Busc@NIMA: um Sistema de Recuperação de Informações com Semântica de Meio Ambiente

André L. C. Rêgo, Daniel S. Guimarães, Marcos V. Villas, Sérgio Lifschitz

[1]Departamento de Informática – (PUC-Rio) - Rio de Janeiro - RJ

{andreregо, danielg}@aluno.puc-rio.br

{villas, sergio}@inf.puc-rio.br

Resumo. *O sistema Busc@NIMA permite identificar competências existentes, na área de meio ambiente, em laboratórios e departamentos coordenados por professores e pesquisadores da PUC-Rio. Neste trabalho descrevemos como se dá a obtenção dos dados com base nos CV Lattes e a conversão para o formato RDF usando ontologias de domínio e transformações pela linguagem XSLT. Utilizamos tanto um SGBD Relacional como dois Sistemas NoSQL. As consultas são expressas na linguagem SparQL com suporte, de forma nativa, de indexação de texto livre. O resultado de qualquer pesquisa no Busc@NIMA é uma lista de nomes de professores ou pesquisadores, suas produções científicas e atividades de ensino, contemplando apenas informações relativas com o tema de meio ambiente. A solução aqui apresentada pode ser estendida e customizada para outros temas específicos e outras universidades. Ilustramos o funcionamento do sistema em produção com um estudo de caso de buscas selecionadas.*

1. Introdução e Motivação

Este artigo descreve o Sistema Busc@NIMA, um sistema de recuperação de informações (*information retrieval*) [Baeza-Yates and Ribeiro-Neto 2011] que possibilita a descoberta de professores-pesquisadores da PUC-Rio que estejam envolvidos com o tema particular de meio ambiente, a partir de uma ou mais palavras ou termos de busca.

Os Sistemas de Recuperação de Informação são projetados para encontrar objetos digitais armazenados em grandes coleções, que atendem a uma necessidade de informação do usuário. A consulta é geralmente especificada em linguagem natural por meio de palavras-chave que representam a intenção da pesquisa e o mecanismo de pesquisa traduz essa consulta em seu modelo de linguagem para retornar uma lista de objetos digitais. Esses objetos podem ser documentos, tabelas, gráficos, vídeos ou imagens, mas em contextos mais específicos (de acordo com a finalidade e organização da coleção de objetos) podem corresponder a outros recursos como triplas ou subgrafos.

A PUC-Rio, assim como qualquer Universidade, oferece cursos de graduação e pós-graduação em diversas áreas, refletindo de certa forma as qualificações e formações de seus professores e pesquisadores. Alguns anos atrás o NIMA, Núcleo Interdisciplinar de Meio Ambiente PUC-Rio, recebeu uma demanda particular: criar um *website* que teria as informações sobre as pessoas da universidade envolvidas com o tema de meio ambiente, seja em projetos de pesquisa, seja em disciplinas oferecidas regularmente ou eletivas. O objetivo seria poder atender pedidos da mídia tradicional quanto aos especialistas que poderiam comentar assuntos relacionados em matérias de jornais ou televisão,

e também uma eventual prestação de serviços na área por parte de professores, pesquisadores, funcionários e alunos.

Nesse contexto, o sistema Busc@NIMA foi desenvolvido, com o entendimento desde o início de que não teríamos uma solução satisfatória com um conjuntos de páginas HTML estáticas. O projeto do sistema começou com um conjunto de termos e palavras-chave fornecido pelos integrantes do NIMA mas que, rapidamente, se mostrou limitado em termos de abrangência e utilidade para a montagem do *website* planejado. Decidimos, assim, considerar como fontes de dados as disciplinas oferecidas na PUC-Rio e as informações contidas nos curriculos Lattes[1] dos membros da comunidade PUC-Rio. As disciplinas são disponibilizadas publicamente pela PUC-Rio e também por API dos serviços computacionais acadêmicos da Universidade. Para obter os dados do Lattes apenas dos integrantes da PUC-Rio contamos com o apoio da Coordenação Central de Planejamento Acadêmico (CCPA) que tem o acesso direto e restrito (sem captcha) aos currículos Lattes, e que fornecem mensalmente um *dataset* selecionado apenas dos professores e pesquisadores da PUC-Rio.

Entretanto, o sistema Busc@NIMA trouxe um grande desafio: *como reduzir o escopo da base de dados formada pelas disciplinas, projetos de pesquisa e produções acadêmicas sem restrição de temática de forma a contemplar somente o escopo de meio ambiente?*. Neste trabalho, discutimos algumas alternativas e apresentamos a solução implementada e que está atualmente em produção[2].

2. Projeto e Especificação do Busc@NIMA

O sistema foi construído em Python em conjunto com o microframework Flask[3], que possibilita a prototipação de aplicações web. O artigo [Salgueiro et al. 2021] traz maiores detalhes do projeto e arquitetura do Busc@NIMA, que utiliza a mesma estrutura de desenvolvimento de um sistema de busca genérico, o Quem@PUC[4].

Considerando a finalidade do sistema, dois requisitos não-funcionais foram identificados na fase inicial de desenvolvimento que nortearam o *design* da arquitetura do sistema: (1) suportar um esquema de dados flexível e (2) permitir a carga de dados em diferentes formatos, como XML, RDF e CSV. Um modelo de dados sem esquema rígido permite que os dados tenham variedade na estrutura. Assim, buscamos um modelo de dados não relacional. O SGBD NoSQL AllegroGraph foi selecionado para o armazenamento centralizado de dados originados de outros sistemas. Trata-se de um *triplestore* que permite a manipulação de triplas RDF e a visualização, em estruturas de grafo. O AllegroGraph permite o maior número de triplas (cinco milhões) por repositório em comparação com outras opções disponíveis em sua versão gratuita, oferece suporte à linguagem SPARQL (linguagem padrão para /textitlinked data) e, também, possui suporte nativo à indexação de dados textuais (*Freetext Indexing*), permitindo mapear rapidamente palavras e frases para as triplas do banco de dados.

Outros dois requisitos foram identificados durante o desenvolvimento: (1) a necessidade de armazenamento dos resultados de uma busca como um cache temporário e (2)

[1] `https://lattes.cnpq.br/`
[2] `https://buscanima.biobd.inf.puc-rio.br/`
[3] `https://flask.palletsprojects.com/en/2.1.x/`
[4] `https://quempuc.biobd.inf.puc-rio.br/`

o suporte às modificações sequenciais dos resultados (das buscas) em um curto período de tempo. Mesmo que pareça desnecessária, a adoção de um SGBD para esse armazenamento de cache permite prover escalabilidade e maior eficiência. Particularmente, optamos por usar o Sistema NoSQL de tipo chave-valor Redis[5]. Redis é um sistema NoSQL para armazenamento em memória de estruturas de dados simples. Esse SGBD permite configurar nativamente tempos de expiração para as estruturas de dados, funcionalidades excelentes para formar uma estrutura de *caching* por exemplo.

Para armazenar dados administrativos da ferramenta, como log ou tabelas de contas dos usuários especiais do sistema, foi utilizado o SGBD relacional PostgreSQL.

2.1. Conversão das Diferentes Fontes de Dados

A plataforma Lattes armazena e disponibiliza a produção dos pesquisadores no país. Para extração destes dados desenvolvemos um script XSLT[6] que transforma cada arquivo XML gerado da plataforma Lattes em RDF, usando ontologias selecionadas (e.g FOAF), e um outro script específico de carga dos arquivos RDF para o sistema NoSQL AllegroGraph.

Além dos dados da plataforma Lattes, outras três fontes de dados são convertidas e inseridas em formato de triplas no banco de dados: A primeira é uma planilha contendo informações administrativas dos professores da PUC, convertida utilizando um script Python que utiliza a API[7] disponibilizada pelo AllegroGraph para inserir estes dados no banco. A segunda é coletada da ferramenta de consulta online de disciplinas da PUC-Rio[8] utilizando uma biblioteca Python especializada[9] e utiliza a mesma API; A terceira e última fonte de dados advém de informações não contidas no CV Lattes que são inseridas pelos próprios professores e pesquisadores utilizando a área de usuário do Busc@NIMA. Permite-se inserir palavras-chave e informações associadas aos laboratórios, como, por exemplo, detalhes dos equipamentos e suas especificações. Para inserir estes dados, desenvolvemos outros scripts Python com a mesma API do AllegroGraph.

2.2. Linguagem de Consulta

As consultas de correspondência de subgrafos foram especificadas na linguagem SPARQL para recuperar recursos de interesse sobre pesquisadores ou professores. A correspondência de palavras-chave nas consultas foi realizada usando o operador (*magic property*) *fti:match*[10], que permite que a consulta use o índice freetext criado. Para utilizar o banco de dados chave-valor e o banco relacional, foram utilizadas abstrações Python disponibilizadas como bibliotecas para Python e Flask.

Observamos, entretanto, uma demora na execução das consultas SPARQL sobre os dados no AllegroGraph. Algumas pesquisas demoravam mais que 20 segundos para serem completamente executadas. Foram implementadas algumas ideias para reduzir este tempo de execução, como alterar os índices de texto livre utilizados, ou colocar as consultas de *threads* simultâneas para execução em paralelo, porém sem muito sucesso.

[5] https://redis.io/
[6] https://www.w3.org/TR/xslt/
[7] https://github.com/franzinc/agraph-python
[8] http://www.puc-rio.br/microhorario
[9] https://github.com/Leinadium/microhorario-dl
[10] https://franz.com/agraph/support/documentation/current/magic-properties.html

A solução de otimização passou, primeiramente, por evitar consultas simultâneas utilizando uma mesma sessão. Além disso, várias consultas semelhantes foram reduzidas em uma só consulta mais geral, reduzindo o total de consultas realizadas para obter um resultado. Além disso, metade das consultas dependiam do nome do professor/pesquisador, o que gerava uma necessidade de executar uma pequena busca repetidamente para obter o seu identificador. Foi observado que se essas consultas já iniciassem o processamento a partir desse identificador, o tempo de execução era reduzido drasticamente. Por isso, o projeto foi alterado de forma que essas consultas não fossem orientadas pelo nome do professor/pesquisador, mas sim pelo seu identificador Lattes.

Ao implementar todas estas otimizações foi observado um tempo de resposta cerca de 15 a 20 vezes menor. Por exemplo, uma pesquisa que possuía um custo de execução de 20 segundos foi reduzida a menos de 2 segundos.

2.3. Solução para a redução do escopo da base de dados

Os sistemas Quem@PUC e Busc@NIMA são similares e compartilham boa parte do código. A diferença está no banco de dados com as informações dos professores. Cada produção (artigos, disciplinas etc.) possui um valor associado entre 0 e 1, onde 0 significa nenhuma relação com o tema meio ambiente, e 1 significa total relação com o tema meio ambiente. O sistema Quem@PUC apresenta qualquer produção, independente do seu valor associado, enquanto o Busc@NIMA apresenta somente produções com um valor próximo de 1, como por exemplo, acima de 0,7. Isto permite que essa ferramenta exiba somente produções relacionadas com o tema escolhido, no caso, meio ambiente.

Estudamos diferentes maneiras de associar esse valor de semântica associada aos dados mas optamos por explorar uma rede neural que treinasse sobre um banco de produções previamente classificadas e aprendesse a classificar, a partir do título, tipo e descrição, se uma produção se relaciona com meio ambiente ou não.

Primeiramente, foi necessário classificar manualmente muitas produções dos professores para esse aprendizado supervisionado. Foi criado um script Python que permite coletar produções aleatórias do banco de dados e perguntar para um indivíduo no papel de juiz, mediante ao nome da produção, tipo, descrição e professor, se a produção podia ser relacionada, se não era relacionada ou se não era possível saber do relacionamento com meio ambiente com somente essas informações. Para evitar qualquer influência sobre as respostas de somente um juiz, três juízes classificaram as mesmas produções, e os resultados divergentes foram rejeitados. Inicialmente foram classificadas cerca de 1000 produções, porém somente 910 foram consideradas.

Em seguida, foi implementado um workflow para determinar a partir de testes quais eram as melhores opções para treinar uma rede neural com estes dados. A ideia é coletar as produções classificadas manualmente a partir de um arquivo CSV e delas passar por etapas de pré-processamento como normalização e retirada de acentuação, *tokenização* das frases, remoção de *stopwords* e, por último, a transformação dos dados resultantes através de um B.O.W (*Bag of Words*). Os dados resultantes foram utilizados como entrada em uma série de algoritmos de aprendizado de máquina, dentre eles: rede neural simples, rede neural com camadas ocultas e SVM. Dentre todos os algoritmos, os que produziram os melhores resultados foram os de rede neural com 1-3 camadas ocultas, cujos resultados variaram entre 85% e 89% de precisão.

Durante essa etapa observamos que o volume de produções classificadas manualmente como relacionadas com meio ambiente impedia as redes neurais de terem base necessária de conhecimento para classificar uma produção positivamente quanto a sua relação com a temática. Por isso, foram classificadas e adicionadas cerca de 200 produções, permitindo haver uma relação de 2.7 produções não relacionadas ao tema de meio ambiente a cada produção relacionada.

O melhor resultado obtido foi uma rede neural, mais especificamente um *perceptron* multicamadas, com três camadas ocultas (50 nós, 70 nós e 20 nós) e um nó de saída, com uma precisão de 89% utilizando 80% dos dados para treinamento e 20% para avaliação. Este modelo foi implementado e um script[11] foi criado para que esse modelo pudesse classificar todas as produções contidas no banco de dados do projeto. A classificação é armazenada no próprio banco de dados por meio de uma tripla contendo o valor obtido pela rede neural sobre aquela produção.

A efetividade dessa solução pode ser ilustrada com a busca de um mesmo termo nos dois sistemas. As figuras 1 e 2 apresentam o resultado da busca do termo "toxicidade" tanto no Quem@PUC quanto no Busc@NIMA, respectivamente.

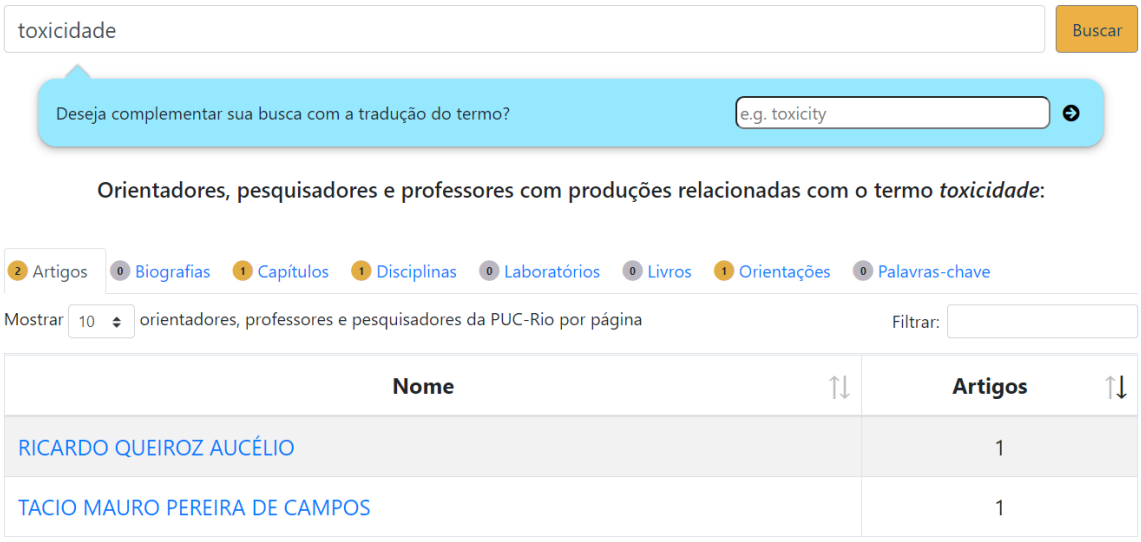

Figura 1. Busca do termo "toxicidade" no Quem@PUC

É possível observar, por exemplo, que no Quem@PUC foram encontrados mais artigos relacionados a esse termo do que no Busc@NIMA, onde o subconjunto pesquisado é restrito ao tema meio ambiente, devido à solução adotada para a redução do escopo da base de dados.

Dentre os autores de artigos relacionados ao termo "toxicidade" no Quem@PUC está professor Ricardo, que não se encontra na lista de autores de artigos no Busc@NIMA para esse mesmo termo. De fato, o artigo encontrado no Quem@PUC desse professor tem o título *"Complexos de Mn(II) e Co(II) de nitro-tiossemicarbazonas: estudo das propriedades espectroscópicas e toxicidade frente a Artemia sp"* que não trata do tema meio ambiente.

[11] `https://github.com/Leinadium/nima-predict`

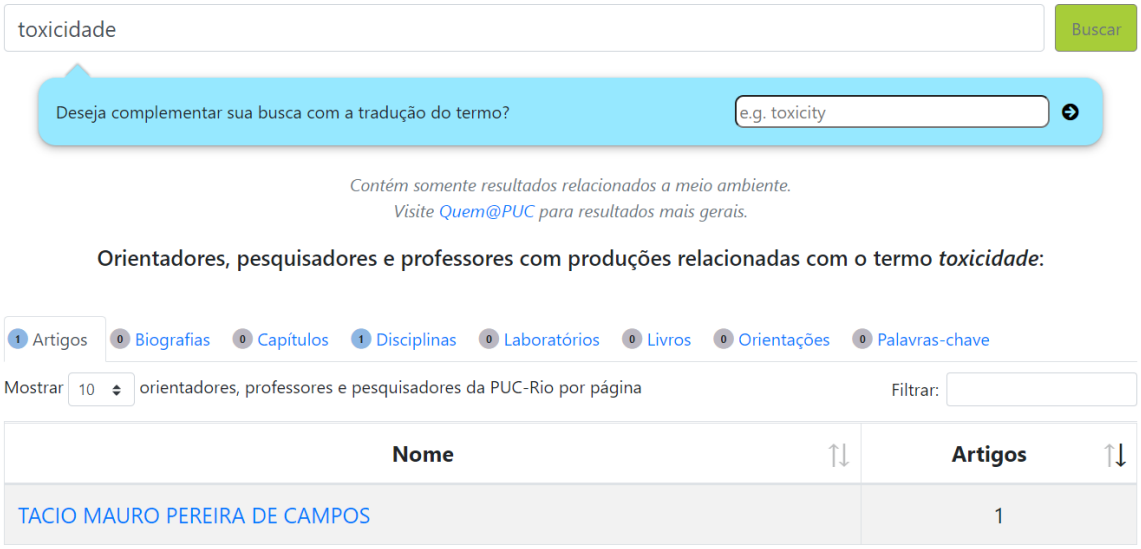

Figura 2. Busca do termo "toxicidade" no Busc@NIMA

Por outro lado, o professor Tacio aparece nos dois sistemas pois o artigo de título *"Influência da salinidade na toxicidade de sedimentos dragados da Lagoa Rodrigues de Freitas e Baía de Guanabara (RJ): Efeitos tóxicos em minhocas"* está relacionado ao meio ambiente.

3. Comentários Finais

O sistema Busc@NIMA é um exemplo de Sistema de Recuperação de Informações que identifica disciplinas oferecidas, projetos e trabalhos de pesquisa ou desenvolvimento, de professores e pesquisadores na PUC-Rio, possuindo o escopo reduzido apenas a uma determinada área, no caso, a área de meio ambiente. No momento estamos estudando estender nossa pesquisa com a possibilidade de implementar bases ou grafos de conhecimento para restringir o contexto de buscas semânticas. O sistema foi desenvolvido sob encomenda para a PUC-Rio e seu Núcleo Interdisciplinar de Meio Ambiente (NIMA) mas seu código está disponível para ser utilizado e estendido por qualquer Universidade ou Instituição de Pesquisa.

Referências

Baeza-Yates, R. and Ribeiro-Neto, B. A. (2011). *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd.

Salgueiro, M., dos Santos, V., R., A.L.C., Guimaraes, D., Haeusler, E.H., d. S., J., V., and M.V., Lifschitz, S. (2021). Quem@puc - a tool to find researchers at puc-rio. In *Anais Estendidos (DEMOS) do Simpósio Brasileiro de Bancos de Dados (SBBD)*, pages 93–98.