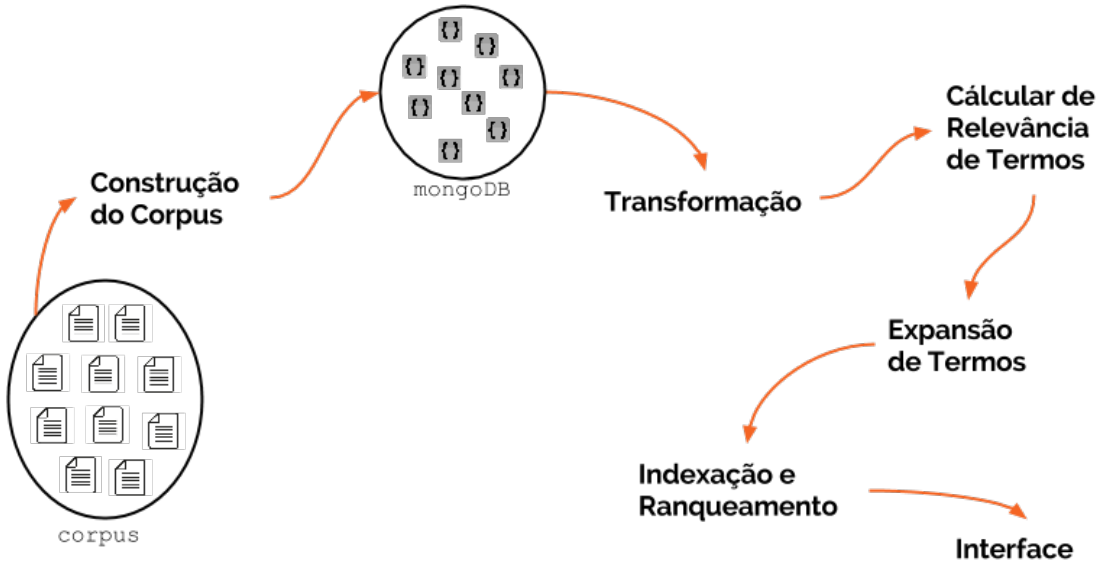

Figura 15: Ilustração do processo adotado pela ferramenta

da informação e um dos principais meios pelos quais aplicações usam para trocar informação como já foi mencionado no item 2.2.2.1.

Os documentos foram armazenados, no seu formato original em XML, junto a ferramenta com o intuito de que esses documentos pudessem ser reaproveitado e até mesmo utilizado em outros momentos de análise e avaliação desta ferramenta. Porém nas etapas de manipulação por esta aplicação o documento é resumido e transformado para as bases abaixo descritas.

Dado o enorme volume de informações que há dentro de cada documento e com o objetivo de diminuir o espaço amostral para auxiliar a melhor compreensão da ferramenta, foram selecionados os campos dos currículos que melhor descrevem e definem o pesquisador. Esta restrição tem como objetivo melhorar a fórmula de ranqueamento e formação do *corpus*.

Na abordagem evidenciou-se já que o processo de expansão dos termos (ET) não poderia ocorrer de forma arbitrária por todo o currículo, dado que este processo ocasionaria inconsistências, que serão abordados mais adiante na seção de Cálculo de Relevância de Termos (item 3.3). No contexto deste trabalho, selecionamos os 12 campos apresentados na Tabela 5.

A fundamentação na escolha dos campos: nome completo, nome de citação, produções bibliográficas, a participação em projetos, orientações e por fim o resumo(*currículo vitae*) para nos se dá pelo fato de que estes são os campos parecer representam o *Status Quo* do autor, são áreas do currículo Lattes que representam suas práticas mais recentes no meio acadêmico e atualizado com mais frequência pelos seus autores.

Esta escolha demonstrou-se fundamental para que não houvesse a expansão de-

Campos	Descrição
lattes-id	Campo Identificador Único de cada um dos pesquisadores composto por 16 dígitos
dados-gerais//nome-completo dados-gerais//nome-em-citacoes-bibliograficas	Dois campos ”nome-completo”e ”nome-de-citacoes-bibligraficas” permitem a identificação do autor
artigos-publicados//artigo-publicado artigos-aceitos-para-publicacao//artigo-aceito-para-publicacao trabalhos-em-eventos//trabalho-em-eventos	Os três campos compreendem a soma das publicações realizadas pelo autor em periódicos
participacao-em-projeto//projeto-de-pesquisa	Permite adquirir as participação do autor em em projetos
orientacoes-concluidas//orientacoes-concluidas-para-doutorado orientacoes-concluidas//orientacoes-concluidas-para-mestrado orientacoes-concluidas//orientacoes-concluidas-para-pos-doutorado orientacoes-concluidas//orientacoes-concluidas	Permite identificar os trabalhos indiretos realizados pelo pesquisador através de suas orientação
//palavra-chave-	Uma série de 1 a 6 palavras chaves cadastradas em: Produção Bibliográfica, Orientações, Produção Técnica, Livros e Capítulos etc.

Tabela 5: Campos selecionados para a formulação

masiada de termos, como por exemplo, dos registros acadêmicos mais primários do autor. Esta restrição impede a expansão da formação inicial do docente, dado que há uma propensão na graduação de uma maior volatilidade dos interesses e participação em pesquisas e projetos.

Como já foi mencionado no ítem 3, os arquivos XML obtidos junto a instituição são meios diretos para a extração dos campos da Tabela 5, essa processo é realizado através de uma biblioteca e os campos resultantes armazenado na estrutura com o intuito de servidor como base para a próxima etapa do SRI.

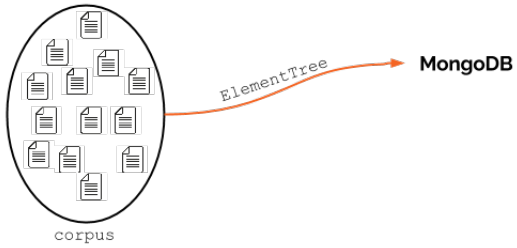

Figura 16: Processo de transformação a partir do XML

3.2 Transformação

Esta etapa sintetiza-se em um processo de limpeza e radicalização com o objetivo de tornar os dados ainda mais enxutos e significativos para o processo de indexação. Este processo é intermediário entre a coleta da informação e a cálculo de relevância de termos que será detalhado na seção seguinte.

O primeiro processo a ser aplicado nos dados, que estão armazenados no Sistema de Gestão de Banco de dados (SGBD), é a de aplicação de caixa baixa (minúsculas). Esta ação é tomada com a finalidade de diminuir a variação e melhorar a contagem e comparação das palavras, evitando que a diferenciação do tipo de caixa (baixa e alta)

da palavra implique em duas entradas diferentes. Assim um termo do tipo “CIÊNCIA” terá a mesma referência que “Ciência” e “ciência” ao final do procedimento.

O processo seguinte então trata-se da tokenização onde ocorre a conversão de texto plano em um vetor de palavras. Trata-se de uma tarefa relativamente simples, porém importante para a etapa de análise morfológica onde se aplicam técnicas de *clustering* com o objetivo de determinar limites de morfemas, tratamento de afixos e pesquisas em dicionários para encontrar a sintaxe do termo.

Os n-gramas resultantes do processos são então submetido ao processo de identificação da classe gramatical a qual pertence e também ao encontro do *word synset* da WordNet. Vejamos pela Figura 17 que o processo dá-se individualmente a cada título dos campos da Tabela 5, assim o processo resultante permite manter referência da origem da palavra.

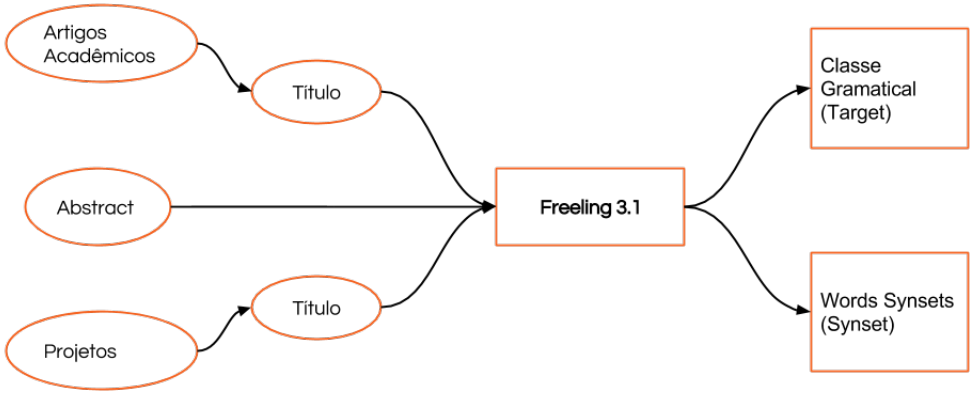

Figura 17: Os campos são submetidos ao *Freeling* para a classificação

Posteriormente removidas as palavras de parada (*stopwords*), por possuírem uma baixa capacidade de representação. Geralmente este processo se dá por uma lista de palavras pré-determinada que varia de acordo com cada autor. Neste trabalho, foi adotada a remoção das palavras pertencentes a classe fechada dado que o processo empregado anteriormente, ilustrado pela Figura 17, retorna a classificação gramatical a qual ela pertence.

As classes fechadas são constituídas por um número limitado, e normalmente reduzido de palavras, às quais a evolução da língua só muito raramente acrescenta novos termos. São palavras que geralmente estão presentes na lista de *stopwords*, porém pelo processo de remoção segundo a classe a qual pertence (*Target*) nos exime a necessidade de compilação da lista e assim remove-se as palavras destas classes em tempo de análise.

As palavras removidas pertencente as classes gramaticais descrita na tabela 17 e todas as classes (*Targets*) retornadas pelo *Freeling* para as termos está descrita no Anexo A. Com a aplicação da caixa baixa e remoção das palavras de classe fechada encerra-se o primeiro processo de transformação que diminui o tamanho do índice

Targets	
Tag	**Valor**
SP	Preposição
CS	Conjunção Subordinada
CC	Conjunção Coordenada
DA	Determinante Arigo
DD	Determinante Demostrativo
DI	Determinante Indefinido
DP	Determinante Possessivo
PD	Pronome Demostrativo
PI	Pronome Indeterminado
PP	Pronome Pessoal
PR	Pronome Relativo
PT	Pronome Interrogativo
PX	Pronome Possessivo

Tabela 6: Palavras pertencentes a estes *targets* são removidos como *StopWords*

para melhoria do desempenho do SRI.

Na sequência, a próxima etapa refere-se a aplicação do processo de radicalização (*stemming*), que permite uma melhora na revogação dos currículos (BAEZA-YATES R., 2013, p. 213). Porém esse processo não ocorre de forma sequencial nesta proposta dado a necessidade de expansão de termos, pois a radicaliação tem como objetivo a remoção dos afixos (prefixos e sufixos) para a obtendo-se o lema[3] que será usado para a indexação e ranquemento (item 3.5) e aplicação da técnica nesse momento causaria uma perda de significância importante para a expansão de termos.

Ao final do processo de transformação, contamos com um *corpus* enxuto para a indexação futura, além do mais contamos com uma série de palavras chaves e sua frequência em cada um dos documentos. Essas tuplas serão úteis para compreender quais n-gramas são importante, sobre o olhar do *corpus*, para que ocorra a expansão do termo.

3.3 Cálculo de Relevância de Termos

Quando observado o conjunto de saída da transformação, não seria interessante a expansão de todos os termos, mesmo que estes estejam já em menor quantidade proveniente do processo de transformação (item 3.2), ainda há palavras de baixa relevância que o processo de ponderação pode verificar.

Portanto a relevância levantada pelo TF-IDF, como já foi visto no item 2.1.4.4, é

[3] *lema* é a forma canônica de uma palavra. Por exemplo “Caminhei” em sua forma canônica é “caminhar”

```
{
    "lattes_id":"0990243704309282",
    "words":{
        "abstract":{
            "assistente":{
                "synset":"09608002-n",
                "target":"NCCS000"
                "len":1,
            }
        }
    }
}
```


Figura 18: “assistente” é um exemplo de n-grama resultante do processo para o campo abstract do documento.

largamente usada na RI dada a sua característica de permitir identificar termos com especificidade mínima[4]. Este mecanismo será usado para verificar quais palavras dentro do *corpus* são relevantes para que ocorra a expansão.

Este processo de corte implica até mesmo na Exaustividade Ótima[5] do documento dado que apenas selecionamos termos, dentro do documento, que o caracterize-o ao mesmo tempo que distingue dos demais.

Isso sugere que o número médio de termos de indexação por documento deve ser otimizado de modo que a probabilidade de relevância de um documento recuperado seja maximizado. Com isso partimos então para a hipótese de que termos que possuem TF-IDF com fator acima de 0.5 são candidatos promissores para que ocorra a Expansão de Termos[6].

Para isso então é calculado o fator TF-IDF para todas as palavras resultantes do processo de Transformação (item 17) e consequentemente armazenada esta informação para uso no passo seguinte que é aonde ocorre a expansão dos novos termos.

$$TF_{i,j} = \begin{cases} (1+\log_2 f_{i,j}) \times \log_2 \frac{N}{n_i} & \text{se } f_{i,j}>0 \\ 0 & \text{caso contrário} \end{cases}$$

Figura 19: Exemplo do TF-IDF resultante para o termo “Assistente”.

Como segundo produto resultante do cálculo do TF-IDF das palavras é um índice invertido contendo cada palavra presente em todos

[4]Quando o termo ocorre tem todos os documentos do *corpus* então diz-se que este termo tem especificidade mínima, logo não é útil para a recuperação dado que trará todos os documentos (BAEZA-YATES R., 2013, p 36)

[5]*Exaustibilidade* da descrição de um documento é interpretada como a abrangência que ela provê para os tópicos principais de um documento (BAEZA-YATES R., 2013)

[6]É considerado o intervalo de 0 a 1 com precisão de 6 casas decimais após a virgula

os documentos e também a referência de onde a mesma ocorre nos documentos (currículos). Bem como o total de ocorrências deles na coleção com o propósito de auxiliar na tomada de decisão na expansão.

3.4 Expansão de Termos

Nesta seção, será exposto como foi realizada a expansão dos temos usando respectivamente WordNet e DBPedia. Em seguida, é apresentado como foram representados os termos expandidos no conjunto de dados utilizado neste trabalho.

3.4.1 Expansão de Termos usando WordNet

Terminada a etapa de cálculo de relevância de termo, que serve de subsídio para este passo, então inicia-se a expansão dos termos usando a WordNet. O propósito é obter a expansão de termos de forma a manter a coesão e a exaustividade ótima, porém com novos termos relacionados. Assim optou-se pela criação de uma métrica para prover pesos para as palavras expandidas, com o objetivo de manter a coerência dos pesos em relação aos termos que as originaram.

Compreendida a necessidade de adicionar termos aos currículos, para enriquecer o vocabulário do mesmos, observou-se que as palavras expandidas deveriam então ser originadas a partir de um conjunto de palavras sinônimas[7]. Neste contexto, optou-se por buscar conexões através dos *synsets* da WordNet.

Porém os *synsets* se relacionam através de estruturas que descrevem a relação semântica entre elas. Logo contamos com relações, como por exemplo, de hiperonímia, hiponímia e meronimia (Figura 20). Com o objetivo de tornar o currículo mais próximo do vocabulário usado pela público alvo, optou-se por realizar expansões dos termos levando apenas em consideração as relações de hiperonímia, equivalência e holonímia.

Esta escolha se justifica pela própria definição que se têm de hiperonímia, já que ela é sinônimo de super-ordenado, nome que se dá ao termo cujo sentido inclui aquele (ou aqueles) de um ou de vários outros termos, chamados hipônimos. Assim temos o *synset* “Animal” que é um hiperônimo de “cão”, “gato” e “elefante” como exemplo. Esta tipo de relação se demonstra ideal, já que poderemos então expandir o termo “inteligência Artificial” a partir do termo “Redes Neurais”, dado a relação de hiperonímia que há da primeira com a segunda.

A holonímia também é uma classe de relação interessante por demonstrar a propriedade de pertencimento como parte de algo. A definição holonímia é então a carac-

[7] Diz-se de palavras que tem o mesmo significado e sentido, no entanto, são escritas com grafia distinta.

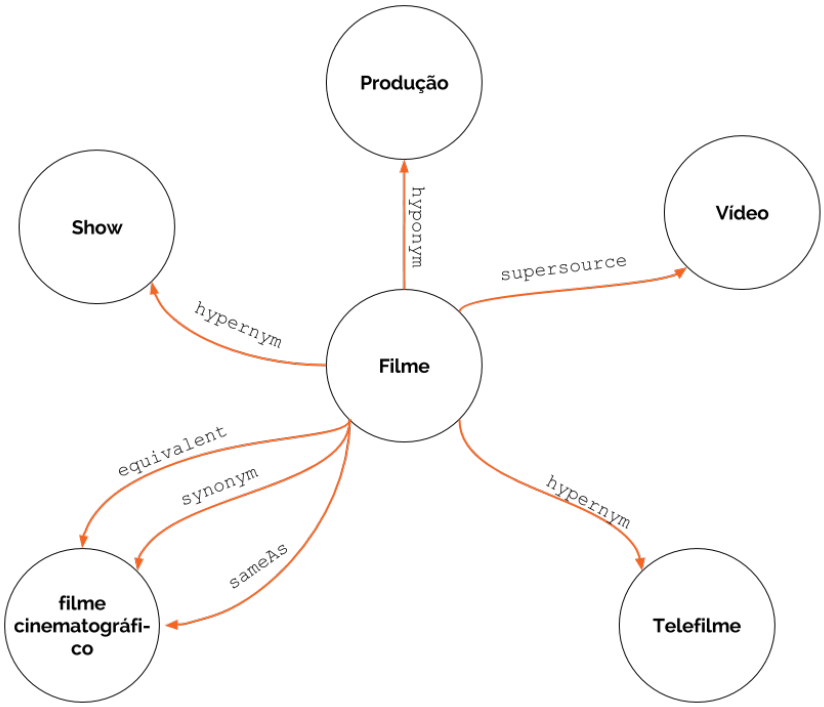

Figura 20: Relação entre o termo “*movie*” com outros termos através das relações de sinonímias, hiponímias e hiperonímias. Fonte: (PADRÓ; STANILOVSKY, 2012)

teristizada como relação de inclusão semântica entre duas unidades lexicais; uma denota um todo (holónimo) sem impor obrigatoriamente as suas prioridades semânticas à outra, considerada sua parte (merónimo). O exemplo “carro” estabelece uma relação de holonímia com “volante”, sem porém lhe impor as suas propriedades; Outros exemplos são “braço” com “corpo” e “vela” com “barco”.

Há outras relações a serem consideradas, como a de equivalência e semelhança(*same as*), por se acreditar que termos equivalentes nem sempre são usais na escrita, destaca-se a utilização desta para que ocorra uma expansão de termos caso o mesmo não esteja presentes no corpo do documento.

Semelhante relação entre palavras que poderia ser usada é a classe de hiponímia, porém esta classe tem com o princípio inverso da hiperonímia, ou seja, nos coloca em um termo mais especializado. A especialização causada pela hiponímia acarreta um inconveniente de identificar se de fato o contexto do termo é o mesmo do termo expandido. Dado o termo “Inteligência Artificial” expandir para “Redes Neurais” poderíamos causar um equívoco no documento, pois nem todos que trabalham com IA trabalham com Redes Neurais, mas a inversa é válida.

Por se tratar de uma rede, e os *synsets* estarem conectado por essas classes de relação, por exemplo, podemos então subir na árvore de hiperonímia, porém identifica-se uma objeção de perda de precisão. Termos hiperônimos possui significado mais abrangente em relação a sua origem, então a escalada na árvore tem como efeito colateral a perda da concisão.

A métrica proposta com o intuito de amortizar essa perda de concisão é o uso de uma progressão de -0.25 sobre o grau em relação ao termo original e este valor é multiplicado ao TF-IDF (peso) do termo original. Esta métrica naturalmente nos coloca um teto de 3 graus sobre o número de vezes que poderá ser expandido um termo.

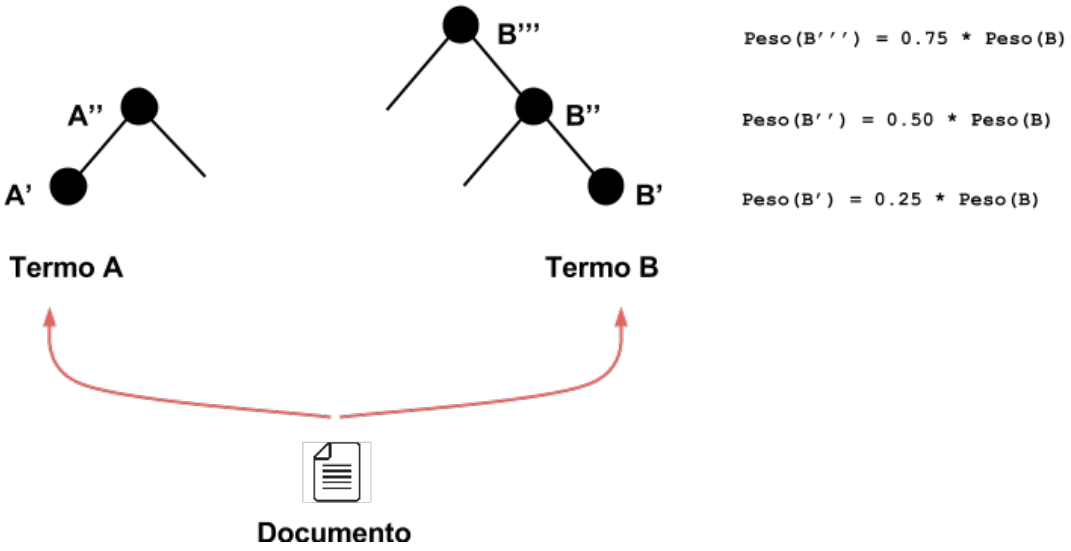

Figura 21: Expansão de termos na árvore de Hiperônimos

Com este processo de calculo, o termo é adicionado a lista de termos do documento, junto a frequência igual a 1 e adicionado também a lista global de termos e sua frequência é incrementada. Ao final da expansão de todos os termos do *corpus* que possuem coeficiente superior a 0.5 contamos então com um *corpus'* com os termos originais, seus TF-IDF e também contando com os termos expandidos.

```
{
    "lattes_id":"0990243704309282",
    "words":{
        "abstract":{
            "assistente":{
                "synset":"09608002-n",
                ...
                "tf_idf":0.006937
            }
        },
        "expandedWords":{
            "abstract":{
                "participante":{
                    "origem": "09608002-n",
                    "level": "0",
                    "tf_idf":0.005203
                }
    ...
}
```

Figura 22: A palavra "participante" é expandida a partir do termo "Assistente".

Diferente do processo aplicado a relação de Hiperonímia entre os termos originais

do corpus, acima descrito, a relação de holonímia apenas é aplicado ao termo original do corpus. O propósito é semelhante ao de amortização em até três níveis da relação de hiperonímia, pois a expansão da relação de holonímia em termos já expandidas provocam uma perda significativa da concisão.

3.4.2 Conectando conceitos da DBPedia

Como parte deste trabalho há ainda o esforço do enriquecimento do vocabulário semântico da base de conhecimento criada a partir dos RDFs gerados pela transformação Slattes sobre a ontologia VIVO-ISF. Tendo em vista que o processo de expansão de termos (item 3.4.1) já permite a conexão dos termos, expandidos e relevantes, a base de conhecimento da WordNet, há ainda a esforço em conectar os conceitos encontrados nos currículos a base de conhecimento DBpedia.

Isto visto que o processo de extração dos termos relevantes retorna importantes conceitos relacionados ao currículos lattes. Assim por exemplo, em um determinado currículo pode haver o termo “Pelotas” que para a expansão de busca trata-se de um nome próprio sem relação com outro termo através da relação semântica (veja Tabela 3. Porém o termo “Pelotas” vinculado a um determinado currículo trata-se de um conceito[8] da DBPedia (item 2.3.5) e portanto entende-se que é uma importante contribuição proporcionar que os termos também sejam relacionadas, não só a melhoria do motor de busca como também a base de conhecimento sobre a ontologia VIVO-ISF que está armazenada no AllegroGraph.

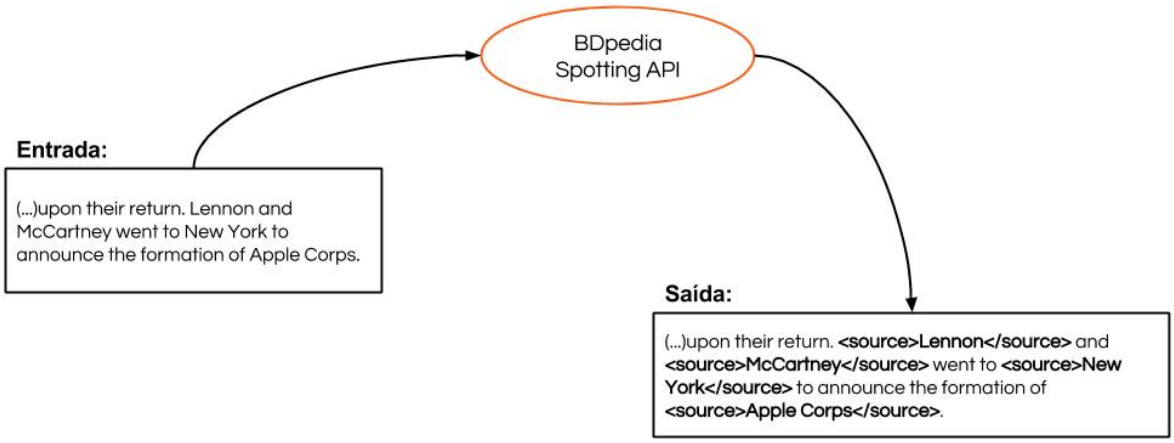

Figura 23: Ilustração da entrada e Saída da DBpedia Spotlight API

Para esta etapa optou-se pela utilização da DBPedia Spotlight API. O primeiro passo é o envio do texto para a ferramenta, os campos selecionados para esse processo foram o abstract e os títulos de artigos, projetos, orientações de cada um dos documentos do *corpus*.

[8] Pode ser também definido como uma unidade semântica

Cada um desses campos, acima mencionados, em cada um dos currículo são enviados individualmente para análise na ferramenta de forma que não fosse contaminado o seu contexto, a ferramenta foi configurada para obter uma confiança de 0.7[9] sobre o resultado do processo. Ao encontrar uma fonte para algum dos termos/conceitos, já extraídos previamente na item 3.2, a fonte é adicionada junto as demais informações que havia anteriormente do termo (Figura 24). A Figura 23 ilustra o processo pelo qual os campos são submetidos como texto e retorna-se o texto anotado com referência ao *source* a qual o termo pertence.

```
{
    "lattes_id":"0990243704309282",
    "words":{
        "abstract":{
            "fotografia":{
                "synset":"00620554-n",
                "tf_idf":0.014929
                "source": "http://pt.dbpedia.org/resource/Fotografia"
                ...
            }
        },
    ...
}
```

Figura 24: Ilustração da Inserção do *source* ao termo “fotografia” encontrado pela ferramenta

3.4.3 Representação dos novos conceitos *hasExtractedConcept*

Apesar desta etapa ser considerada opcional não impactando o processo do SRI, mas com o objetivo de prover o enriquecimento da base de conhecimento, propomos aqui uma extensão a antologia VIVO-ISF ao adicionar um novo predicado ao sujeito *Article Academic* que representa a informação relativa aos artigos publicados nos currículos. Assim sugerimos a adição da propriedade *hasExtractedConcept*[10] para que as informações obtidas na etapa acima também sejam preservadas em uma base semântica.

Esta nova propriedade é aplicada sobre o domínio *documents* e seus subclasses como o *Academic Article*, *Book*, *Journal*, *Patent* etc. O conceito “hasExtractedConcept” pode ser então inserido em forma de triplas na base de conhecimento para permitir, por exemplo, que seja possível realizar buscas semânticas sobre este novo conhecimento inserido. Assim poderá-se permitir a busca SPARQL, por exemplo, orientada para o encontro de artigos com o conceito “Banco de Dados”. Além é claro,

[9] Uma confidência de 0.7 eliminará 70% dos casos de desambiguação incorretos como pode ser visto no item 2.3.5

[10] Vinculável em http://glaucomunsberg.com/onto/vivoext

```
<owl:ObjectProperty rdf:about="&vivoext;hasExtractedConcept">
   <rdfs:domain rdf:resource="&bibo;Document"/>
</owl:ObjectProperty>
```

Figura 25: A definição da propriedade *hasExtractedConcept*

de viabilizar a ligação entre diferentes bases de dados na Web, exemplificando mais uma base ligada através de Linked Data. O *range* da propriedade *hasExtractedConcept* poderá ser populado com referências (Figura 26) a quaisquer conceitos de bases de conhecimento disponíveis na Web, tais como: WordNet[11], DBPedia[12], SUMO[13] e outras.

Figura 26: A propriedade *hasExtractedConcept* de um *Academic Article*

Para isso criou-se uma base semântica com base nos mesmos arquivos XML da seguinte forma: Primeiramente os lattes no formato XML são transformados pelo SLattes (item 2.3.1) que gera as triplas de informação em arquivos de dados RDF[14]. Estes arquivos sobre a semântica da Ontologia Vivo-ISF são então carregados para o a base de dados Allegrograph. Pelo processo de identificação do autor e seus produção o novo conhecimento é adicionado pelo conceito “hasExtractedConcept”.

3.5 Indexação e Ranqueamento

Com o término da etapa de expansão de termos parte-se para o passo de indexação, componente este que é responsável pela requisito de eficiência do SRIs. A ferramenta proposta aqui gera então uma solução de interface secundária chamada *Quantum* do qual será abordado mais amplamente no item 3.6. Aqui limita-se apenas a questão de Indexação e ranqueamento utilizado por ele, mas que finaliza a proposta de metodologia de expansão de termos bem como viabilizar as avaliações.

Com o objetivo de tratar a eficiência em retornar dos elementos mais significativos usam-se aqui a estrutura de dados conhecida como índice invertido, trata-se de uma

[11] WordNet: https://wordnet.princeton.edu
[12] DBPedia: http://dbpedia.org
[13] Suggested Upper Merged Ontology (SUMO): http://www.adampease.org/OP/
[14] A Resource Description Framework (RDF) é uma linguagem para representar informação na Internet

Boost Aplicado	
Campo	**Impulso**
dados-gerais//nome-completo	2.0
artigos-publicados trabalhos-em-eventos	2.0
participacao-em-projetos	2.0
palavra-chaves-*	1.5
//orientacoes-concluidas//	1.5
palavras-expandidas//abstract	1.0
palavras-expandidas//artigos	1.0
palavras-expandidas//projetos	1.0
palavras-expandidas//orientacoes	1.0

Tabela 7: Impulsos aplicado aos campos presentes nos documentos

solução simples, também um dos métodos mais consolidados e continua sendo largamente utilizado até hoje pela maioria dos SRIs. Também a mesma estrutura serve como solução para a função de ranqueamento necessária no SRI.

Com a etapa de indexação de currículos finalizada, é necessário definir a etapa de ranqueamento, neste SRI o modelo de ranqueamento faz o uso de um modelo híbrido entre o Modelo Booleano e o Modelo Vetorial (modelos já apresentados no item 2.1.4). No SRI é feita a composição dos dois modelos da seguinte forma: Os documentos aprovados pelo Modelo Booleando, são classificados pelo Modelo Vetorial. A abordagem é adotada dado que Modelo Vetorial possuem uma abordagem concisa e robusca para recuperação em *corpus* genéricos, bem como também, escolhido dado o escopo deste trabalho.

Para esse propósito e também para que sirva uma solução integrada e simples do projeto, a tecnologia usada pela solução *Quantum*[15], foi alinhada a tecnologia para a utilização desse modelo híbrido. Com isso propõem-se uma modificação sobre o aspecto de ranqueamento, onde o *Boost*(impulso) que será usado em cada um dos campos dos documentos, já indexado pelo SRI, seja diferente dado o grau que se considera importante para a classificação geral.

No modelo vetorial há um recurso chamado *Boost* que pode ser aplicado tanto a documentos como a campos e tem como objetivo alavancar a importância deste documento ou campo dentre os resultados ranqueados. Assim é dado uma importância através de uma atribuição de um peso. Desta formam quando o recurso é aplicado, a ponderação de cada documento ou campo é multiplicado por este *boost* que reflete em uma melhoria do seu posicionamento no *ranking* geral. Em tempo de busca (*search time*) também é possível especificar *boosts* para consultas ou para cada subconsulta realizada a partir do termo da consulta.

[15]O Quantum usa como motor de busca a tecnologia ElasticSearch (item 2.3.6)

Para este trabalho optou-se dar *boosts* em tempo de consulta. Os campos de Nome do Autor, Título das Publicações e Participação em Projetos recebem um impulso de $2,0$ pontos dado que estes são, de forma geral, nos parece os melhores campos que identificam o autor e também sua produção no *Status Quo*. Já as Palavras Chaves e Orientações Concluídas (Doutorado, Pós-Doutorado, Mestrado e Outras) recebem um impulso de 1.5 pontos e as Palavras Expandidas recebem impulso de 1.0 ponto, o que de forma virtualmente implica em uma multiplicação que não modifica o seu ranqueamento geral.

Para compreender o ranqueamento a Figura 27 apresenta a fórmula utilizada pelo ferramenta proposta para realizar o ranqueamento. Onde o $score(q,d)$ é o ponto de relevância do documento d na busca q e através da Tabela 8 podemos compreender os componentes da formula e como impactam no *score*. No Anexo 35 pode se ver a aplicação ao ElastichSearch.

$$score(q,d) = coord(q,d) \cdot queryNorm(q) \cdot \sum_{t}^{q} (tf(t\ in\ d) \cdot idf(t)^2 \cdot t.getBoost() \cdot norm(t,d))$$

Figura 27: Forma de ranqueamento utilizado

Função	**Descrição**
tf(t in d)	**TF** é a relação com a frequência, onde o "term frequency" é a quantas vezes **t** aparece no documento **d**.
idf(t)	O **IDF** aponta já o inverso, onde o peso maior é dado aos termos mais raros.
coord(q,d)	É o fator baseado em quantos termos da consulta estão presentes no documento específico.
queryNorm(q)	Este fator de normalização não afeta o ranqueamento dos documentos e trata-se de uma tentativa de normalizar uma consulta de modo a que os resultados de uma consulta tornem comparáveis aos os resultados de uma outra.
t.getBoost()	É o impulso em tempo de busca do termo **t** na consulta **q**
norm(t,d)	é a compilação de impulsos e fatores em tempo de indexação. Encontra-se nesse conjunto. Boost de documento e campo e Normalização pelo Comprimento.

Tabela 8: Descrição do ranqueamento utilziado pelo SRI

A etapa seguinte é destinada a demonstrar como é executado a consulta pelo SRI bem como elementos de interface.

3.6 Interface Quantum

Com todo o processo desenvolvido para a expansão e motor de busca, então a próxima etapa concentra-se na elaboração de uma interface gráfica que permitisse ao usuário interagir com o sistema. O modelo conceitual da interface apoia-se sobre a experiência prévias dos usuários que notamos sobre como interagem com os principais motores de busca como o Google[16], Bing[17] e Yahoo[18]. Para isso disponibilizamos na página inicial um campo textual centralizado no meio da página com o objetivo de ser o meio único para digitar a consulta (Figura 28).

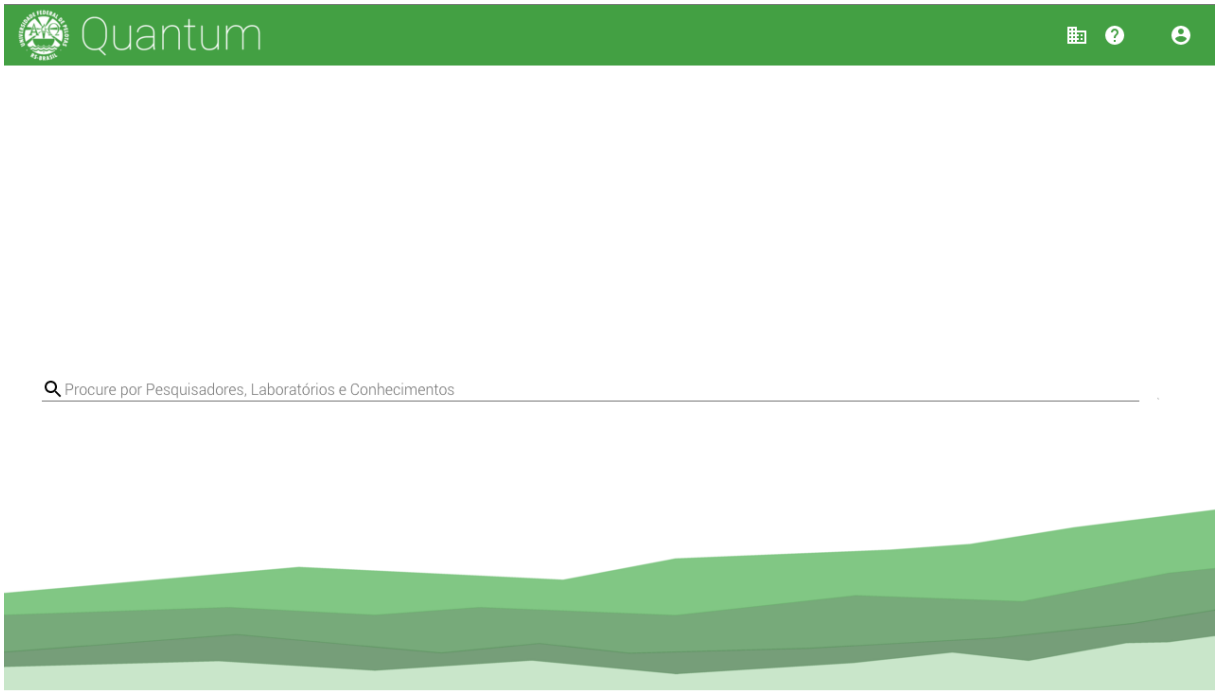

Figura 28: Página Inicial do Quantum

Já o modelo de interface adotado pela CNPq para o motor de busca possuem uma série de mecanismos para adequar a busca ao que se espera (Figura 29). Há filtros e preferências (Figura 36 em Anexos) que foram evitadas na interface aqui proposta(Quantum) para evitar que estes controles acabem por confundir o usuário quanto a qual filtro deve estar ativado ou não para obter o resultado desejado.

O *Quantum* disponibiliza o resultado de forma ordenada pela relevância do documento e paginado a cada dez documentos resultantes da consulta.

[16]https://www.google.com.br
[17]https://www.google.com.br
[18]https://br.yahoo.com/

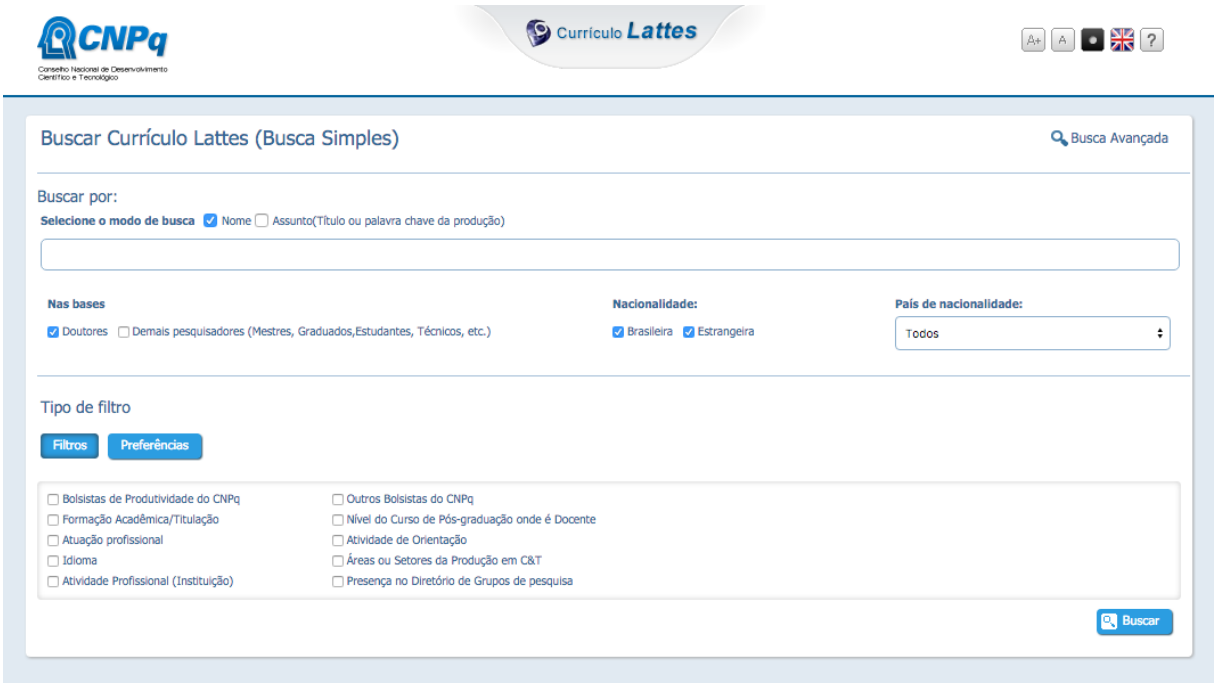

Figura 29: Página Inicial do Busca Padrão

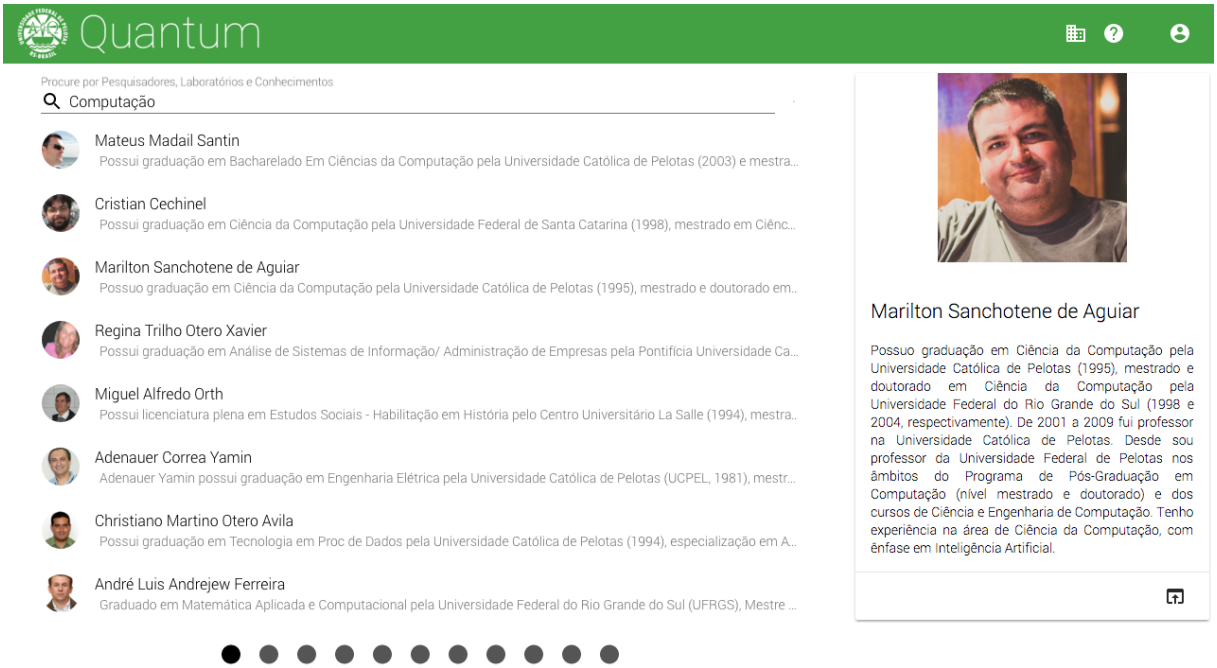

Figura 30: Página de Resultados do Quantum

4 AVALIAÇÕES E RESULTADOS

Este capítulo é dedicado a realizar avaliações com a intenção de verificar a validade deste sistema para a busca e recuperação de currículos. Com isso as avaliações estão divididas em duas sessões aonde o primeiro dedica-se a demonstrar a comparação dos resultados obtidos entre o motor de busca proposto com o oficial da CNPq. Já a segunda seção é dedicada a compreender e avaliar as expansões obtidas e como estas impactaram o ranqueamento dos currículos Lattes.

4.1 Avaliações e Comparação entre os motores de busca

Este é uma avaliação com propósito semelhante ao visto em (SOUZA MEIRELES, 2014) e tem como objetivo realizar comparações entre ambos os sistemas (Quantum e CNPq), quando aplicável, são realizadas sem que haja a modificação de filtros e preferências das ferramentas.

4.1.1 Teste 1: Busca pelo Nome

O Rótulo “SIM” é dado para o motor de busca que retorna entre os 10 primeiros ranqueados e “NÃO” para quando não encontra entre os 10 primeiros ou a busca não retornou nenhum valor. Este teste foca-se no encontro de docentes pelo nome, trata-se de uma busca simples e objetiva

Como pode ser observado ambos as ferramentas são flexíveis sobre questões

Consulta		**Quantum**	**CNPq**
1	“Ricardo Matsumura de Araujo”	**SIM**	**SIM**
2	“Ricardo Araújo Matsumura”	**SIM**	**SIM**
3	“Ricardo Araujo”	**SIM**	**NÃO**
4	“ricardo araujo ufpel”	**SIM**	**NÃO**
5	“ricard araújo”	**SIM**	**NÃO**
6	“Afrânio Kruger”	**SIM**	**NÃO**
7	“Afrânio Alberto Tavares Krüger”	**SIM**	**NÃO**

Tabela 9: Resultados para as buscas por nome

Consulta		Com Expansões	Sem Expansões
1	“carro”	**60**	**3**
2	“veículo”	**90**	**20**
3	“canino”	**51**	**51**
4	“dente”	**60**	**40**
5	“partícula”	**4**	**0**
6	“cavalo”	**37**	**20**
7	“células”	**14**	**14**
7	“celular”	**190**	**110**

Tabela 10: Número de documentos retornados no Quantum com e sem expansões

sobre acentuação, caixa baixa e também a ordem pelo qual os termos que compõem o nome da pessoa pesquisada, isto pode ser visto pelas pesquisas **(1)** e **(2)** . O resultado **(3)** retornou o valor esperado, porém não entre as 10 primeiras respostas, já para os resultados **(4)** a **(7)** o mecanismo proposto pela CNPq não foi capaz de retornar nenhum resultado para a busca.

Observa-se que a busca **4** conta com adição de um termo a mais, o nome da instituição, o que deveria promover o encontro de documentos relativo a instituição, porém o resultado foi negativo para o motor proposto pela CNPq e não retornou nenhum documento.

4.1.2 Teste 2: Conhecimento Expandido

A segunda abordagem é para testar de fato como as expansões foram capazes de impactar o número de documentos retornados. Para isso foram selecionadas algumas palavras chaves para verificar qual foi o impacto. Para isso o motor de busca foi testado com duas configurações: Com os mesmos campos e *boots* usado no Anexo C e assim contanto com as expansões, já a segunda configuração conta com os mesmos campos e *boots* porém sem as expansões.

Para isso realizamos algumas consultas com o objetivo de verificar o aumento de documentos e como essa aproximação do vocabulário ocorreu. Na Tabela 10 é possível verificar algumas buscas e nota-se por exemplo que os resultados das consultas (1) e (2) sem a expansão não conseguiriamos, por exemplo, retornar nenhum documento da área de agrícola, o que para nos que demonstra um bom resultado para o Quantum com expansão.

Outro resultado interessante a destacar é o (5) onde o resultado foi capaz de trazer 4 físicos para o motor de busca que anteriormente não era possível listá-los. Assim além de trazer relevância permitiu também que o motor de busca recuperasse algo em vez de nenhum resultado para aquela pesquisa.

Interação	Descrição
1	Realizar foco no campo de busca
2	Digitar a busca desejada (nome , conceito ou conhecimento)
3	Pressionar a tecla ENTER ou clicar no ícone de pesquisar para executar.

Tabela 11: Número de interações que o usuário precisa ter com o sistema para enviar uma requisição de busca para o Quantum

Interação	Descrição
1	Definir o modo de busca
2	Definir as bases que devem ser utilizadas na busca
3	Realizar o foco do campo de busca.
4	Digitar a consulta.
5	Pressionar a tecla ENTER ou clicar no ícone de pesquisar

Tabela 12: Número de interações que o usuário precisa ter com o sistema para enviar uma requisições de busca para a PL

4.1.3 Teste 3: Comparação das Interfaces

Assim como SOUZA MEIRELES (2014, p.76-77) comparamos a interface proposta com a motor de busca padrão da CNPq. Este teste tem como objetivo mostrar, a luz do usuário, como complexo pode ser ou não uma busca em ambas implementações. Começaremos pelo Quantum, a Tabela 11 demostra que são necessário apenas 3 passos para a realização do processo.

A Tabela 12 representa o número de interações que é preciso ser feito para realizar uma busca na ferramenta. Como pode ser visto na Figura 36, já mencionado antes e também por SOUZA MEIRELES (2014) o grande número de filtros que são manipulados pelo usuário na iteração 1 e 2 fazem com que o usuário precise ter domínio do que busca, por exemplo, a busca do CNPq não trás currículos não doutores por padrão, se o usuário não estiver consciente desta restrição inicial, sua consulta pode ser frustrada ao não retornar o currículo desejado.

O motor de busca[1] disponibilizado pelo CNPq permite a realização de busca em todos os currículos da sua base (veja Figura 29) tanto por nome como por assunto (título ou palavra chaves), junto com mais filtros, porém não permite a filtragem por instituição o que inviabilizou a comparação direta desta ferramenta com o Quantum. A busca por “ufpel automóvel” não gera resultado algum no motor de busca proposto pela CNPq entretanto temos um número significativo de retornos pelo Quantum.

Comparações entre o motor de busca Quantum e da CNPq não são feitas, por exemplo, apenas adicionando a palavra “ufpel” ou “Universidade Federal de Pelotas”, pois a base da CNPq retornaria pessoas que se formaram e/ou já tiveram vínculos com a instituição, sendo assim injusta a comparação por este e meios semelhantes.

[1] http://buscatextual.cnpq.br/buscatextual

4.2 Resultados da expansão de termos no motor de busca

O segundo conjunto de testes foram idealizados com o objetivo de coletar o quanto as expansões realizadas impactariam os resultados das buscas. Para isso então foram coletadas uma série de informações de forma anônima aos os usuários, mas que identifica-se como eles realizavam as pesquisas e interagiram com o resultado. As informações coletas foram:

As palavras chaves nas consultas, o dia em que ocorreu a busca e o navegador de origem do usuário;

O identificador do currículo lattes que foi clicado;

A posição em que o resultado clicado estava na lista geral de resultados retornadas pelo motor de busca;

Se alguma posição da lista visível ao usuário tinha a expansão da palavra buscada;

Se o resultado em que o usuário clicou havia ele sido retornado por causa do termo expandido naquele documento.

Também foram coletadas informações de *feedback* através de um formulário que esporadicamente aparecia para os usuários. Neste formulário foram coletadas informações sobre a experiência que o usuário obteve. Foram feitas as seguintes perguntas:

”O que você estava buscando no Quantum”com múltiplas escolha;

”Qual é o grau de satisfação com a(s) busca(s) realizada(s)”em uma escala de 0 a 5;

”Você identifica-se como”com múltiplas escolhas;

”Como você chegou até o Quantum”com múltiplas escolhas;

”Ajude a melhorar a Ferramenta descrevendo sua experiência”múltipla escolha;

Ainda para auxiliar a obter mais informações sobre os usuários o sistema Quantum contou com a integração ao *Google Analytics*[2] que é uma ferramenta que permite uma análise sobre o tráfego realizado em sistemas web. Por este mecanismo foram coletadas as seguintes informações

Tempo de duração do usuário na página;

Número de Páginas Visitas

Tipo de dispositivo

[2]https://analytics.google.com/analytics/

4.2.1 Dados Coletados

Com o objetivo de mensurar o impacto das expansões o sistema foi liberado para acesso ao público no dia 07 de Novembro de 2015 e foram realizadas as coletas até o dia 25 de Novembro de 2015. Contabilizando assim 18 dias que foram coletadas as informações que dão base a aos resultados abaixo descritos. Foram realizadas um total de 1,063 consultas no sistema pelos usuários, sendo que deste montante, 604 resultados foram clicados para visualizar mais informações sobre o currículo. Estes 604 resultados estão distribuídos sobre 280 lattes dos 1995 currículos cadastrados no sistema.

Do formulário foram contabilizadas 51 participações, destas participações 24 foram de discentes, 11 de professores internos e externos à UFPel e 9 participação de pessoas em empresas privadas e 5 participações em ouras categorias. O grau de satisfação de 1 de 5 obtivemos uma média de 3,66. Vejamos que em 71,2% das buscas contavam com a necessidade de encontrar Conhecimento e Competências e que do montante 28,8% procuravam uma pessoa específica.

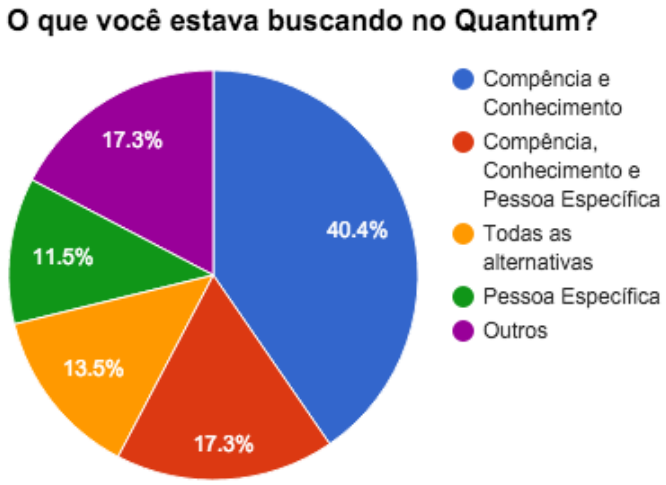

Figura 31: Resultado: Busca por tipo de conteúdo

Dos dados coletados pelo *Google Analytics* destacam-se as seguintes informações: Foram realizadas 2.671 visualizações de páginas do Quantum, onde 72,09% das visitas foram novas e a interação com o sistema durou em média 8 minutos e 13 segundos.

4.2.2 Resultado por tipo de documento

Com o cuidado de preservar a informação de cada clique para que pode-se ser feita a catalogação e qualificação delas culminou nos resultados abaixo descrito. Primeiramente analisamos a questão de distribuição dos cliques por posição, ou seja,

	Entre as 10 primeiras posições	A partir da 10ª posição
Em Números	562	42
Em Porcentagem	93%	7%

Tabela 13: Resultado dos cliques por posição

o quanto bem posicionado estava o resultado esperado para o usuário. Assim entre as 10 melhores posições encontrou-se 562 cliques, ou seja, 93% das buscas foram realizadas e o esperado estava na primeira página, dado que o Quantum listava os 10 primeiros resultados e paginava os demais onde apenas 7% precisou ir buscar o resultado em outras páginas que não fosse a primeira.

Quando observamos a ocorrência do clique por tipo observamos que dos 604 cliques que ocorreram, 202 das pesquisas realizadas contaram com a expansão de termo entre os documentos listados porém 402 delas não contaram com nenhuma expansão. E que também 131 cliques, dos 202 cliques, que ocorreram foram em documentos que tinha o termo expandido (Veja na Figura 32).

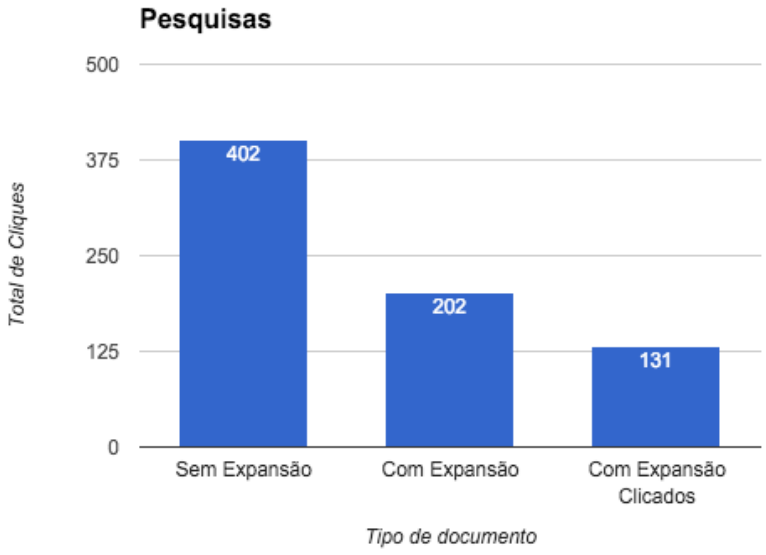

Figura 32: Número de cliques por tipo

Assim 23.1% das buscas que foram realizadas no sistema tiveram o resultado retornado e clicados por consequência direta das expansões realizadas pelo processo sugerido neste trabalho, deste montante de 202 pesquisas com termos expandidos, 131 delas ou 64.9% foram clicadas pelo usuário.

Na Figura 33 podemos observar a distribuição completa de todos os cliques que foram realizados nas 604 buscas clicadas do sistema. Como pode ser visto na Tabela 33 a maioria dos resultados encontram-se entre as 10 primeiros posição. Quando levado em consideração apenas as três primeiras posição contamos com 445 cliques o que corresponde a 57.6% dos cliques.

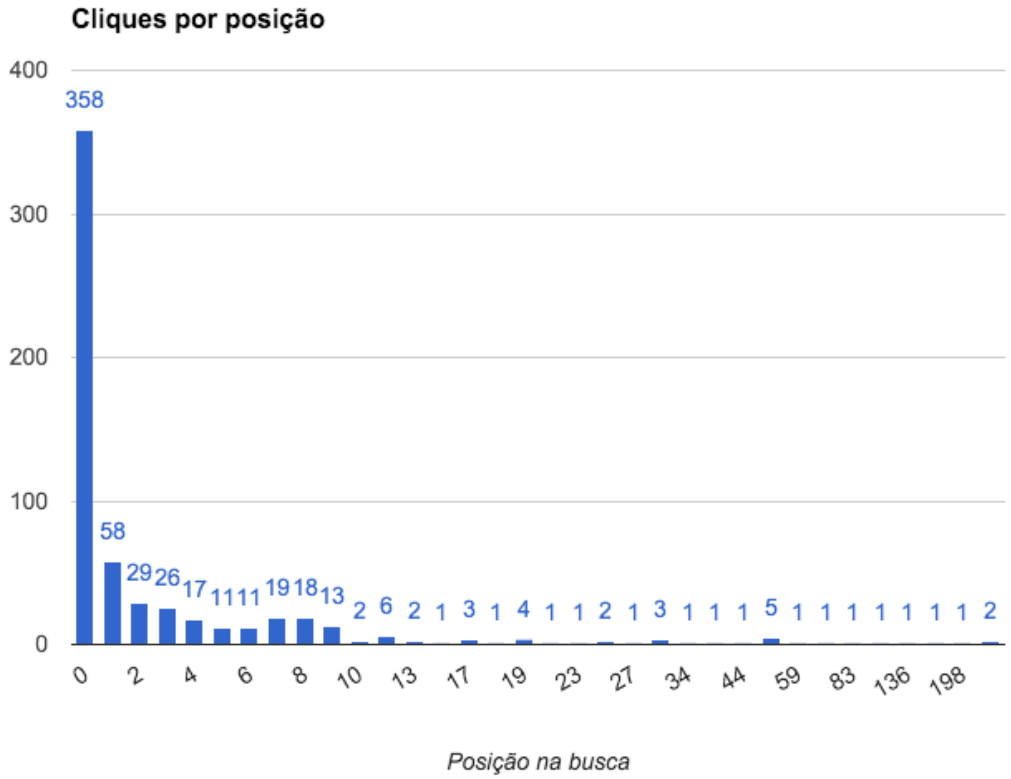

Figura 33: Distribuição dos cliques por posição

A Figura 34 demonstra a distribuição por posição dos 131 cliques que foram realizados unicamente por causa da expansão de busca. As posições que não receberam nenhum clique foram omitidos no gráfico.

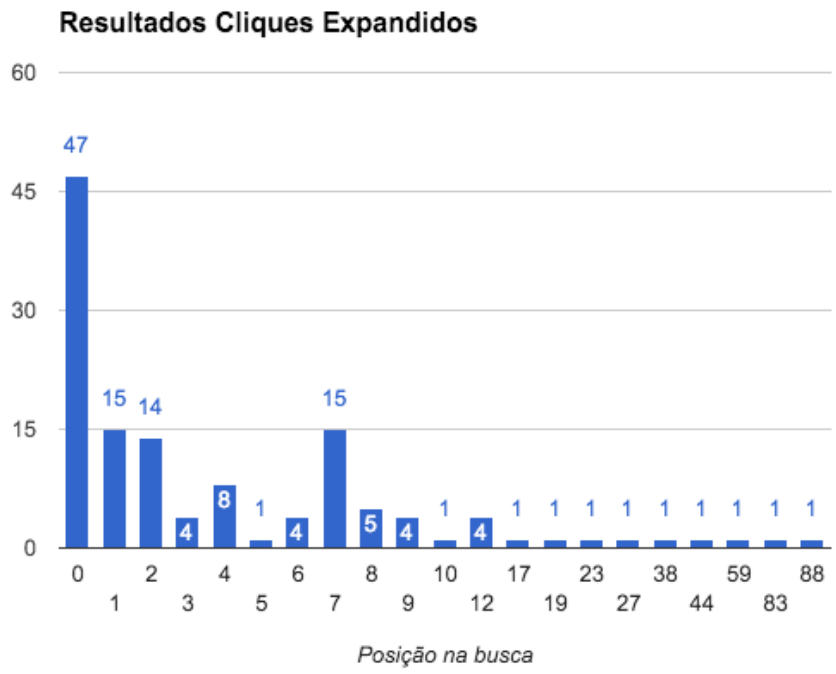

Figura 34: Distribuição dos cliques por posição

5 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo promover a expansão de termos realizados pela base de conhecimento lexical WordNet para aproximar o vocábulo dos docentes ao da comunidade. E também prover um meio pelo qual o público poderia encontrar competências, conhecimentos e pessoas através dos currículos da plataforma lattes.

Os resultados obtidos no motor de busca, com o sem expansões de termos, mostraram que houve um ganho significativo na aproximação do vocabulário entre o utilizado pela comunidade e pelas publicações indexadas. Já observando que 23.1% das consultas (202) realizadas contaram com uma expansão e que estas consultas 131 delas foram clicadas pelo usuário demonstra, que para esse conjunto de texto, houve uma relevância significativa para o motor de busca. Porém os resultados sobre os cliques nos parece ainda carecer de maiores avaliações para compreender se o clique realizado foi efetuado em um resultado realmente desejado ou apenas clicou-se por estar em uma das primeiras posições do ranqueamento.

Infelizmente a ferramenta de busca do Lattes não permitiu uma comparação direta entre o buscador da PL com o Quantum, dado que a primeira não oferece mais a funcionalidade que permitia filtrar por instituição como comparou SOUZA MEIRELES (2014) em seu trabalho. A falta de funcionalidades e ferramentas disponíveis pela PL para a extração do conhecimento desde importante repositório de conhecimento técnico-cientifico são então a chave para o incentivo de novos trabalhos que permitam, como este, solucionar a complexa tarefa de extração de competências e informações desta plataforma.

Esta monografia propõem uma metodologia para expansão de termos que foram posteriormente indexados em um motor de busca. Porém acredita-se que dado esse trabalho há ainda espaço para diversos trabalhos futuros tanto na área do SRI como em relação a Semântica. Com este propósito na sequência são apresentados algumas propostas que poderão ser realizados para melhorar a ferramenta.

- A conexão que este trabalho propõem entre termos dos currículos com o *sources* da base DBpedia (item 3.4.2) nos permite que a abordagem de expansão de busca, aqui proposta pela rede de sinonima, poderia ser também averiguada sobre a

luz da expansão de termos com base na rede do *source* com outros *sources* da DBpedia. Tendo em vista que a base da DBpedia é uma rica rede de informação, os termos com *sources* identificados poderiam adicionar novos termos pela sua sub-rede.

- Há espaço para o refinação dos métodos de avaliação, assim por exemplo, realizar testes com amostragens temporais maiores com grupos de usuário com e sem expansão sobre o(s) mesmo(s) termo(s) de busca. Isto ajudaria a compreender como a expansão de busca está impactando a distância entre a resposta esperada do usuário com o ranqueamento retornado pelo SRI.

- A utilização do SLattes para conversão dos lattes em uma base semântica RDF e posteriormente armazenados no AllegroGraph gerou uma importante repositório de conhecimento semântico sobre os trabalhos da instituição, logo, abre-se precedentes para outros trabalhos que visam, por exemplo, a extração de conhecimento desta base de conhecimento.

- Dada a base de conhecimento gerada pelos arquivos RDF sobre a ontologia VIVO-ISF há a possibilidade de explorar inferências lógicas sobre os termos expandidos pelo processo exposto aqui neste trabalho. Visto que usou-se a ontologia WordNet para representar os termos expandidos e também contar com conceitos ligados a DBpedia.

- Ainda sobre a perspetiva de melhoraria do ranqueamento, pensa-se na utilização do fator H para melhorar o ranqueamento. Este é um modelo fortemente indicado, dado o Fator H é um cálculo para compreender quantas citações têm o artigo de um determinado autor. A incorporação deste fator poderá alavancar os currículos que possuem mais trabalhos com citações externas a base. Porém o Fator H é apenas uma das abordagens possíveis para a incorporações de informações externas para o ranqueamento, o número de *links* que há externamente ao currículo do Lattes nos parece também uma abordagem interessante.

- Compreendemos ao final do trabalho que a realimentação de relevância nos parece uma abordagem interessante a se melhorar o ranqueamento do documentos, assim quando um documento é clicado ele ganha uma maior relevância nas próximas consultas, tendo invista que ao realizar novamente a mesma consulta o documento já terá melhorado sua posição.

- Uma abordagem interessante a ser empregada a partir deste momento é a realimentação por meio de cliques (BAEZA-YATES R., 2013, p.165) com o objetivo de aumentar o ranqueamento dos documentos mais populares do *corpus*.

- A base de conhecimento aqui utilizada foi construida sobre os 12 campos extraído dos arquivos lattes e armazenados no MongoDB. De forma lateral foi também construído uma base de conhecimento a partir dos RDFs originados da transformação dos XMLs. O processo aqui descrito usou-se do conhecimento do banco de dados e apenas inseriu-se novos atributos semânticos a base de conhecimento, um trabalho futuro seria trabalhar o processo inteiramente sobre a base de conhecimento aqui construido.

REFERÊNCIAS

AGIRRE, E.; SOROA, A. Personalizing pagerank for word sense disambiguation. In: CONFERENCE OF THE EUROPEAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 12., 2009. **Proceedings...** [S.l.: s.n.], 2009. p.33–41.

ALMEIDA, M. B.; BAX, M. P. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. **Ciência da Informação, Brasília**, [S.l.], v.32, n.3, p.7–20, 2003.

BAEZA-YATES, R.; FRAKES, W. B. **Information retrieval**: data structures & algorithms. [S.l.]: Prentice Hall, 1992.

BAEZA-YATES R., R.-N. B. **Recuperação de informação**: Conceitos e tecnologias das máquinas de busca. 2.ed. [S.l.]: Porto Alegre: Bookman, 2013.

BORST, W. N. **Construction of engineering ontologies for knowledge sharing and reuse**. [S.l.]: Universiteit Twente, 1997.

BÜTTCHER, S.; CLARKE, C. L.; CORMACK, G. V. **Information retrieval**: Implementing and evaluating search engines. [S.l.]: Mit Press, 2010.

CARRERAS, X.; CHAO, I.; PADRÓ, L.; PADRÓ, M. FreeLing: An Open-Source Suite of Language Analyzers. In: LREC, 2004. **Anais...** [S.l.: s.n.], 2004.

CROFT, W. B.; METZLER, D.; STROHMAN, T. **Search engines**: Information retrieval in practice. [S.l.]: Addison-Wesley Reading, 2010.

DIAS, T. D.; SANTOS, N. Web Semântica: Conceitos Básicos e Tecnologias Associadas. **Cadernos do IME-Série Informática**, [S.l.], v.14, p.80–92, 2013.

FELLBAUM, C. **WordNet**: An Electronic Lexical Database. 1.ed. [S.l.]: Bradford Books, 2009.

FENSEL, D.; FACCA, F. M.; SIMPERL, E.; TOMA, I. **Semantic web services**. [S.l.]: Springer Science and Business Media, 2011.

GARCIA, M.; GAMALLO, P. Análise Morfossintáctica para Português Europeu e Galego: Problemas, Soluções e Avaliação. **Linguantica**, [S.l.], v.2, n.2, p.59–67, 2010.

KUC, R.; ROGOZINSKI, M. **ElasticSearch server**. [S.l.]: Packt Publishing Ltd, 2013.

LEACOCK, C.; CHODOROW, M. Combining Local Context and WordNet Similarity for Word Sense Indentification. In: FELLBAUM, C. (Ed.). **WordNet**: An eletronic Lexical Database. Oxford: Library of Congress Cataloging-in-Publication Data, 1998.

MARRAFA, P.; AMARO, R.; CHAVES, R. P.; LOUROSA, S.; MARTINS, C.; MENDES, S. WordNet. PT - Uma rede léxico-conceptual do Português on-line. **XXI Encontro da Associação Portuguesa de Linguística, Porto, Portugal**, [S.l.], 2005.

MARTIN PORTER, R. B. **snowball**.

MENDES, P. N.; JAKOB, M.; GARCÍA-SILVA, A.; BIZER, C. DBpedia spotlight: shedding light on the web of documents. In: INTERNATIONAL CONFERENCE ON SEMANTIC SYSTEMS, 7., 2011. **Proceedings...** [S.l.: s.n.], 2011. p.1–8.

ORENGO V. M.; HUYCK, C. A Stemming Algorithm for Portuguese Language. Symposium On String Processing And Information Retrieval,. In: EIGHTH INTERNATIONAL SYMPOSIUM ON STRING PROCESSING AND INFORMATION RETRIEVAL, 2001, 2011. **Proceedings...** [S.l.: s.n.], 2011. p.1192–1199.

PADRÓ, L.; STANILOVSKY, E. FreeLing 3.0: Towards Wider Multilinguality. In: LANGUAGE RESOURCES AND EVALUATION CONFERENCE (LREC 2012), 2012, Istanbul, Turkey. **Proceedings...** [S.l.: s.n.], 2012.

PAIVA, V. de; RADEMAKER, A.; MELO, G. de. OpenWordNet-PT: An Open Brazilian WordNet for Reasoning. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS, 24., 2012. **Proceedings...** [S.l.: s.n.], 2012. See at http://www.coling2012-iitb.org (Demonstration Paper). Published also as Techreport http://hdl.handle.net/10438/10274.

RADEMAKER, A.; HAEUSLER, E. H. Semantic Lattes and VIVO Project. In: VIVO 2013, 2013. **Proceedings...** [S.l.: s.n.], 2013.

SHEKARPOUR, S.; HOFFNER, K.; LEHMANN, J.; AUER, S. Keyword query expansion on linked data using linguistic and semantic features. In: SEMANTIC COMPUTING (ICSC), 2013 IEEE SEVENTH INTERNATIONAL CONFERENCE ON, 2013. **Anais...** [S.l.: s.n.], 2013. p.191–197.

SOUZA MEIRELES, G. de. **Currículo Lattes**: Uma abordagem de busca explorando a recuperação de informação. 2014. Dissertação (Mestrado em Ciência da Computação) — CDTec/Universidade Federal de Pelotas.

VOORHEESS, E. M. Using WordNet for Text Retrieval. In: FAGERBERG, J.; MOWERY, D.; NELSON, R. (Ed.). **WordNet**: An eletronic Lexical Database. Oxford: Library of Congress Cataloging-in-Publication Data, 1998.

ANEXO A LISTA DE *TARGETS*

Lista da construção do target do *Freeling* e maiores detalhes estão disponíveis em (GARCIA; GAMALLO, 2010, p. 8).

Targets	
Tag	**Valor**
AO	Adjetivo Ordinal
AQ	Adjetivo Qualitativo
CS	Conjunção Subordinada
CC	Conjunção Coordenada
DA	Determinante Arigo
DD	Determinante Demostrativo
DI	Determinante Indefinido
DP	Determinante Possessivo
I	Interjeição
NC	Nome Comum
NP	Nome Próprio
PD	Pronome Demostrativo
PI	Pronome Indeterminado
PP	Pronome Pessoal
PR	Pronome Relativo
PT	Pronome Interrogativo
PX	Pronome Possessivo
RG	Advérbio Geral
RN	Advérbio Negativo
SP	Preposição
VG	Verbo: Gerúndio
VI	Verbo: Modo Indicativo
VM	Verbo: Modo Imperativo
VN	Verbo: Infinitivo
VP	Verbo: Particípio
VS	Verbo: Memo Conjuntivo
Z	Numeral

Tabela 14: Target e Classes Gramaticais. Fonte: (PADRÓ; STANILOVSKY, 2012)

ANEXO B INFRAESTRUTURA

Ao que tange a infraestrutura de *hardware* para executar a aplicação foi utilizada uma instância em nuvem provida pela DigitalOcean[1], porém o processo pode ser replicado em uma máquina com uma configuração mínima como é possível ver na Tabela 15.

Servidor	
Recurso	**Quantidade Mínimo**
Memória RAM	2GB
Espaço em Disco	20GB
Sistema Operacional	Linux/Fedora

Tabela 15: Configuração Mínima exigida par a execução do processo

A Tabela 16 tem como objetivo listar as bibliotecas e pacotes que foram necessários para a execução e replicação o processo aqui descrito.

Software	
Nome	**Versão**
Freeling	3.1
Python	2.7
MongoDB	3.0.5
ElasticSearch	1.7.2
AllegroGraph	5.1.1
WordNet	3.0
Ruby On Rails	2.1
MySQL	5.0

Tabela 16: Softwares necessários e suas versões

[1] DigitalOcean é uma empresa que provê servidores escaláveis para infraestrutura web e pode ser acessado pelo link http://www.digitalocean.com

ANEXO C *MARTCH* DE CONSULTA

```
{ match: { "expandedWords.abstract" => { query: $query,boost:1}} },
{ match: { "expandedWords.orientations" => { query: $query,boost:1}} },
{ match: { "expandedWords.articles" => { query: $query,boost:1}} },
{ match: { "expandedWords.projects" => { query: $query,boost:1}} },
{ match: { "key_words.*" => { query: $query,boost:1.5}} },
{ match: { name: { query: $query,boost:2}} },
{ match: { abstract: { query: $query,boost:1.5}} },
{ match: { "orientations.*" => { query: $query,boost:1.5}} },
{ match: { "projects.*" => { query: $query,boost:2}} },
{ match: { "articles.*" => { query: $query,boost:2}} }
```

Figura 35: Configuração das consultas vetoriais usadas pelo modelo booleano

ANEXO D FILTROS DA FERRAMENTA DO LATTES

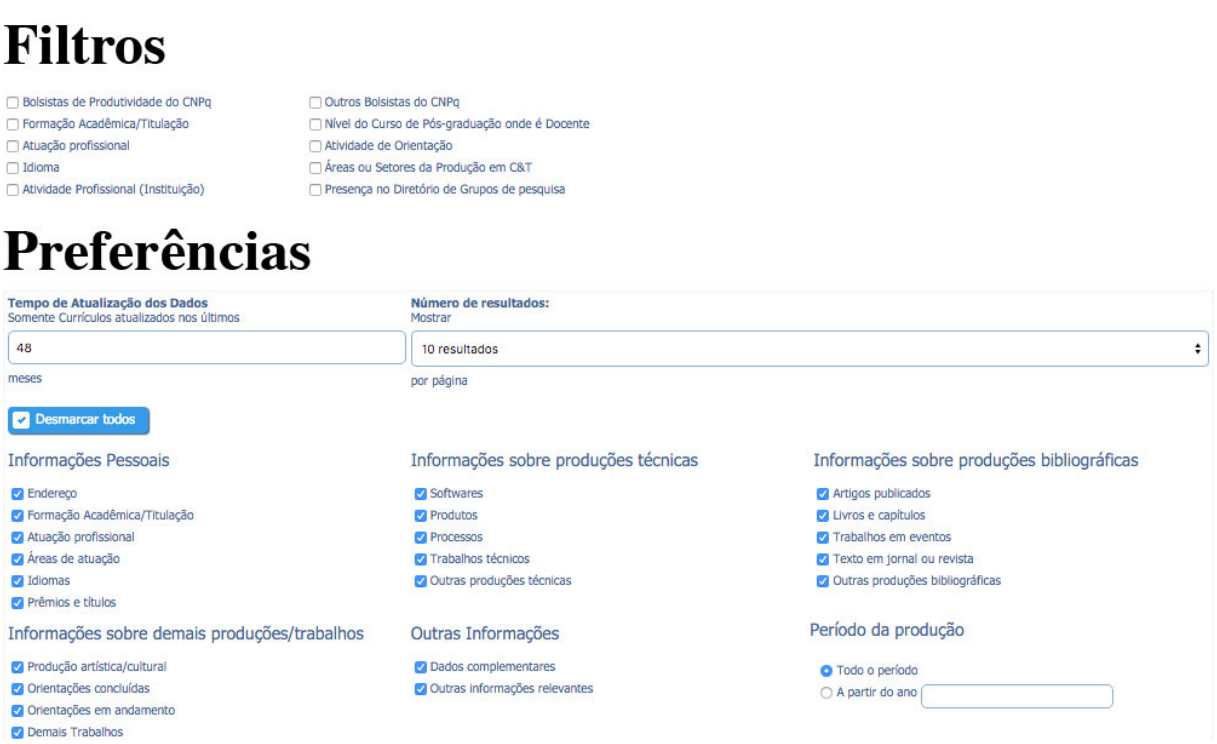

Figura 36: A ferramenta disponível pela CNPq disponibiliza 8 grandes agrupadores de preferências e 10 filtros para uma única busca

Currículos Lattes: Expansão Automática de Termos baseada em Ontologia – Glauco Roberto Munsberg dos Santos

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Curso de Bacharelado em Ciência da Computação

Trabalho de Conclusão de Curso

Currículos Lattes: Expansão Automática de Termos baseada em Ontologia

GLAUCO ROBERTO MUNSBERG DOS SANTOS

Pelotas, 2015