

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
BIBLIOTECA UNIVERSITÁRIA**

Carlos Bonetti

**ANÁLISE DO USO DE FEEDBACK DE RELEVÂNCIA
NO SISTEMA DE INTEGRAÇÃO LATTES-QUALIS
(SILQ)**

Florianópolis

2016

Carlos Bonetti

**ANÁLISE DO USO DE FEEDBACK DE RELEVÂNCIA
NO SISTEMA DE INTEGRAÇÃO LATTES-QUALIS
(SILQ)**

Tese submetida ao Programa de Graduação em Ciência da Computação para a obtenção do Grau de Bacharel.
Orientador: Prof. Dr. Carina F. Dorneles

Florianópolis

2016

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Bonetti, Carlos

Análise do uso de feedback de relevância no Sistema de
Integração Lattes-Qualis (SILQ) / Carlos Bonetti ;
orientadora, Carina Friedrich Dorneles - Florianópolis, SC,
2016.

70 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico.
Graduação em Ciências da Computação.

Inclui referências

1. Ciências da Computação. 2. Feedback de Relevância. 3.
Similaridade. 4. Data-Matching. I. Dorneles, Carina
Friedrich. II. Universidade Federal de Santa Catarina.
Graduação em Ciências da Computação. III. Título.

Carlos Bonetti

**ANÁLISE DO USO DE FEEDBACK DE RELEVÂNCIA
NO SISTEMA DE INTEGRAÇÃO LATTES-QUALIS
(SILQ)**

Esta Tese foi julgada aprovada para a obtenção do Título de “Bacharel”, e aprovada em sua forma final pelo Programa de Graduação em Ciência da Computação.

Florianópolis, 01 de outubro 2016.

Prof. Dr. Renato Cislaghi
Coordenador de Projetos

Banca Examinadora:

Prof. Dr. Carina F. Dorneles
Orientador

Prof. Dr. Maurício Floriano Galimberti

Prof. Dr. Márcio Bastos Castro

AGRADECIMENTOS

Aos meus pais, Ivanete e Valsir, por tornarem possível minha jornada acadêmica; à minha irmã Daiana por facilitá-la ainda mais.

Aos meus amigos Felipe Nedel e Maria Eloísa Costa por acreditarem em mim na tarefa de levar adiante o trabalho começado por eles.

À minha orientadora, Prof. Carina Dorneles, pelos valiosos conselhos durante o processo de realização deste trabalho.

A todos os meus amigos e pessoas especiais que me incentivaram no decorrer do curso e na realização deste trabalho.

Meus sinceros agradecimentos.

RESUMO

Os indicadores da qualidade de produções científicas vinculadas a Programas de Pós-Graduação são gerados através da qualificação dos veículos onde essas produções foram publicadas. O SILQ surgiu em 2015 com o objetivo de integrar o Lattes, uma plataforma para cadastro de pesquisadores, com o Qualis, um conjunto de procedimentos da CAPES para classificação da produção intelectual. O objetivo do SILQ é realizar o *matching* automático dos veículos qualificados no Qualis com os veículos cadastrados no currículo Lattes de pesquisadores, a fim de automatizar o processo de geração de indicadores das produções contidas nesses currículos. Tal ferramenta foi disponibilizada pública e gratuitamente, porém carece de atualizações dos dados Qualis mais recentes, mudanças de arquitetura e de usabilidade, e, em especial, inclusão de controles que permitam ao usuário sugerir *matchings* ao sistema, técnica denominada *feedback de relevância* e amplamente utilizada no contexto de *Information Retrieval (IR)*. O objetivo do uso desta técnica no SILQ é diminuir os falsos positivos retornados pelo sistema, bem como aumentar a precisão dos resultados de maneira geral. Tais modificações foram realizadas por este trabalho, dando origem a uma nova versão do SILQ, junto com a proposta e avaliação experimental de dois novos algoritmos que utilizam as informações de *feedback de relevância* para melhorar a precisão do sistema. Conforme os experimentos realizados, ambos os algoritmos propostos foram capazes de melhorar a taxa de acerto do sistema, ou seja, aumentar o número de publicações corretamente avaliadas.

Palavras-chave: Similaridade. Lattes. Qualis. Feedback de Relevância. SILQ.

ABSTRACT

The quality indicators of scientific publications linked to graduate programs are measured through the qualification of the vehicles where they were published. SILQ emerged in 2015 with the purpose of integrating the Lattes platform, a register system for researchers, with Qualis, a set of procedures by CAPES for qualification of intellectual production. The SILQ's goal is to match vehicles registered on Qualis with vehicles registered on the Lattes curriculum of researchers, in order to automate the process of generating quality indicators of scientific production from these curriculums. Such tool was provided publicly and for free but lacks of the most recent Qualis data update, architectural and usability changes and, especially, inclusion of controls allowing the user to suggest matchings to the system, a technique named *relevance feedback* and widely used in Information Retrieval (IR) context. The purpose to use this technique on SILQ is to lower the false positive values returned by the system and to improve the overall precision. These changes were made by this work, generating a new SILQ version, alongside with the proposal and experimental evaluation of two new algorithms that use these feedback information to improve the system precision. According to the experiments, both algorithms were capable to improve the success rate of the system maximizing the number of publications correctly assessed.

Keywords: Similarity. Lattes. Qualis. Relevance Feedback. SILQ.

LISTA DE FIGURAS

| | | |
|----------|---|----|
| Figura 1 | Exemplo de conjunto de dados do Qualis 2014 | 25 |
| Figura 2 | Diálogo de configurações de avaliação | 43 |
| Figura 3 | Nova arquitetura do SILQ | 46 |
| Figura 4 | Página de resultados de avaliação do SILQ 1 | 48 |
| Figura 5 | Página de resultados de avaliação do SILQ 2 | 48 |
| Figura 6 | Novo esquema do modelo lógico do SILQ | 49 |
| Figura 7 | Detalhe dos controles de feedback de relevância na página de avaliação de currículo Lattes | 52 |
| Figura 8 | Taxa de trabalhos corretamente avaliados e Média de Rank Recíproco (MRR) para diferentes <i>thresholds</i> | 60 |
| Figura 9 | Comparação da taxa de acerto do algoritmo <i>trgm</i> e do <i>trgm + query_aliasing</i> para diferentes <i>thresholds</i> | 62 |

LISTA DE TABELAS

| | | |
|----------|---|----|
| Tabela 1 | Resultados retornados pelo SILQ para a <i>query</i> “2016 IEEE 7th Latin American Symposium on Circuits & Systems (LASCAS)” | 42 |
| Tabela 2 | Número de periódicos extraídos dos dados Qualis | 45 |
| Tabela 3 | Resultados retornados pelo SILQ para a <i>query</i> “Software Engineering Knowledge Engineering” | 53 |
| Tabela 4 | Resultados retornados pelo SILQ para a <i>query</i> “Software Engineering and Knowledge Engineering” utilizando <code>query_aliasing</code> | 56 |
| Tabela 5 | Comparação da exatidão dos diferentes algoritmos testados (utilizando <i>threshold</i> de 0.55)..... | 61 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|-------|---|
| AJAX | Asynchronous JavaScript and XML |
| API | Application Program Interface |
| CAPES | Coordenação de Aperfeiçoamento de Pessoal de Nível Superior |
| CNPq | Conselho Nacional de Desenvolvimento Científico e Tecnológico |
| CSS | Cascading Style Sheets |
| CSV | Comma-Separated Values |
| HTTP | Hypertext Transfer Protocol |
| IR | Information Retrieval |
| ISSN | International Standard Serial Number |
| JSON | JavaScript Object Notation |
| MCTI | Ministério de Ciência, Tecnologia e Inovação |
| MRR | Mean Reciprocal Rank |
| PDF | Portable Document Format |
| REST | Representational state transfer |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| W3C | World Wide Web Consortium |
| XLS | eXcel Spreadsheet |
| XML | Extensible Markup Language |

SUMÁRIO

| | | |
|----------|---|----|
| 1 | INTRODUÇÃO | 19 |
| 1.1 | OBJETIVOS | 21 |
| 1.2 | PROCEDIMENTOS METODOLÓGICOS | 21 |
| 2 | FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS | 23 |
| 2.1 | PRODUÇÃO E AVALIAÇÃO CIENTÍFICAS NO BRASIL | 23 |
| 2.1.1 | Qualis | 24 |
| 2.1.2 | Lattes | 25 |
| 2.2 | SISTEMAS DE <i>IR</i> E MÉTRICAS DE AVALIAÇÃO | 26 |
| 2.2.1 | <i>Data-Matching</i> | 26 |
| 2.2.2 | Feedback de Relevância | 28 |
| 2.2.3 | Avaliação de sistemas <i>IR</i> | 29 |
| 2.2.3.1 | Precisão e Revocação | 29 |
| 2.2.3.2 | <i>Precision at k (P@k)</i> e <i>R-Precision</i> | 29 |
| 2.2.3.3 | Média de Rank Recíproco | 30 |
| 2.2.3.4 | Conjunto de testes | 31 |
| 2.3 | ARQUITETURA DE SISTEMAS BASEADOS NA <i>WEB</i> | 31 |
| 2.3.1 | <i>Web services</i> e a arquitetura <i>REST</i> | 31 |
| 2.3.2 | Teste automatizado de software | 33 |
| 2.4 | TRABALHOS CORRELATOS | 34 |
| 2.4.1 | Quanto ao objetivo da ferramenta | 35 |
| 2.4.2 | Quanto ao uso de <i>feedback</i> de relevância | 36 |
| 3 | SILQ 2: ALTERAÇÕES ESTRUTURAIS | 37 |
| 3.1 | HISTÓRICO E VISÃO GERAL DO SILQ 1 | 37 |
| 3.1.1 | “Trabalhos futuros” citados pelo SILQ 1 | 38 |
| 3.1.2 | Como o SILQ realiza o <i>matching</i> Qualis-Lattes | 41 |
| 3.2 | SILQ 2 | 44 |
| 3.2.1 | Extração e inserção dos novos dados Qualis | 44 |
| 3.2.2 | Criação do <i>Web Service</i> | 45 |
| 3.2.3 | Alterações no <i>front-end</i> | 47 |
| 3.2.4 | Alterações no modelo lógico | 49 |
| 3.2.5 | Garantia da qualidade | 50 |
| 4 | USO DE <i>FEEDBACK</i> DE RELEVÂNCIA | 51 |
| 4.1 | OBTENÇÃO DE <i>FEEDBACK</i> | 51 |
| 4.2 | ALGORITMOS DE AVALIAÇÃO COM <i>FEEDBACK</i> | 52 |
| 4.2.1 | Algoritmo <i>fb(t)</i> | 53 |
| 4.2.2 | Algoritmo <i>query.aliasing</i> | 55 |

| | | |
|-------|---|----|
| 4.3 | AVALIAÇÃO EXPERIMENTAL | 57 |
| 4.3.1 | Conjunto de testes | 57 |
| 4.3.2 | Métricas utilizadas | 58 |
| 4.3.3 | Avaliação de <i>threshold</i> ideal | 59 |
| 4.3.4 | Avaliação dos algoritmos | 61 |
| 5 | CONCLUSÕES E TRABALHOS FUTUROS | 63 |
| | REFERÊNCIAS | 65 |
| | APÊNDICE A – Artigo | 71 |

1 INTRODUÇÃO

A Plataforma Lattes, criada e mantida pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), é um sistema de informação responsável pela integração da base de dados de currículos, grupos de pesquisa e instituições. O Currículo Lattes se tornou o padrão nacional no registro da vida científica de estudantes, professores e pesquisadores e é hoje adotada por institutos e universidades de todo o país (CNPQ, 2015b). No Currículo Lattes pode-se inserir dados gerais do pesquisador, produção bibliográfica, orientações, citações, participações em eventos científicos entre outros dados. No módulo Produção Bibliográfica, por exemplo, é possível a inserção de artigos publicados ou aceitos para publicação em periódicos indexados pelo ISSN (CNPQ, 2015a).

A qualidade da produção bibliográfica dos Programas de Pós-Graduação é classificada pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) através de um conjunto de procedimentos denominado Qualis. O Qualis define estratos de qualificação a partir da análise da qualidade dos veículos onde a produção científica de pesquisadores brasileiros foi divulgada, ou seja, periódicos e eventos científicos. Esta análise é realizada em um processo anual de atualização, sendo os veículos enquadrados nos estratos indicativos de qualidade A1, A2, B1, B2, B3, B4, B5 e C, do maior para o menor peso, para cada área do conhecimento (CAPES, 2015a).

Apesar da Plataforma Lattes possuir um módulo de inclusão de publicações e permitir a definição do veículo onde este foi publicado, não há qualquer tipo de conexão entre os sistemas Lattes e Qualis, ou seja, o processo de avaliação de uma publicação (que é feita através da avaliação do veículo onde este foi publicado) deve ser realizado de forma manual.

O Sistema de Integração Lattes Qualis (SILQ) surgiu no ano de 2015, desenvolvido como Trabalho de Conclusão de Curso dos alunos Felipe Nedel Mendes de Aguiar e Maria Eloísa Costa do curso de Ciência da Computação da Universidade Federal de Santa Catarina (UFSC), orientados pela Prof. Carina F. Dorneles, da mesma instituição. O objetivo do sistema é a classificação automática da produção científica do currículo Lattes do pesquisador, através do *matching* por similaridade de dados extraídos do Qualis, usando uma interface web amigável (AGUIAR; COSTA, 2015).

A primeira versão do sistema foi finalizada em 2015 e desde então

encontra-se disponível de forma pública e gratuita através do endereço <http://silq.inf.ufsc.br/>. No entanto, esta primeira versão apresenta alguns pontos que merecem melhorias.

O SILQ possui em seu banco de dados de eventos e periódicos extraídos do Qualis apenas dados referentes ao triênio 2010-2012, já que, até metade de 2015, o Qualis realizava classificações de forma trienal. Em 2016, no entanto, após a finalização do projeto SILQ, o Qualis alterou seu modo de operação para classificações anuais, já disponibilizando dados referentes aos anos 2013 e 2014, que ainda não estavam incluídos nesta primeira versão da base de dados SILQ. O primeiro ponto de melhoria é fazer com que o SILQ permita a atualização automática da qualificação a cada novo ano de lançamento do Qualis. Como o formato no qual a CAPES disponibiliza o Qualis não é interoperável, o SILQ deve oferecer esta funcionalidade.

O segundo ponto a ser trabalhado é a falta de alguns recursos que poderiam facilitar o processo de avaliação e acompanhamento de currículos, como gráficos de classificações dentro Programas de Pós-Graduação. Uma característica importante também seria a capacidade de integração com outros sistemas, facilitando o reuso do serviço disponibilizado pelo SILQ em trabalhos futuros.

O terceiro ponto de melhoria seria a avaliação experimental do algoritmo de classificação do SILQ para o estabelecimento de uma medida do grau de precisão da ferramenta. A partir disto, outras técnicas podem ser propostas e comparadas com as anteriores a fim de provar se elas resultaram em ganho de precisão.

Como o *matching* entre os dados do Qualis e do Lattes de pesquisadores é feito através de funções de similaridade, o resultado pode ser um falso positivo. Portanto, um ponto importante a ser trabalhado é permitir ao usuário informar ao sistema sugestões de resultados. Esta técnica, conhecida por *feedback* de relevância, exige alterar o algoritmo de classificação a fim de treiná-lo com as informações providas pelo usuário e melhorar os resultados de pesquisas similares subsequentemente realizadas. Também se faz necessária a mensuração da taxa de acerto do sistema após a inclusão desta técnica e avaliar se ela beneficia a precisão do algoritmo de classificação.

A proposta deste trabalho, portanto, é a criação de uma segunda versão do SILQ, desenvolvida a partir do código existente da primeira, a fim de realizar as mudanças sugeridas acima, junto com a atualização da base de dados e arquitetura do sistema para a inclusão das novas qualificações Qualis em um ritmo anual.

A hipótese de pesquisa a ser avaliada a partir desta ideia, por-

tanto, é avaliar se o *feedback* de usuários pode melhorar a taxa de acerto do SILQ, ou seja, aumentar o número de trabalhos corretamente avaliados pelo sistema.

1.1 OBJETIVOS

O objetivo geral deste trabalho é analisar o impacto que o uso de *feedback* de relevância tem na precisão dos resultados de avaliações realizadas pelo SILQ, efetuado sobre uma nova arquitetura da ferramenta que inclui a criação de API de integração com outros sistemas e a atualização da base de dados conforme as novas classificações Qualis.

Os objetivos específicos são os seguintes:

- Reestruturação da arquitetura e banco de dados do SILQ a fim de suportar classificações de eventos e periódicos disponibilizados em um ritmo anual;
- Atualização do banco de dados do sistema com as últimas classificações disponibilizadas pelo Qualis (anos 2013 e 2014);
- Criação de uma API pública de disponibilização dos serviços do SILQ, via camada de aplicação REST para integração com outros sistemas;
- Alterações na interface do sistema incluindo migração de *framework* de interface, inclusão de controles de *feedback*, novos gráficos de acompanhamento de grupos de pesquisa e melhorias gerais de usabilidade;
- Propor novos algoritmos de avaliação baseados em similaridade textual e *feedback* de relevância e verificar se a taxa de acerto do sistema foi melhorada com tal ação.

1.2 PROCEDIMENTOS METODOLÓGICOS

A primeira etapa do trabalho envolveu a migração tecnológica da camada de interface e desacoplamento entre *server* e *client side*, originando um serviço *RESTful* de integração. Para isto, foram utilizadas práticas ágeis de desenvolvimento de software incluindo desenvolvimento orientado a testes e eventuais modelagens lógicas de arquitetura de sistemas usando práticas da Engenharia de Software.

Concomitantemente, ocorreu a alteração da base de dados para suporte aos novos registros disponibilizados pelo Qualis. Certos aspectos do banco foram alterados para suportar tais registros novos. Os dados também passaram por um processo de *data-cleaning* manual antes de serem inseridos no banco, conforme detalhado na Seção 3.2.1.

A segunda etapa do trabalho envolveu o processo de construção do módulo de captação de *feedback* de relevância, envolvendo a modelagem lógica, criação da camada de serviço, API e alterações na página de resultados de avaliação do sistema com a inclusão dos controles que permitem aos usuários “julgar” os resultados retornados. Após esta etapa o algoritmo de classificação baseado em similaridade textual do SILQ foi estendido para levar em conta os *feedbacks* dos usuários. Deste processo surgiram dois algoritmos que passaram por uma avaliação experimental, realizada usando um conjunto de testes contendo trabalhos extraídos de currículos Lattes de pesquisadores e manualmente avaliados por um especialista, conforme relatado na Seção 4.3.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Esta revisão de literatura não será exaustiva ao ponto de revisar os mesmos assuntos abordados no trabalho original de (AGUIAR; COSTA, 2015) e focará somente nos pontos onde este novo trabalho divergiu. Alguns assuntos fundamentais, entretanto, foram expostos novamente com o intuito de facilitar o processo de entendimento deste TCC como um trabalho independente. É o caso, por exemplo, das seções relacionadas aos Qualis, Lattes e das tecnologias base utilizadas no sistema.

Em um primeiro momento, é apresentada a problemática envolvida no processo de avaliação da produção científica no Brasil e a motivação para a criação do SILQ. Após esta Seção, são introduzidos conceitos relacionados a sistemas de Recuperação de Informação e Feedback de Relevância, tópicos explorados por este trabalho para a tentativa de aumento da exatidão do algoritmo de classificação e avaliação da eficácia do sistema. Por último, são apresentados conceitos e padrões arquiteturais e tecnológicos que foram explorados para o desenvolvimento da segunda versão do SILQ.

2.1 PRODUÇÃO E AVALIAÇÃO CIENTÍFICAS NO BRASIL

A produção científica é uma das atividades universitárias que cumpre uma função básica dentro das instituições e que merece notável destaque. É através dela que o conhecimento produzido na Universidade é difundido à sociedade, externalizando descobertas para outros pesquisadores e para a própria comunidade (CRISPIM, 2014).

No Brasil, o Ministério de Ciência, Tecnologia e Inovação (MCTI) é o órgão de administração direta que possui, entre sua lista de competências, a política nacional de pesquisa científica, tecnológica e inovação; e o planejamento, coordenação, supervisão e controle das atividades da ciência e tecnologia (MCTI, 2016). Uma das agências subordinadas ao MCTI é o CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), órgão que tem como principais atribuições “fomentar a pesquisa científica e tecnológica e incentivar a formação de pesquisadores brasileiros” (CNPQ, 2016).

Analisando não só a produção científica, cabe ao Ministério da Educação (MEC), através de sua fundação subordinada, a CAPES (Co-

ordenação de Aperfeiçoamento de Pessoal de Nível Superior) a avaliação dos programas de pós-graduação das universidades Brasileiras (CAPES, 2015b). Um dos principais critérios de avaliação se dá pela quantidade e qualidade das produções científicas, através de um conjunto de procedimentos denominado Qualis (CAPES, 2015a).

2.1.1 Qualis

A classificação da qualidade da produção intelectual é realizada pela CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) através de um conjunto de procedimentos denominado Qualis. O Qualis afere a qualidade de artigos e outras produções a partir da análise da qualidade dos veículos de divulgação, ou seja, periódicos científicos. Esta classificação é realizada pelas áreas de avaliação em um processo anual de atualização, sendo os veículos enquadrados em estratos indicativos de qualidade: A1 - o mais elevado; A2; B1; B2; B3; B4; B5; C - com peso zero (CAPES, 2015a).

O resultado desta estratificação é uma lista com a classificação dos veículos de publicação utilizados pelos programas de pós-graduação (*journals*, revistas, periódicos etc), separados por área e ano de avaliação (um mesmo veículo tem uma avaliação diferente para cada ano de avaliação e área de conhecimento). Desta forma, a avaliação da qualidade de uma produção de um pesquisador é realizada de forma indireta, atribuindo-se a esta produção o estrato dado àquele veículo onde ela foi publicada.

Note-se que o mesmo periódico, ao ser classificado em duas ou mais áreas distintas, pode receber diferentes avaliações. Isto não constitui inconsistência, mas expressa o valor atribuído, em cada área, à pertinência do conteúdo veiculado. Por isso, não se pretende com esta classificação que é específica para o processo de avaliação de cada área, definir qualidade de periódicos de forma absoluta (CAPES, 2015a).

Até o ano de 2014, como cita o trabalho de (AGUIAR; COSTA, 2015, p. 31), os dados dos resultados das avaliações Qualis encontravam-se estruturados e disponibilizados em documentos no formato PDF e XLS. Também encontravam-se disponível apenas dados referentes ao triênio de 2010-2012. Em 2015, porém, a CAPES alterou seu modo de operação e passou a realizar avaliações anuais, já disponibilizando os novos dados de 2013 e 2014, junto com os antigos, em formato CSV

através do WebQualis¹ (Hoje plataforma Sucupira).

A Figura 1 mostra um exemplo de dados extraídos da tabela Qualis de 2014, contendo ISSN, título, área de avaliação e estrato atribuído a cada periódico avaliado.

Figura 1 – Exemplo de conjunto de dados do Qualis 2014

| ISSN | Título | Área de Avaliação | Estrato |
|-----------|--|-----------------------|---------|
| 1041-4347 | IEEE Transactions on Knowledge and Data Engineering (Print) | CIÊNCIA DA COMPUTAÇÃO | A1 |
| 0018-9464 | IEEE Transactions on Magnetics | CIÊNCIA DA COMPUTAÇÃO | B4 |
| 0278-0062 | IEEE Transactions on Medical Imaging (Print) | CIÊNCIA DA COMPUTAÇÃO | A1 |
| 1536-1233 | IEEE Transactions on Mobile Computing | CIÊNCIA DA COMPUTAÇÃO | A2 |
| 1520-9210 | IEEE Transactions on Multimedia | CIÊNCIA DA COMPUTAÇÃO | A2 |
| 2162-237X | IEEE Transactions on Neural Networks and Learning Systems | CIÊNCIA DA COMPUTAÇÃO | A1 |
| 0018-9499 | IEEE Transactions on Nuclear Science | CIÊNCIA DA COMPUTAÇÃO | B1 |
| 1045-9219 | IEEE Transactions on Parallel and Distributed Systems (Print) | CIÊNCIA DA COMPUTAÇÃO | A2 |
| 0885-8950 | IEEE Transactions on Power Systems | CIÊNCIA DA COMPUTAÇÃO | B2 |
| 0098-5589 | IEEE Transactions on Software Engineering | CIÊNCIA DA COMPUTAÇÃO | A1 |
| 1083-4427 | IEEE Transactions on Systems, Man and Cybernetics. Part A. Systems, Man, and Cybernetics | CIÊNCIA DA COMPUTAÇÃO | A2 |
| 1094-6977 | IEEE Transactions on Systems, Man and Cybernetics. Part C, Applications and Reviews | CIÊNCIA DA COMPUTAÇÃO | A2 |
| 0018-9545 | IEEE Transactions on Vehicular Technology | CIÊNCIA DA COMPUTAÇÃO | A1 |
| 1063-8210 | IEEE Transactions on Very Large Scale Integration (VLSI) Systems | CIÊNCIA DA COMPUTAÇÃO | A2 |
| 1077-2626 | IEEE Transactions on Visualization and Computer Graphics | CIÊNCIA DA COMPUTAÇÃO | A2 |
| 1536-1276 | IEEE Transactions on Wireless Communications | CIÊNCIA DA COMPUTAÇÃO | A1 |
| 1536-1284 | IEEE Wireless Communications | CIÊNCIA DA COMPUTAÇÃO | A1 |
| 2162-2337 | IEEE Wireless Communications Letters | CIÊNCIA DA COMPUTAÇÃO | B4 |
| 1932-4537 | IEEE eTransactions on Network and Service Management | CIÊNCIA DA COMPUTAÇÃO | B3 |
| 1932-8540 | IEEE-RITA | CIÊNCIA DA COMPUTAÇÃO | B5 |
| 1545-5963 | IEEE/ACM Transactions on Computational Biology and Bioinformatics | CIÊNCIA DA COMPUTAÇÃO | B1 |
| 0916-8532 | IEICE Transactions on Information and Systems | CIÊNCIA DA COMPUTAÇÃO | B1 |
| 1751-861X | IET Computers & Digital Techniques (Online) | CIÊNCIA DA COMPUTAÇÃO | B1 |
| 1751-8601 | IET Computers & Digital Techniques (Print) | CIÊNCIA DA COMPUTAÇÃO | B1 |
| 1751-8806 | IET Software (Print) | CIÊNCIA DA COMPUTAÇÃO | B1 |
| 1091-9856 | INFORMS Journal on Computing | CIÊNCIA DA COMPUTAÇÃO | B1 |
| 1526-5528 | INFORMS Journal on Computing (Online) | CIÊNCIA DA COMPUTAÇÃO | B1 |
| 2090-4355 | ISRN Communications and Networking | CIÊNCIA DA COMPUTAÇÃO | C |
| 2237-4523 | Iberoamerican Journal of Applied Computing | CIÊNCIA DA COMPUTAÇÃO | B5 |
| 0262-8856 | Image and Vision Computing | CIÊNCIA DA COMPUTAÇÃO | A1 |

2.1.2 Lattes

Desde a década de 80 notava-se a necessidade de um formulário padrão para registro dos currículos de pesquisadores brasileiros que possibilitasse a busca e seleção de especialistas bem como a geração de um mapa sobre a distribuição da pesquisa científica no Brasil. Em 1999, o CNPq lançou a Plataforma Lattes², um sistema de informação para registro e consulta de currículos acadêmicos, e padronizou o Currículo Lattes como o formulário de currículo a ser utilizado no âmbito do Ministério da Ciência e Tecnologia e CNPq (CNPQ, 2016, 2015b).

¹<http://qualis.capes.gov.br/webqualis/>

²<http://lattes.cnpq.br/>

Entre os registros contidos em um currículo Lattes, estão os dados gerais do pesquisador, produção bibliográfica, orientações, citações, formação acadêmica, etc. No módulo Produção Bibliográfica, por exemplo, é possível a inserção de artigos publicados em periódicos indexados pelo ISSN, livros e capítulos, textos em jornais ou revistas, trabalhos publicados em anais de eventos, entre outras formas de produções (CNPQ, 2015a).

A Plataforma Lattes, além de apresentar seus currículos através de uma interface web usando HTML, também disponibiliza os dados em formato XML, uma linguagem de marcação semiestruturada legível por humanos e máquinas (AGUIAR; COSTA, 2015, p. 34). Foi este o formato utilizado pelo SILQ para a extração dos dados Lattes de um pesquisador, conforme relatado em (AGUIAR; COSTA, 2015).

2.2 SISTEMAS DE IR E MÉTRICAS DE AVALIAÇÃO

Recuperação de Informação, ou *Information Retrieval* (IR), é a obtenção de informações relevantes, dados certos critérios de busca, de uma coleção maior de informações. Os maiores exemplos atuais de sistemas de IR são os buscadores web (BAEZA-YATES; RIBEIRO-NETO, 1999). Áreas tipicamente relacionadas com IR e cujas técnicas e definições muita vezes se intersectam são as de *Data Mining*, Sistemas de Recomendação e *Data-Matching*.

2.2.1 *Data-Matching*

Data matching é o processo de comparação entre informações de diferentes fontes com o objetivo de descobrir se de fato representam o mesmo objeto do mundo real (DORNELES; GONÇALVES; MELLO, 2011). Neste contexto, é denominada de *query* toda consulta submetida ao sistema de IR e de *documentos* o conjunto de dados sobre o qual ocorrerá a pesquisa e do qual, possivelmente, resultará um subconjunto contendo os resultados para a *query*. Desta forma, o algoritmo de avaliação de currículos do SILQ pode ser classificado como um sistema de IR baseado em *data matching* aproximado, utilizando uma função de similaridade textual entre os dados de publicações informados no currículo Lattes de pesquisadores e os registros Qualis para classificação dos resultados.

Um processo de *data matching* aproximado emprega funções de similaridade (ou de distância) para comparar informações, atribuindo

um valor, geralmente pertencente ao intervalo real $[0, 1]$, ao quão parecidos (similares) são os dados comparados. Esse valor é chamado de valor de similaridade, no caso da função calcular o quão parecidos são os objetos, ou valor de dissimilaridade no caso da função calcular o quão diferentes são os objetos. Dois objetos são considerados similares se o valor atribuído pela função de similaridade for maior do que um outro valor pré-estipulado pertencente ao intervalo $[0, 1]$. Esse valor é denominado de *threshold* e deve ser estipulado pelo projetista de acordo com a características do sistema, que incluem a função de similaridade utilizada e a natureza semântica e estrutural dos documentos (DORNELES; GONÇALVES; MELLO, 2011).

A função de similaridade utilizada pelo algoritmo de avaliação de currículos do SILQ é a *n-gram*, com valor de n igual a 3 (sendo chamado, neste caso, de *trigram*) (AGUIAR; COSTA, 2015). Um *n-gram* é uma sequência de n caracteres contíguos extraídos de um texto qualquer. Por exemplo, a palavra “Revista” contém o trigramma “Rev”. O conjunto de todos os trigramas possíveis para este mesmo exemplo é $\{_R, _Re, Rev, evi, vis, ist, sta, ta_ \}$ (o início e final da palavra são modelados com “espaços” por certas funções para explicitamente indicar o início e o final da sequência). A função de similaridade *n-grams* recebe como parâmetro duas sequências de caracteres, calcula os conjuntos de *n-gramas* de ambas e retorna a similaridade dos parâmetros como sendo a fração de elementos em comum dos dois conjuntos (BRODER et al., 1997).

Desta forma, ao avaliar uma publicação qualquer, o SILQ compara, através da função *trigram*, o nome do veículo onde esta publicação foi submetida com cada um dos registros Qualis presentes na base de dados. Se o valor de similaridade for 0.6 ou maior³, o veículo é considerado como um *matching* positivo. O evento com maior similaridade, caso exista, é considerado e seu estrato Qualis é atribuído à publicação que está sendo avaliada.

Algumas questões relacionadas a este processo são a definição do valor de *threshold* ideal para maximizar a precisão do algoritmo (DORNELES; GONÇALVES; MELLO, 2011) e a própria função de similaridade mais indicada para o conjunto de dados em questão (SILVA et al., 2007). Para ambas as questões, entretanto, é preciso estipular métricas para a medição de quão preciso é o algoritmo de similaridade, para só então testar diferentes técnicas e variáveis até chegar a uma configuração ideal para o conjunto de dados trabalhados. Duas medidas clássicas de IR

³Este valor é o *threshold* padrão para avaliações, mas pode ser configurado pelo sistema.

que podem ser utilizadas neste caso são a precisão (*precision*) e a revocação (*recall*), além de outras derivadas mais indicadas para sistemas que utilizam *ranking*, conforme descrito na Seção 2.2.3.

2.2.2 Feedback de Relevância

Feedback de relevância (ou *Relevance Feedback*) é uma característica de alguns sistemas de recuperação de informação que utilizam dados providos pelo usuário para aumentar a exatidão de suas respostas. De maneira particular, o usuário dá *feedback* sobre um conjunto inicial de valores retornados pelo sistema, que então passa a considerar as informações de resultados relevantes ou não relevantes para processar novas *queries* (MANNING; RAGHAVAN; SCHUTZE, 2008, p. 178).

Tradicionalmente, sistemas que utilizam esta técnica dão a possibilidade para o usuário informar palavras-chave, selecionar ou marcar resultados relevantes ou não relevantes ou responder questões sobre seus interesses (KELLY; TEEVAN, 2003). Pode-se distinguir dois tipos de *feedback* de usuário: explícito e implícito. No primeiro, o usuário interage com o sistema alimentando-o com informações de julgamento (marcando resultados relevantes, selecionando melhores resultados etc); no segundo, o sistema infere que alguns resultados foram melhores que outros por métricas derivadas, como itens selecionados para visualização, tempo de permanência na página ou ações de *scroll* do navegador (WHITE; JOSE; RUTHVEN, 2001). *Feedback* implícito, de maneira geral, é menos confiável do que explícito por utilizar fontes indiretas de evidência (MANNING; RAGHAVAN; SCHUTZE, 2008).

Segundo (MANNING; RAGHAVAN; SCHUTZE, 2008), *feedback* de relevância pode proporcionar ganhos substanciais na performance de sistemas de recuperação de informação, porém a avaliação de sua eficácia em um sistema de IR pode não ser trivial. Um dos métodos para tal mensuração é traçar gráficos de precisão-revocação antes e depois de uma rodada de *feedback* do usuário e comparar os resultados. Esta técnica, porém, é trapaceira pois os documentos julgados pelo usuário com certeza serão melhor ranqueados. Para uma avaliação justa devem ser levados em conta somente documentos que não foram vistos pelo usuário. Um outro método de avaliação requer dois conjuntos, o primeiro utilizado para a *query* inicial e para receber *feedback* do usuário, e o segundo para avaliação via comparação, que contém os resultados desejados previamente estabelecidos por um especialista. A performance do sistema pode ser medido através da comparação dos dois conjuntos,

antes e após a inclusão dos dados de *feedback* do usuário.

2.2.3 Avaliação de sistemas IR

Um dos desafios em projetar sistemas de IR está em escolher técnicas que sejam efetivas para a aplicação. Um processo importante no desenvolvimento de tais aplicações, portanto, é o de avaliação, que envolve a medição de certos aspectos do sistema baseados em métricas (MANNING; RAGHAVAN; SCHUTZE, 2008).

2.2.3.1 Precisão e Revocação

Em sistemas de reconhecimento de padrões e/ou informações baseados em classificações binárias⁴, precisão (ou valor preditivo positivo) é a fração dos objetos retornados que são relevantes, enquanto a revocação (ou sensibilidade) é a fração de todos os objetos relevantes que foram retornados (SAMMUT; WEBB, 2011). Desta forma, *precision* (precisão) e *recall* (revocação) podem ser consideradas medidas de correção e completude e são métricas base para o entendimento da relevância dos resultados retornados (EUZENAT, 2007).

Entretanto, precisão e revocação são medidas baseadas em conjuntos, computadas usando coleções de documentos não ordenados, ou seja, que são retornados sem uma distinção de ordem. São necessárias diferentes medidas para avaliar resultados ordenados (MANNING; RAGHAVAN; SCHUTZE, 2008).

2.2.3.2 *Precision at k* ($P@k$) e *R-Precision*

Para alguns sistemas de IR, em especial aqueles que manipulam grandes volumes de dados, não são relevantes todas as respostas a uma *query*, mas somente os primeiros k resultados. Neste caso, podem ser realizadas mensurações da precisão considerando somente um certo limite de resultados, por exemplo, 10 ou 20 documentos. Este valor é conhecido como *precision at k*, por exemplo, *precision at 10*, sendo 10 o número de resultados considerados (MANNING; RAGHAVAN; SCHUTZE, 2008).

⁴Que só fazem distinção entre dois possíveis resultados para uma busca: ou o elemento é relevante para a *query* em questão ou não é.

Uma das desvantagens desta medida é sua instabilidade em comparação com outras, principalmente de sua média já que o número total de documentos relevantes para uma *query* tem uma forte influência no valor. Um exemplo é uma *query* que só possua 1 item relevante e que ele seja o primeiro resultado retornado, mas ainda assim o valor de *precision at 10* é 0.1, já que outros 9 itens não relevantes foram retornados.

Uma alternativa que alivia este problema é chamado de *R-Precision*. Esta medida requer que o conjunto R de todos os documentos relevantes para certa *query* seja conhecido. A partir deste conjunto calcula-se a precisão de uma *query* que retornou r resultados relevantes como sendo $r/|R|$, ou seja, a fração dos itens relevantes que foram de fato retornados. Neste caso, ao analisar os primeiros $|R|$ resultados de uma *query*, por definição, não apenas o valor da precisão é $r/|R|$ como o próprio valor de revocação também é $r/|R|$, ou seja, os valores de precisão e revocação são idênticos ao utilizar tal métrica (MANNING; RAGHAVAN; SCHUTZE, 2008).

2.2.3.3 Média de Rank Recíproco

A média de rank recíproco (*MRR - Mean Reciprocal Rank*), é uma medida estatística utilizada na avaliação de processos que produzem uma lista de resultados a uma *query* ordenados por probabilidade de correteude (LIU; ZSU, 2009).

A média é dada por:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank(i)}$$

Sendo Q o conjunto de *queries* sobre a qual a média é calculada e $rank(i)$ a posição do primeiro item relevante para a i -ésima *query*.

Se um item relevante é o primeiro resultado retornado, então o rank recíproco (RR) é 1. Se o item relevante foi retornado na segunda posição, $RR = 1/2 = 0.5$ e assim por diante. Em outras palavras, MRR é o inverso da posição do primeiro item relevante de uma *query* e uma medida particularmente interessante para sistemas em que somente o primeiro resultado importa (MCFEE; LANCKRIET, 2010), como no SILQ. A métrica não define, porém, o que fazer quando mais de um resultado correto foi retornado. Se nenhum resultado correto foi retornado, entretanto, comumente utiliza-se valor de rank recíproco igual a

0.

2.2.3.4 Conjunto de testes

A forma padrão de realizar medições em sistemas de IR envolve o uso de um *conjunto de testes*. Este conjunto é uma coleção de exemplos contendo *queries* e seus respectivos resultados corretos, geralmente avaliados por um especialista, e usados apenas para realizar a avaliação do sistema. Em sistemas que dependem de um conjunto de treino para alimentar o algoritmo, como os baseados em *machine learning*, o conjunto de testes não é previamente conhecido pelo sistema (RIPLEY; HJORT, 1995).

As *queries* do conjunto de testes são submetidas ao sistema e os resultados retornados comparados com aqueles informados pelo especialista, produzindo uma série de medidas que podem ser utilizadas para a avaliação do sistema (WITTEN; FRANK, 2005).

2.3 ARQUITETURA DE SISTEMAS BASEADOS NA WEB

Nesta Seção são introduzidos conceitos e fundamentos relacionados à construção, interoperabilidade e arquitetura geral de sistemas baseados na Web. A maioria destes conceitos já estão consolidados na indústria de software e são aqui explanados com o objetivo de familiarizar o leitor com os assuntos discutidos no decorrer do desenvolvimento deste trabalho, bem como para apresentar as justificativas tecnológicas que motivaram as alterações arquiteturais realizadas no SILQ.

2.3.1 *Web services* e a arquitetura *REST*

Segundo a W3C (*World Wide Web Consortium*), *web services* são serviços que provém meios de interoperabilidade padronizados entre diferentes aplicações de software rodando em ambientes e/ou plataformas heterogêneas. Um *web service*, portanto, oferece tal serviço de interoperabilidade através da *World Wide Web*, expondo uma interface descrita em um formato legível por máquinas. Diferentes entidades consomem tal serviço através de trocas de mensagem utilizando o formato especificado (BOOTH et al., 2004).

Representational state transfer (REST) é um estilo arquitetural utilizado para a implementação de um *web service*, baseado no proto-

colo HTTP (FIELDING; TAYLOR, 2002). Serviços web que implementam uma API no estilo REST também são chamadas de *RESTful APIs* e comumente utilizam JSON como formato de intercomunicação, podendo utilizar também XML, Atom, texto, entre outros (RICHARDSON, 2013).

Um exemplo de comunicação entre um servidor web rodando um serviço de consulta de uma biblioteca e um cliente (outra aplicação) que desejasse saber a coleção de livros de seu acervo seria o seguinte:

1. O cliente envia um requisição HTTP do tipo GET para a URI `http://api.biblioteca.exemplo.com/acervo`;
2. O servidor da biblioteca responde ao pedido com a coleção de livros de seu acervo em formato JSON. Um exemplo de resposta poderia ser:

Listing 2.1 – Exemplo de uma resposta JSON dada por um serviço RESTful

```
1 {  
2   acervo: [  
3     {  
4       titulo: "Uma breve história do tempo",  
5       autor: "Stephen W. Hawking"  
6     },  
7     {  
8       titulo: "Metodologia de Pesquisa para  
9         Ciência da Computação",  
10      autor: "Raul Sidnei Wazlawick"  
11    },  
12    ...  
13  ]  
14 }
```

3. A resposta em JSON é recebida pelo cliente, interpretada e o conteúdo processado (consumido) da maneira que a aplicação necessitar.

Desta forma, clientes que necessitem da informação de acervo da biblioteca do exemplo acima não precisam se preocupar com os detalhes tecnológicos necessários para realizar essa consulta (se os dados estão em um banco de dados local ou distribuído, relacional ou não, qual linguagem é utilizada para consulta, etc), eles apenas fazem uma

requisição HTTP de um certo tipo e em uma URI pré-estipulada (normalmente definidas na documentação do *web service*) e são servidos com a informação desejada.

Uma interpretação moderna de tal arquitetura é denominada *microserviço*, que compartilha das características de um *web service* clássico, mas cuja maior diferença seria que os *microservices* tem como objetivo prover um serviço específico, desempenhando uma única função minimalista, porém completa (FOWLER; LEWIS, 2014). Deste ponto de vista, um *web service* clássico monolítico⁵ poderia, na maior parte das vezes, ser decomposto em vários *microserviços*, cada um especializado em uma função e podendo ter características tecnológicas completamente distintas.

Outra definição comumente utilizada em aplicações que implementam a arquitetura REST (ou mesmo outras arquiteturas web) é a divisão da aplicação em dois níveis lógicos, o *server-side* e o *client-side*. Dentro do modelo cliente-servidor, o *server-side* de uma aplicação é toda infraestrutura de um servidor que recebe e processa requisições, geralmente retornando uma resposta à este pedido. O *client-side*, por sua vez, é toda infraestrutura que efetua a requisição ao servidor, recebe sua resposta e a manipula para apresentar o resultado ao usuário. No contexto das aplicações web modernas, *server/client-side* são geralmente associados aos termos *back-end* e *front-end* da aplicação, respectivamente, apesar de não possuírem significados completamente idênticos (JIA; ZHOU, 2005).

Tais conceitos estimularam a criação de uma API REST para o SILQ, com o objetivo de tornar disponível, de forma programática, os dados Qualis inseridos em sua base de dados e o serviço de avaliação de currículos. Desta forma, outras aplicações podem consumir os serviços oferecidos pelo sistema sem se preocupar com os detalhes tecnológicos utilizados pelo SILQ para implementar tais funcionalidades, aumentando a facilidade de integração com outros sistemas.

2.3.2 Teste automatizado de software

O processo de desenvolvimento de software é uma atividade em que comumente ocorrem, apesar dos esforços e do emprego de técnicas e ferramentas, diversos erros (*bugs*) no produto final. A Garantia da

⁵Dentro do conceito da arquitetura de *microserviços*, monolítica é uma aplicação construída como uma unidade única e indivisível, em oposição ao conceito composto e distribuído dos *microserviços* (FOWLER; LEWIS, 2014).

Qualidade de Software é uma disciplina que agrega as atividades que ocorrem de forma paralela ao desenvolvimento com o objetivo de minimizar a ocorrência de erros e riscos no produto (BARBOSA; MALDONADO; VINCENZI, 2000).

Três das técnicas utilizadas para a garantia da qualidade de teste são a Verificação, Validação e Teste (BARBOSA; MALDONADO; VINCENZI, 2000). A verificação avalia se o produto está sendo desenvolvido corretamente, enquanto a validação avalia se o produto correto está sendo desenvolvido (BOEHM, 1981). Já o teste, no contexto de software, geralmente implica na execução do próprio produto, considerando-se valores de entrada pertinentes para teste, para verificação de certas funcionalidades do sistema (BARBOSA; MALDONADO; VINCENZI, 2000).

Segundo (PRESSMAN, 2001) e (BOURQUE; FAIRLEY, 2014, cap. 5), diferentes tipos de testes podem ser agrupados em níveis de acordo com o grau de especificidade do módulo ou do requerimento sendo testado. Testes unitários têm como objetivo verificar o funcionamento de unidades indivisíveis e isoladas de um software, garantindo que cada uma delas funcione conforme especificado. Testes de integração, por sua vez, tem por objetivo encontrar falhas na integração entre as unidades de um sistema. Testes de sistema, ou testes *end-to-end* (fim a fim), executam a validação do sistema final como um todo, simulando as ações que um usuário real realizaria ao utilizar a aplicação.

2.4 TRABALHOS CORRELATOS

Nesta Seção são expostos trabalhos e ferramentas que se assemelham ao SILQ quanto ao objetivo do sistema (integração do Lattes e Qualis) e quanto ao uso de análise de *feedback* de relevância para o aumento da exatidão do algoritmo de classificação. A primeira Seção aborda, de forma superficial, alguns trabalhos correlatos já citados por (AGUIAR; COSTA, 2015), mas agora levando em conta as alterações propostas por este trabalho, e alguns novos esforços encontrados na literatura e na *web*. A segunda seção aborda trabalhos que, não necessariamente com objetivos ligados ao Lattes ou ao Qualis, de alguma forma utilizaram *feedback* de relevância para aumento de precisão sistemas de *information retrieval*.

2.4.1 Quanto ao objetivo da ferramenta

Na Seção “Trabalhos correlatos” do trabalho de (AGUIAR; COSTA, 2015), os autores citam 5 ferramentas semelhantes ao SILQ. Não é objetivo desta seção reavaliar estas ferramentas de forma extensiva, porém um novo estudo destes trabalhos foi realizado com a finalidade de verificar quais mudanças houveram, caso tenham ocorrido, e quais suas semelhanças com as alterações propostas por esta nova versão do SILQ.

Nenhum dos trabalhos mencionados sofreu alterações significativas que anulem as conclusões e análises comparativas realizadas por (AGUIAR; COSTA, 2015). No quesito de integração com outros sistemas através de API programática, apenas o ScriptLattes (MENA-CHALCO; CESAR-JR, 2009) oferece uma API de integração com outros sistemas, através de seu conjunto de funções escritas em Python. Apesar disso, trata-se de uma ferramenta de extração e geração de relatórios de currículos Lattes apenas, sem integração com o Qualis.

O trabalho de (CARVALHO, M. G. P. et al, 2009) é o que mais se assemelha ao SILQ em termos de objetivos da ferramenta, que seria a gestão da produção científica para programas de pós-graduação e grupos de pesquisa, inclusive dando a opção de atribuição automática dos estratos Qualis em publicações. Esta funcionalidade, porém, não pôde ser testada já que as URLs informadas para aquisição da ferramenta não são mais válidas e o projeto foi aparentemente descontinuado.

Os demais trabalhos citados ou se resumem à extração de dados Lattes via HTML ou XML (e não tem conexão com o Qualis), ou não possuem API de integração ou forma de comunicação com outros sistemas; outros ainda, não são iniciativas abertas ou são projetos aparentemente descontinuados.

Quanto a novos trabalhos, não citados por (AGUIAR; COSTA, 2015) e referentes à extração/disponibilização dos dados Qualis, os trabalhos de (PINTO, 2012; CESAR.EDU, 2014) tratam-se de páginas web com consultas à periódicos e conferências do WebQualis referentes ao triênio 2010-2012, somente da área de Ciência da Computação, extraídos da página oficial da CAPES. Não há, entretanto, dados de outras áreas de avaliação e tampouco os novos dados Qualis referentes aos anos de 2013 e 2014. Também não há integração com o Lattes, sendo apenas bases de dados contendo os registros Qualis para consulta.

2.4.2 Quanto ao uso de *feedback* de relevância

O trabalho de (TAO et al., 2006) propõe novas estratégias para a busca e seleção de imagens de uma base de dados utilizando *feedback* de relevância. Este problema é conhecido por *content-based image retrieval (CBIR)* e tem o objetivo de extrair e retornar imagens semanticamente relevantes à uma *query*, através de atributos automaticamente extraídos da imagem, como cor, textura e forma. Segundo os autores, porém, sistemas deste tipo geralmente apresentam baixa precisão. Os autores foram capazes de aumentar a performance do sistema utilizando *feedback* de relevância e modificações da técnica de *biased discriminant analysis*, conforme demonstram seus experimentos.

Outro artigo da área de CBIR, (RUI; HUANG; MEHROTRA, 1997), um dos primeiros a unir as disciplinas de processamento de imagens e recuperação de informação, propõe converter vetores de atributos de imagens para *weighted-term vectors*, do modelo vetorial, um dos modelos mais utilizados em IR. Os autores demonstraram que podem atribuir pesos aos termos das *queries* baseados em *feedback* de relevância e assim aumentar a exatidão do sistema.

No domínio de *web search engines*, o trabalho de (SPINK; JANSEN; OZMULTU, 2000) examina a ocorrência de uso de reformulação de *query* e *feedback* de relevância de usuários do buscador “Excite”. Uma forma de *feedback* implícito dado por usuários é reformular sua *query* logo após realizar uma pesquisa mal sucedida, a fim de alterar seus parâmetros de busca. Os autores citam que neste caso os usuários tendem a adicionar ou remover termos de busca. Outra forma de *feedback* dado por usuários neste sistema é o botão “*More like this*”, indicando que deseja ver mais resultados semelhantes ao indicado. Segundo a análise dos autores, poucos usuários reformulam suas *queries* ou utilizam os controles de *feedback*. Aproximadamente 33% dos usuários realizam mais de uma *query* e apenas 5% das buscas realizadas são provenientes de *feedback* de relevância. O padrão mais comum de uso é a realização de uma única busca e a visualização dos 10 primeiros resultados, apenas. As conclusões de (SPINK; JANSEN; OZMULTU, 2000) apontam que é necessária uma interface mais amigável a fim de estimular o usuário a fornecer *feedback*.

O trabalho de (KELLY; BELKIN, 2001) mostra como técnicas de *feedback* implícito podem ser utilizadas para obter informações relevantes e de forma transparente ao usuário. Em especial, utilizaram informações de tempo de leitura, ações de *scroll* de tela e interação com resultados como fonte de *feedback* de relevância.

3 SILQ 2: ALTERAÇÕES ESTRUTURAIS

Como já mencionado, o presente trabalho é uma continuação dos esforços iniciados por (AGUIAR; COSTA, 2015) e que deu origem ao Sistema de Integração Lattes-Qualis (SILQ). Neste capítulo será apresentado o histórico do trabalho original, os motivos e justificativas que levaram ao desenvolvimento de um novo trabalho para sua continuação, comparações do que foi alterado nesta nova versão do sistema e por que estas alterações foram necessárias.

Para mais fácil compreensão e contextualização, durante o texto serão utilizados os termos “trabalho original”, “SILQ 1” ou ainda “primeira versão do sistema” para referir-se ao trabalho de (AGUIAR; COSTA, 2015). Os termos “SILQ 2” e “nova versão do sistema”, em contrapartida, referem-se a este trabalho.

3.1 HISTÓRICO E VISÃO GERAL DO SILQ 1

Este trabalho é uma continuação de (AGUIAR; COSTA, 2015), um Trabalho de Conclusão de Curso de alunos do curso de Ciência da Computação da UFSC, orientados pela Professora Carina F. Dorneles. O objetivo desse trabalho de 2015 era a criação de um sistema que deveria ser capaz de qualificar produções científicas, nas categorias artigos e trabalhos apresentados em eventos, por busca por similaridade de dados com os dados extraídos do WebQualis (AGUIAR; COSTA, 2015, p. 26-27). Este objetivo foi alcançado com a criação da primeira versão do Sistema de Integração Lattes-Qualis (SILQ), lançado no segundo semestre de 2015 e disponível no sítio <http://silq.inf.ufsc.br/>.

Apesar de estável e com sua função principal sendo desempenhada de forma satisfatória, o SILQ 1 deixou algumas lacunas e melhorias a serem desenvolvidas por trabalhos futuros. Segundo os próprios autores, “[...] o SILQ foi concebido para ser uma ferramenta de domínio público e vários projetos devem nascer a partir dele. A continuidade do projeto só tem a acrescentar ao mundo acadêmico [...]” (AGUIAR; COSTA, 2015, p. 79), o que motivou a criação do trabalho para a continuação da proposta original. O ponto de partida para a nova versão do SILQ, portanto, foi a Seção “Trabalhos futuros” do trabalho original.

3.1.1 “Trabalhos futuros” citados pelo SILQ 1

Na Seção de “Conclusões e Trabalhos Futuros” do trabalho original, (AGUIAR; COSTA, 2015) citam 8 itens de melhoria que poderiam ser revisitados por trabalhos futuros. Cada um destes itens está exposto abaixo junto uma explicação de quais foram as atitudes tomadas por este trabalho para resolvê-los.

1. Controle de persistência e transações

definir um novo container de execução para automatizar o uso de uma API de persistência. No momento, as transações são controladas de forma manual; (AGUIAR; COSTA, 2015, p. 79)

Esta melhoria trata-se de um detalhe de implementação do SILQ 1 que, devido à sua arquitetura, forçava a criação e gerenciamento manual de transações ao realizar acessos ao banco de dados. A automatização desta tarefa foi rapidamente alcançada no SILQ 2 com a migração para o *Spring Framework*, conforme relatado na Seção 3.2.2.

2. Atualização dos dados Qualis

as publicações periódicas dos documentos de avaliação do WebQualis levam à necessidade de atualizar o banco de dados periodicamente. A construção de uma funcionalidade automatizada é pertinente; (AGUIAR; COSTA, 2015, p. 79)

Os dados Qualis utilizados pelo sistema para realizar a avaliação de currículos Lattes continuam sendo inseridos de forma manual, já que a disponibilização de novos registros Qualis pela CAPES ocorre atualmente a cada ano, no máximo. Um dos objetivos deste trabalho é a atualização da base de dados para inclusão dos novos registros divulgados no ano de 2016.

Não foi construída uma funcionalidade automática para tal tarefa já que as tabelas fornecidas pela CAPES constantemente possuem erros de formatação que necessitam de ajuste manual e sofreram mudanças no formato de disponibilização nos últimos anos, passando de PDF para CSV. Os dados Qualis 2013 e 2014,

divulgados somente no ano de 2016, portanto, foram inseridos manualmente na base de dados do SILQ 2. O processo é descrito na Seção 3.2.1.

3. *Fine-tuning* da função de similaridade

desenvolvimento de uma nova função de similaridade ou novos procedimentos de cálculo de similaridades. Foram testadas funções já bem conhecidas pela comunidade científica, com testes concretos para levantar qual delas é a mais indicada no escopo da ferramenta. Porém, é possível que se alcance resultados melhores utilizando uma combinação de funções, ou até mesmo criando uma nova especificamente para o cálculo de similaridade de nomes de periódicos e conferências; (AGUIAR; COSTA, 2015, p. 79)

Os testes realizados pelos autores do trabalho original somente levaram em conta o tempo de processamento entre diferentes tipos de algoritmos de similaridade textual, escolhendo a função *trigram* em detrimento das outras por sua integração nativa com o PostgreSQL e por seu tempo de resposta superior aos outros métodos (AGUIAR; COSTA, 2015, p. 57-58). Não foi realizada, porém, uma comparação da precisão dos resultados de diferentes métodos de similaridade.

Um dos objetivos do presente trabalho é a estipulação de métricas de avaliação para algoritmo do SILQ que estabeleça uma média do número de respostas corretas dadas pelo sistema e, assim, realizar modificações no algoritmo e comparar com os resultados anteriores, sempre com o objetivo de maximizar o número de trabalhos corretamente avaliados pelo sistema.

A avaliação do algoritmo do SILQ 1, junto com a comparação do novo algoritmo desta nova versão do sistema que leva em conta *feedback* de usuário, são descritos no Capítulo 4.

4. Considerar outras informações do pesquisador

levantamento das outras informações contidas no currículo, como o desempenho do pesquisador antes de ser incluído em um grupo de pós-graduação ou a quantidade e

a qualidade de outras atividade acadêmicas;
(AGUIAR; COSTA, 2015, p. 79)

Este item envolve o processamento de outras informações contidas no currículo Lattes de um pesquisador a fim de credenciá-lo em um Programa de Pós-Graduação, utilizando as publicações e informações de onde foram publicados/apresentados os trabalhos e artigos do pesquisador e utilizá-los para sua avaliação automática. Este item não foi abordado neste trabalho.

5. Relação entre orientações e produções associadas

levantamento da relação entre as atividades de orientação, como bolsas de iniciação científica, e as produções acadêmicas associadas; (AGUIAR; COSTA, 2015, p. 80)

Não foi abordado neste trabalho, pelos mesmos motivos do item anterior e por estar sendo tratado por outro TCC.

6. Busca por informações de eventos

expansão do módulo da busca por informações de eventos. A complexidade do tema impediu o desenvolvimento completo desse módulo na ferramenta SILQ. Procedimentos automáticos de busca utilizando-se web crawlers e o tratamento correto dessas informações fornecem muitos temas para desenvolvimento de módulos novos; (AGUIAR; COSTA, 2015, p. 80)

A inserção de dados do Qualis Conferência¹ no sistema se dá de forma manual, à medida em que novas informações são disponibilizadas pela CAPES. O objetivo deste item seria a criação de um *web crawler* para captura automática de dados de eventos científicos na web. Devido à complexidade deste tema e por se tratar de um assunto não relacionado com o foco deste trabalho, este item não foi abordado.

¹Dados e conceitos relacionados a eventos científicos disponibilizados pela CAPES para algumas áreas específicas

7. Feedback do usuário

permitir que o usuário auxilie a ferramenta na qualificação, ou seja, no caso dele observar algum matching errado, ele próprio pode sugerir o matching correto; (AGUIAR; COSTA, 2015, p. 80)

Esta é a principal contribuição deste trabalho e que deu origem à hipótese de pesquisa. É possível aumentar a precisão da ferramenta de qualificação, de uma maneira geral, utilizando sugestões de usuários? Para responder a esta pergunta foi necessário estipular métricas para calcular o grau de precisão atual da ferramenta e então comparar o algoritmo antigo com um novo que levasse em consideração estas sugestões. Dois novos algoritmos foram propostos e relatados na Seção 4.2. As avaliações experimentais de ambos são descritos na Seção 4.3.

8. Período de participação em grupo de pesquisa

indicar o período em que o professor está vinculado a um programa de pós-graduação, permitindo que as avaliações dos currículos também possam ser realizadas considerando-se apenas os períodos nos quais o pesquisador estava vinculado a esse programa. (AGUIAR; COSTA, 2015, p. 80)

Este item trata de um detalhe de gerenciamento de grupos e de sua avaliação, que considera publicações de um pesquisador mesmo quando este foi membro do grupo em um período que não cobre totalmente o período de avaliação estipulado. Para correção deste comportamento, basta dar ao administrador do grupo a opção de informar o período de ingresso ou de participação de um pesquisador no grupo de pesquisa e atualizar o algoritmo de avaliação para considerar somente publicações neste período. No entanto, esta atualização não foi realizada neste trabalho.

3.1.2 Como o SILQ realiza o *matching* Qualis-Lattes

O *matching* da produção científica de um currículo Lattes com o Qualis é realizado pelo SILQ com base na similaridade textual entre o título do evento de cada trabalho presente no currículo e o título

do evento Qualis². O título do veículo onde o trabalho foi publicado, juntamente com seu ano e área de avaliação cadastrados no Lattes são dados como *query* para o sistema. O sistema busca sobre todo o conjunto de dados Qualis presentes na base de dados aquele com maior similaridade textual em relação à *query*. Se esta similaridade for maior do que um *threshold* pré-estabelecido, então o resultado é considerado um *match* válido e o estrato atribuído pelo Qualis a este evento é também atribuído ao trabalho do pesquisador.

Pode-se tomar o exemplo real de um trabalho qualquer extraído de um currículo Lattes cujo evento tenha sido especificado como “2016 IEEE 7th Latin American Symposium on Circuits & Systems (LASCAS)”. A Tabela 1 mostra os resultados retornados pelo SILQ junto com seus respectivos valores de similaridade textual.

Tabela 1 – Resultados retornados pelo SILQ para a *query* “2016 IEEE 7th Latin American Symposium on Circuits & Systems (LASCAS)”

| Evento | Estrato | Similaridade |
|---|----------------|---------------------|
| IEEE Latin American Symposium on Circuits and Systems (LASCAS) | B5 | 0.87 |
| IEEE International Symposium on Circuits and Systems (ISCAS) | A1 | 0.48 |
| IEEE Latin American Robotics Symposium (LARS) | B4 | 0.45 |
| IEEE International Midwest Symposium on Circuits and Systems (MWSCAS) | B1 | 0.43 |
| Symposium on Asynchronous Circuits and Systems (ASYNC) | A2 | 0.40 |

O primeiro resultado retornado é o escolhido, por possuir a maior similaridade em relação à *query*. Neste caso, o trabalho é avaliado com o conceito B5, já que este é conceito Qualis atribuído ao evento.

O algoritmo de avaliação da primeira versão do SILQ retorna os resultados da Tabela 1 ao ser configurado para utilizar um “nível de confiança” de 40%. Esse nível de confiança é o *threshold* utilizado pelo algoritmo de classificação. Qualquer resultado cujo nível de similaridade em relação à *query* seja inferior ao nível de confiança utilizado

²O *matching* por similaridade é utilizado somente para eventos. Para periódicos, o *matching* é realizado através de comparação entre o ISSN informado no Lattes e nos registros Qualis.

não é retornado.

Figura 2 – Diálogo de configurações de avaliação

Escolha as configurações de avaliação

Área de Atuação *

Ciência da Computação

As avaliações serão realizadas de acordo com a área selecionada.

Publicações de

2012

Até

2016

Nível de confiança

Normal (60%)

Estabelece um limite mínimo da qualidade da avaliação.

Iniciar Avaliação

O nível de confiança pode ser ajustado através das opções de avaliação, apresentadas quando o usuário requisita uma avaliação de currículo Lattes (Figura 2). Diminuir o nível de confiança (e em consequência o *threshold* do algoritmo de classificação) implica em obter mais resultados e classificar mais trabalhos, porém diminuir a precisão do algoritmo, já que resultados não relevantes serão retornados para as *queries* que não obtiveram bons resultados (resultados com nível de similaridade alto). A Seção 4.3 apresenta testes de validação do algoritmo que indicam o nível de precisão obtidos para cada valor de *threshold* utilizado, além de sugerir um nível de confiança ideal a ser utilizado para maximizar o número de trabalhos corretamente avaliados pelo sistema.

3.2 SILQ 2

3.2.1 Extração e inserção dos novos dados Qualis

Os dados Qualis de eventos e periódicos foram extraídos pelo trabalho original a partir de documentos em PDF e planilhas. Como relatado em (AGUIAR; COSTA, 2015, p. 50-52), os dados foram extraídos dos PDFs utilizando bibliotecas para manipulação desse formato e passaram por um processo de limpeza. A partir deste processo, aproximadamente 107 mil tuplas foram criadas representando os periódicos do triênio 2010-2012.

No primeiro semestre do ano de 2016, porém, a CAPES disponibilizou novos dados Qualis referentes ao ano de 2013 e 2014, alterando sua periodicidade de avaliação de trimestral para anual. Estes novos dados foram divulgados em formato CSV, junto com os antigos dados do triênio 2010-2012, mas desta vez separados por ano. Estes dados foram obtidos pela plataforma Sucupira (antigo WebQualis): <http://qualis.capes.gov.br/webqualis/>.

Como o conjunto de dados do SILQ 1 contava apenas com dados trimestrais, as tuplas originais não possuíam informação de ano (só podia-se deduzir que os dados faziam parte do triênio, mas não de qual ano). Desta forma, todas as planilhas divulgadas passaram por um novo processo de extração, incluindo os dados já inseridos no SILQ 1, mas agora considerando a nova informação de ano.

As cinco novas planilhas passaram por um processo de *data cleaning* manual:

1. Substituição de entidades HTML especiais por seu caractere correspondente. Ex.: “&” por “&”;
2. Normalização do campo ISSN para o formato “9999-9999”: alguns registros não possuíam o dígito separador ou omitiam os zeros à esquerda;
3. Correção de ISSNs errôneos: alguns registros possuíam o campo ISSN com dígitos faltando. Ex.: 0034-167 (Revista Brasileira de Enfermagem), cujo número correto seria 0034-7167. Estes casos foram tratados um a um e os números corretos identificados através de pesquisas na web.

As planilhas CSV foram então dadas como *input* à base de dados resultando na criação de 339.204 tuplas, referentes a cada um dos

anos do período divulgado 2012-2015. A Tabela 2 mostra o número de periódicos extraídos para cada ano de avaliação do Qualis.

Tabela 2 – Número de periódicos extraídos dos dados Qualis

| Ano | Nº de periódicos extraídos |
|--------------|----------------------------|
| 2010 | 75.786 |
| 2011 | 66.171 |
| 2012 | 108.272 |
| 2013 | 44.437 |
| 2014 | 44.538 |
| Total | 339.204 |

Após esta etapa, verificou-se que todos os registros da base de dados antiga do SILQ 1 estavam contidos nesta nova versão. Desta forma, os dados do SILQ 1 foram mantidos, porém agora incluindo o ano de avaliação, além do nome do periódico, ISSN, estrato Qualis atribuído pela CAPES e área de avaliação.

A inclusão dos novos registros aumentou o número de publicações passíveis de serem avaliadas pelo sistema. Os dados Qualis de 2015 e 2016 ainda não foram divulgados no momento de escrita deste trabalho, mas podem ser incluídos na base de dados, assim que forem disponibilizados pela CAPES, passando por um processo semelhante ao descrito nesta Seção.

3.2.2 Criação do *Web Service*

A primeira versão do SILQ foi desenvolvida com base no *Play*³ *Framework*, um *web application framework* escrito em Java e Scala que simplifica o processo de construção de uma aplicação web. Este mesmo framework provê APIs e bibliotecas para o desenvolvimento tanto do *back-end* quando do *front-end* de uma aplicação web.

Para esta nova versão do SILQ, porém, um dos objetivos seria a criação de uma API de integração programática, tornando o SILQ não só um sistema para uso de usuários finais, mas também para integração com outras ferramentas, como um *web service*. Para tanto, uma alteração significativa na arquitetura da aplicação foi feita, separando o sistema em um *server-side* rodando Spring⁴ e servindo seu conteúdo

³<https://www.playframework.com/>

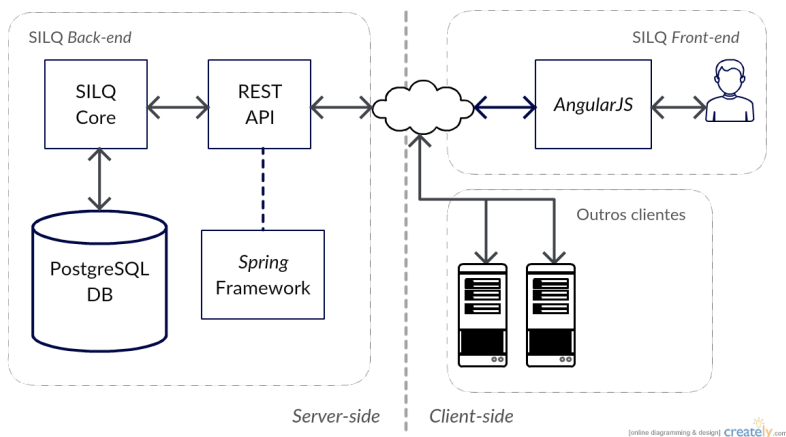
⁴<http://projects.spring.io/spring-framework/>

através de uma API REST; e um *client-side* remodelado utilizando AngularJS⁵ e consumindo o serviço através de requisições HTTP.

Em um primeiro momento, o “núcleo” do SILQ, que inclui os algoritmos de avaliação e processamento dos currículos, não foi alterado, apenas a forma com que este serviço é disponibilizado e consumido. Desta forma, a linguagem utilizada no sistema, o Java, foi mantida. A escolha da alteração do framework back-end, de *Play!* para *Spring*, se deu ao fato da mudança no paradigma arquitetural da aplicação, com a introdução da camada REST. O Spring se trata de um framework Java *open-source* largamente utilizado pela comunidade que oferece suporte a vários aspectos de uma aplicação web, incluindo a construção de interfaces RESTful.

A Figura 3 mostra um esquema da nova arquitetura do SILQ. A caixa “SILQ Core” representa a camada de serviço da aplicação, responsável pelo acesso ao banco de dados, avaliação de currículos e gerenciamento dos dados Qualis, cuja implementação foi realizada no trabalho passado (apesar de também sofrer algumas mudanças tecnológicas no trabalho atual).

Figura 3 – Nova arquitetura do SILQ



⁵<https://angularjs.org/>

3.2.3 Alterações no *front-end*

Na primeira versão do SILQ, as páginas HTML do sistema eram dinamicamente renderizadas no server-side pelo *Play!* Framework e servidas conforme as requisições dos clientes. Desta forma, além de executar toda a lógica de aplicação, o servidor também fica responsável pela lógica de apresentação do conteúdo.

Em arquiteturas REST, a lógica de apresentação de dados (ou visão), fica completamente separada do servidor, executando no *web browser* do cliente. O servidor só recebe requisições HTTP, as processa e retorna uma resposta em formato JSON. Esta resposta é recebida pelo cliente que então apresenta o conteúdo através da interface gráfica HTML renderizada pelo browser. Desta forma, a carga no servidor é diminuída, já que ele não está mais incumbido da tarefa de renderização das páginas HTML que serão servidas.

Utilizando esta ideia, no SILQ 2, a camada de visão foi totalmente reescrita em *Javascript*, utilizando o framework *AngularJS*. Trata-se de um framework *open-source* desenvolvido pela Google que auxilia na criação de interfaces web dinâmicas e integração com *web-services* via REST.

Desta forma, quando um cliente realiza a primeira requisição à URL do SILQ, o servidor serve uma página HTML (geralmente denominada de *index*) junto com os *assets* necessários para a construção de conteúdo dinâmico e estilização da página (imagens, arquivos CSS e Javascript). As requisições subsequentes, entretanto, utilizam somente requisições assíncronas ao servidor, utilizando Javascript: uma requisição conhecida por *Ajax* (Asynchronous JavaScript and XML). Estas requisições utilizam a camada REST do servidor que retornará respostas em JSON e cujo conteúdo será processado no cliente e montará as páginas de forma dinâmica. Desta forma, parte da lógica da aplicação como um todo é transferida para o cliente, reduzindo a carga no servidor.

Outra mudança significativa realizada no *front-end* da aplicação foi a atualização do *framework* CSS utilizado, inserida no processo de reescrita das páginas HTML. A primeira versão do sistema utilizava a versão 2 do *framework open-source* Bootstrap⁶, que foi atualizada para a versão 3 no SILQ 2. De forma paralela foram realizadas algumas mudanças na interface do sistema, principalmente na página de resultados de avaliação que passou a incluir controles permitindo ao usuário in-

⁶<http://getbootstrap.com/>

Figura 4 – Página de resultados de avaliação do SILQ 1

| Título do trabalho | Ano de publicação | Nome do evento | Conceito |
|---|-------------------|--|------------|
| A Filtered-Page Ranking: An Approach for Previously Filtered HTML Documents Ranking | 2016 | International Conference on Internet and Web Applications and Services | B3 (1.0) ✓ |
| Towards Automatic Document Classification by Exploiting only Knowledge Resources | 2015 | International Conference of the Chilean Computer Science Society | B3 (1.0) ✓ |
| Implementação de um esquema de extração de dados tabulares da web | 2015 | XII Workshop de Trabalhos de Iniciação Científica (WTIC) | ✗ |
| Nazca: A Context-based Matching Method for Searching Heterogeneous Structures | 2014 | International Conference on Computer and Information Technology | B1 (1.0) ✓ |
| Uso de Expressões Temporais em Busca na Web: Uma análise através das sugestões de consulta. | 2014 | Escola Regional de Banco de Dados | ✗ |
| SSUP - A URL-based method to entity-page discovery | 2014 | International Conference on Web Engineering | B1 (1.0) ✓ |
| Descoberta de Domínio Conceitual de Páginas Web | 2014 | Workshop de Teses e Dissertações em Banco de Dados | ✗ |
| Proposta de um Framework para Visualização de Dados Agregados por Similaridade para Auxiliar Consultas durante a Navegação na Web | 2013 | Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais | B4 (1.0) ✓ |
| Detectando similaridade entre diferentes representações de entidades usando Classificadores Bayesianos | 2012 | Escola Regional de Banco de Dados | ✗ |
| Pré-processamento de tabelas Web heterogêneas para execução do algoritmo de junção merge | 2012 | Escola Regional de Banco de Dados | ✗ |
| Usando informações do contexto para melhoria da precisão nas buscas por similaridade | 2012 | Escola Regional de Banco de Dados | ✗ |

Figura 5 – Página de resultados de avaliação do SILQ 2

Nome
Carina Friedrich Dorneles

Área do conhecimento
Ciência da Computação

Especialidade
Banco de Dados

Área utilizada na avaliação
Ciência da Computação

Período de avaliação
Todos

Totalizador

Nível de confiança: Normal

| Conceito | Qtde. |
|----------|-------|
| A1 | 4 |
| A2 | 2 |
| B1 | 5 |
| B2 | 3 |
| B3 | 6 |
| B4 | 1 |
| B5 | 4 |
| C | 1 |

Trabalhos 60 registros

Download .csv

2016 | A Filtered-Page Ranking: An Approach for Previously Filtered HTML Documents Ranking

International Conference on Internet and Web Applications and Services

B3

100%

2012

ICIW

International Conference on Internet and Web Applications and Services

2015 | Towards Automatic Document Classification by Exploiting only Knowledge Resources

International Conference of the Chilean Computer Science Society

B3

75%

2012

ISWSA

International Conference on Intelligent Semantic Web-Services and Applications

A1

70%

2012

MobileSys

International Conference on Mobile Systems, Applications, and Services

A1

65%

2012

ICWS

IEEE International Conference on Web Services

B3

63%

2012

ICNS

International conference on Networking and Services

Ver menos resultados

2015 | Implementação de um esquema de extração de dados tabulares da web

XII Workshop de Trabalhos de Iniciação Científica (WTIC)

Nenhum conceito encontrado | Sugerir matching

2014 | Nazca: A Context-based Matching Method for Searching Heterogeneous Structures

International Conference on Computer and Information Technology

B1

100%

2012

CIT

International Conference on Computer and Information Technology

Ver mais resultados

2014 | Uso de Expressões Temporais em Busca na Web: Uma análise através das sugestões de consulta.

Escola Regional de Banco de Dados

Nenhum conceito encontrado | Sugerir matching

formar *feedback* de relevância sobre os resultados retornados. A Figura 4 mostra um exemplo da página de resultados de uma avaliação realizada pelo SILQ 1. A página foi remodelada para permitir a inclusão de controles de *feedback* e para apresentar múltiplos resultados candidatos para um mesmo artigo ou trabalho, conforme mostra a Figura 5.

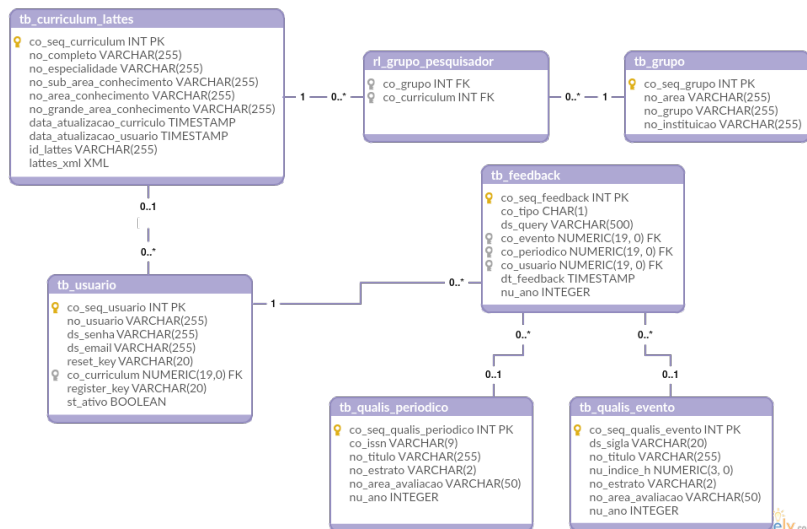
3.2.4 Alterações no modelo lógico

O banco de dados relacional implementado em *PostgreSQL* também sofreu algumas modificações em relação ao trabalho anterior. A Figura 6 mostra o esquema lógico do banco de dados da versão atual do SILQ.

Algumas mudanças pequenas de nomenclatura foram realizadas. Uma mudança significativa, porém, foi a unificação de duas tabelas, *tb_dado_geral* e *tb_profissional*, para uma única tabela, *tb_curriculum_lattes*. A função destas duas tabelas antigas era guardar os currículos enviados de usuários e de pesquisadores de grupos, respectivamente. Nesta nova versão do SILQ, porém, em ambos os casos os currículos Lattes enviados pelo sistema são salvos na mesma tabela de currículos (*tb_curriculum_lattes*), inclusive tendo seus registros reutilizados em caso de, por exemplo, dois usuários enviarem o mesmo currículo. Neste caso, o sistema armazena somente uma vez o currículo e cria duas referências diferentes a ele, poupando-o do trabalho de duplicar os dados do currículo Lattes, cuja representação em XML contém em média algumas dezenas de *kilobytes*.

Outra alteração foi a adição da tabela *tb_feedback* para registro dos *feedbacks* de relevância informados pelos usuários, conforme relata a Seção 4.1.

Figura 6 – Novo esquema do modelo lógico do SILQ



3.2.5 Garantia da qualidade

Para a garantia da qualidade desta nova versão do SILQ, foram introduzidas duas camadas de testes automatizados: a primeira, envolvendo *testes unitários* e *testes de integração* escritos em Java e com o objetivo de garantir a corretude do server-side da aplicação; e a segunda, *testes de sistema* escritos em Javascript e simulando casos de uso reais, garantindo a corretude do client-side e de sua integração com o servidor.

No SILQ, os testes unitários e de integração foram escritos em Java utilizando o framework de testes *jUnit*⁷. Eles testam cada função da camada de serviço da aplicação, por exemplo simulando o upload de um currículo qualquer e verificando se os dados extraídos e retornados são de fato aqueles contidos no currículo. Foram criados testes, inclusive, para as funcionalidades já existentes desde o SILQ 1, mas que ainda não eram cobertos com casos de teste automatizados, para assim aumentar a confiabilidade do sistema e garantir que mudanças futuras não ocasionem *bugs* nestes módulos antigos.

Este nível de teste, porém, não valida a interface de usuário e sua integração com a camada de serviço. Todo o client-side da aplicação ainda estaria “descoberto” de casos de testes. Para sanar este problema, foram incluídos no SILQ casos de teste de sistema escritos em Javascript e utilizando o framework de testes *Protractor*⁸, criado especificamente para testes *end-to-end* de aplicações feitas com AngularJS. O Protractor simula as ações de um usuário real realizando cliques em botões, preenchendo formulários e navegando através de links da aplicação. Desta forma, são garantidos algum nível de corretude da interface do sistema e da integração entre o *client* e *server-side*.

⁷<http://junit.org/>

⁸<http://www.protractortest.org/>

4 USO DE *FEEDBACK* DE RELEVÂNCIA

Como já mencionado, um dos itens de melhoria indicados por (AGUIAR; COSTA, 2015) após a realização da primeira versão do sistema seria “permitir que o usuário auxilie a ferramenta na qualificação”. Em sistemas de IR, esta característica é denominada *Feedback de Relevância*.

A ideia por trás desta técnica é permitir que o usuário julgue resultados iniciais retornados pelo sistema, classificando-os como relevantes ou não para a *query* vigente. O sistema então é capaz de utilizar esta informação para melhorar seu algoritmo de classificação e retornar melhores resultados para novas *queries*.

O capítulo atual apresenta como esta técnica foi aplicada no SILQ, qual modelo lógico foi utilizado para o armazenamento dos *feedbacks* e qual técnica utilizada para sua obtenção. Por fim, são apresentadas medições realizadas no sistema que validam se a técnica foi relevante para o aumento da taxa de acerto do sistema.

4.1 OBTENÇÃO DE *FEEDBACK*

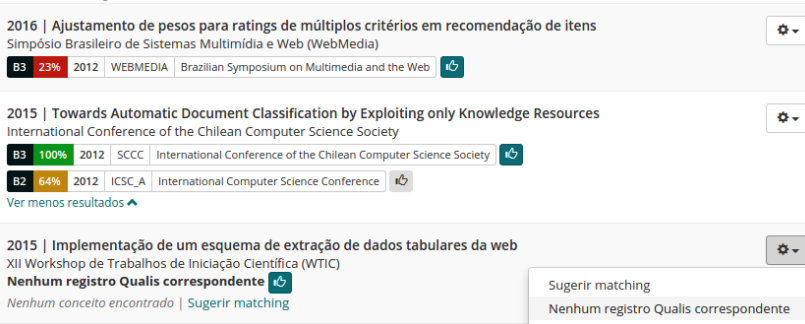
Em sistemas de IR que implementam *feedback* de relevância, tipicamente é permitido ao usuário julgar como relevante ou não relevante cada item retornado pelo sistema. Isso é natural pois o objetivo deste tipo de sistema é retornar a totalidade do conjunto de itens relevantes, cujo tamanho é variável dependendo da *query*. O SILQ, porém, é um caso específico em que existem somente 0 ou 1 item relevante para toda *query*. Isso acontece pois cada trabalho indicado no currículo Lattes de um pesquisador aparece qualificado no Qualis apenas uma única vez no ano em que o pesquisador o publicou. O objetivo do algoritmo de avaliação do SILQ é deduzir que registro Qualis é esse, caso exista na base de dados.

Para a implementação de *feedback* de relevância no SILQ, portanto, não é necessário o julgamento de cada item retornado pelo sistema em uma avaliação, mas apenas marcar *qual* dos itens da base de dados Qualis é um *match* correto para a *query* vigente. Desta forma, o usuário deve ser capaz de indicar o registro Qualis que deve ser considerado para cada trabalho avaliado. Também existe o caso especial em que não existe um registro Qualis correspondente ao trabalho, neste caso o usuário deve ser capaz de indicar que não existem *matches* cor-

retos para o trabalho.

A Figura 7 mostra o exemplo de três trabalhos extraídos de um currículo Lattes e avaliados pelo SILQ que receberam *feedback* do usuário. O botão de “joinha” é utilizado para marcar o resultado que o usuário considera relevante para cada trabalho. Os botões grifados (com fundo azul) representam que o resultado já foi previamente marcado. Neste caso, o registro Qualis marcado como relevante é associado à *query* atual, que engloba o título, ano e área do trabalho avaliado, juntamente com o usuário que está realizando o julgamento. Também é possível marcar a opção “Nenhum registro Qualis correspondente” ou sugerir algum resultado não retornado previamente pelo sistema em “Sugerir matching” (mostrado no canto inferior direito da Figura exemplo).

Figura 7 – Detalhe dos controles de feedback de relevância na página de avaliação de currículo Lattes



Os *feedbacks* de relevância dados pelos usuários são salvos na base de dados SILQ na tabela *tb_feedback* (Figura 6) para uso posterior pelo algoritmo de classificação, conforme descrito na Seção 4.2. Já que um trabalho qualquer pode ter no máximo um registro Qualis associado, é salvo somente um *feedback* por *query* por usuário. Desta forma, uma chave única é utilizada na tabela de *feedbacks*, dada pela dupla (título do trabalho, usuário).

4.2 ALGORITMOS DE AVALIAÇÃO COM FEEDBACK

Uma vez registrados os *feedbacks* dos usuários, passa-se a questionar de que forma utilizá-los para aumentar a taxa de acerto do sistema

para futuras consultas. A Seção atual apresenta dois novos algoritmos que foram propostos e avaliados neste trabalho, e como foram implementados, enquanto a próxima Seção apresenta os resultados e comparações de exatidão dos mesmos.

4.2.1 Algoritmo $\text{fb}(\tau)$

Um das abordagens mais simples que podem ser usadas neste caso é utilizar o resultado marcado pelo usuário sempre que uma *query* idêntica ao do *feedback* seja submetida ao sistema. O algoritmo $\text{fb}(1.0)$, portanto, foi desenvolvido com base nesta ideia. O valor “1.0” presente no nome do algoritmo apenas indica que o *feedback* é considerado em detrimento de qualquer outro resultado dado pelo sistema quando a *query* submetida for 100% similar (ou seja, idêntica) à *query* do *feedback*.

Pode-se dar o exemplo real de um nome de evento extraído de um currículo Lattes cadastrado pelo pesquisador como “Software Engineering Knowledge Engineering”, no ano de 2009 e com área de avaliação Ciência da Computação. Ao avaliar tal trabalho, o sistema retorna a lista da Tabela 3.

Tabela 3 – Resultados retornados pelo SILQ para a *query* “Software Engineering Knowledge Engineering”

| # | Evento | Similaridade |
|-----|---|--------------|
| 1 | Software Engineering and Data Engineering (SEDE) | 0.53 |
| 2 | International Conference on Software Engineering and Knowledge Engineering (SEKE) | 0.49 |
| 3 | Software Engineering and Applications (SEA_A) | 0.45 |
| ... | ... | ... |

Após analisar esta lista, o usuário submeteu um *feedback* ao sistema marcando o resultado #2 como o correto. Desta forma, utilizando o algoritmo $\text{fb}(1.0)$, toda *query* subsequente idêntica a “Software Engineering Knowledge Engineering” terá o resultado #2 retornado na primeira posição.

O $\text{fb}(1.0)$ considera apenas *feedbacks* que sejam idênticos à

query submetida, *queries* similares não são consideradas. O usuário do exemplo anterior possui um outro trabalho cadastrado em seu currículo Lattes cujo título de evento é “Software Engineering **and** Knowledge Engineering”. Pode-se deduzir que o usuário quis se referir ao mesmo evento, porém o título não é idêntico ao exemplo anterior por causa do termo “*and*”. Neste caso, o algoritmo **fb**(1.0) não é capaz de deduzir que os dois casos se referem ao mesmo evento, apesar da semelhança entre eles. Uma modificação que pode ser realizada no algoritmo é utilizar uma função de similaridade entre novas *queries* submetidas ao sistema com aquelas anteriormente submetidas e que possuem *feedback* do usuário. Se a similaridade entre a nova *query* e algum dos *feedbacks* for maior do que certo *threshold de feedback*, então é provável que a nova *query* se refira ao mesmo evento do *feedback* anteriormente fornecido.

O algoritmo **fb**(*t*) (para $0.0 \leq t \leq 1.0$) é uma generalização de **fb**(1.0) que considera *feedbacks* cuja similaridade textual em relação à *query* seja maior que o *threshold t*. Por exemplo, *fb*(0.75) irá considerar *feedbacks* cujo valor de similaridade textual em relação à *query* seja 0.75 ou superior. No exemplo anterior, ao submeter a nova *query* “Software Engineering and Knowledge Engineering” ao sistema, o algoritmo *fb*(0.75) calcula a similaridade entre ela e os *feedbacks* anteriores fornecidos pelo usuário e encontra o *feedback* da primeira *query* “Software Engineering Knowledge Engineering” por ser 88% similar à *query* atual. Neste caso, por ter uma similaridade maior do que o *threshold* de 0.75 estipulado, o algoritmo retorna o mesmo evento marcado no *feedback* para a *query* atual (o evento #2 da Tabela 3).

O algoritmo **fb**(*t*), entretanto, leva a outros questionamentos, já que utiliza a mesma técnica de *data-matching* que foi proposta a melhorar. Qual o valor de *t* (*threshold de feedback*) ideal? Qual o algoritmo de similaridade textual ideal para este caso? A Seção 4.3.4 apresenta testes de validação do algoritmo **fb**(*t*) para diferentes valores de *t*.

O Algoritmo 1 é a representação em pseudocódigo de **fb**(*t*). O parâmetro *q* representa a *query*, *t* é o valor de *threshold de feedback*, *D* é o conjunto de todos os documentos a serem pesquisados e *F* o conjunto de *feedbacks* fornecidos contendo as duplas (*q_f*, *d*), *query* do *feedback* e documento dado como *feedback*, respectivamente, tal que *d* ∈ *D*. A variável *m* é o registro provindo de *feedback* com maior probabilidade de ser um *match* correto para a *query*, caso exista. A saída *R* é uma lista 0-indexada contendo os resultados da consulta, ordenada por ordem decrescente da probabilidade do resultado ser um *match* correto para a *query*. A função **trigram_sim** calcula a similaridade textual entre duas *strings* utilizando o método *trigrams* e retornando um valor no

intervalo $[0, 1]$. A função **trigram_rank** é o algoritmo de avaliação da primeira versão do SILQ, que cria o rank R de similaridade a partir da comparação entre q e cada um dos documentos de D . A função **insert_rank_top** insere um registro no topo do *rank*, removendo itens duplicados previamente inseridos.

Algoritmo 1: **fb(t)**

Input : q, t, D, F
Output: R

```

1  $R \leftarrow \text{trigram\_rank}(q, D)$ 
2  $s_m \leftarrow -1$ 
3 for  $(q_f, d) \in F$  do
4    $s \leftarrow \text{trigram\_sim}(q, q_f)$ 
5   if  $s \geq t$  and  $s \geq s_m$  then
6      $m \leftarrow d$ 
7      $s_m \leftarrow s$ 
8   end if
9 end for
10 if  $m$  then
11    $\text{insert\_rank\_top}(R, m)$ 
12 end if
13 return  $R$ 
```

4.2.2 Algoritmo **query_aliasing**

Uma adaptação de **fb(t)** que mostrou-se de mais fácil implementação e que não gera o questionamento de qual valor de t utilizar, foi considerar as *queries* de *feedbacks* anteriormente fornecidos pelo usuário, da mesma forma que o **fb(t)**, porém inseri-las no *rank* de resultados de novas *queries* submetidas com base em seus valores de similaridade textual em relação à nova *query*, junto com os resultados previamente selecionados. Assim, ao invés de escolhê-lo em detrimento dos demais, o evento marcado com *feedback* só é retornado se for mais bem ranqueado que os demais resultados.

Considerando os mesmos exemplos dados na Seção anterior, em que o usuário submete a nova *query* “Software Engineering and Knowledge Engineering” ao sistema, o algoritmo de **query_aliasing** realiza comparação textual entre a nova *query* e as *queries* anteriores que

possuam *feedback*, da mesma forma que o $fb(t)$, encontrando a *query* “Software Engineering Knowledge Engineering”, com um valor de similaridade de 0.88. Ao contrário do $fb(t)$, o algoritmo de `query_aliasing` irá inserir o evento dado como *feedback* a esta *query* junto com a lista de resultados previamente encontrados apenas via similaridade textual, usando o valor de 0.88 para posicionamento no *ranking*. A Tabela 4 mostra o *ranking* retornado para este exemplo. Nota-se que o evento #2, marcado pelo usuário como correto, foi elevado no *ranking* por receber o novo valor de similaridade da comparação com o *feedback*.

Tabela 4 – Resultados retornados pelo SILQ para a *query* “Software Engineering and Knowledge Engineering” utilizando `query_aliasing`

| # | Evento | Similaridade |
|-----|---|--------------|
| 2 | International Conference on Software Engineering and Knowledge Engineering (SEKE) | 0.88 |
| 1 | Software Engineering and Data Engineering (SEDE) | 0.62 |
| 3 | Software Engineering and Applications (SEA_A) | 0.53 |
| ... | ... | ... |

O valor de similaridade atribuído, porém, perde seu significado semântico pois não é mais a similaridade textual entre o título do evento do Lattes e do Qualis calculado através do *trigrams*, mas um valor adimensional usado apenas para ordenação relativa dentro do *ranking*.

Desta forma, ao processar uma *query* q qualquer, o sistema processa um *rank* de resultados primários com base no algoritmo *trigrams* inicial. O *rank* é ordenado do resultado mais similar à q ao menos similar. Após esta etapa, ele também compara q com cada uma das *queries* anteriormente submetidas pelo usuário e que possuem *feedback* de relevância utilizando o mesmo algoritmo de similaridade textual. O resultado mais similar é inserido no *ranking* de resultados. Assim, se q é idêntico a um *feedback* já submetido pelo usuário, o evento deste *feedback* será retornado e inserido no topo do *ranking* de resultados, por ser 100% similar à *query*. Outros resultados similares, porém não idênticos, serão inseridos no *ranking* conforme seu valor de similaridade e só serão escolhidos em detrimento de outros resultados primários se seus valores de similaridade forem superiores a eles.

É como se, ao dar um *feedback* de relevância qualquer, o usuário

criasse um *alias* (um apelido) ao resultado que está sugerindo. Assim, o sistema deve avaliar novas *queries* não só comparando-as com o nome real dos documentos, mas também com os apelidos dados a eles pelo usuário. Por este motivo o algoritmo foi chamado de **query_aliasing**. A avaliação deste algoritmo foi realizada e comparada com os demais na Seção 4.3.4.

O Algoritmo 2 é a representação em pseudocódigo do método proposto. O significado semântico das variáveis é equivalente ao do Algoritmo 1. A saída deste algoritmo é o próprio *rank* R , possivelmente contendo novos resultados devido à comparação com os *feedbacks* de F . O método **insert_rank** da linha 4 insere o registro d na lista ordenada R com um valor de *rank* s , preservando a ordenação da lista, de forma que o elemento em $R[0]$ seja aquele com maior valor de *rank* e, assim, o registro com maior probabilidade de ser um *match* correto para a *query*. A função também elimina itens duplicados, preservando aquele com maior valor de *rank*.

Algoritmo 2: query_aliasing

Input : q, D, F

Output: R

```

1  $R \leftarrow \text{trigram\_rank}(q, D)$ 
2 for  $(q_f, d) \in F$  do
3    $s \leftarrow \text{trigram\_sim}(q, q_f)$ 
4   insert_rank( $R, s, d$ )
5 end for
6 return  $R$ 
```

4.3 AVALIAÇÃO EXPERIMENTAL

Esta Seção apresenta os procedimentos realizados para avaliar as alterações promovidas no algoritmo de avaliação da nova versão do SILQ e se elas contribuíram para o aumento da taxa de acerto do sistema.

4.3.1 Conjunto de testes

O conjunto de testes utilizado para a avaliação do sistema foi criado a partir dos currículos Lattes de 33 pesquisadores do programa

de pós-graduação em Ciência da Computação da Universidade Federal de Santa Catarina (UFSC).

Destes 33 currículos, 300 publicações foram selecionadas de forma aleatória e manualmente avaliadas: caso possuísem um registro Qualis equivalente então a publicação juntamente com o Qualis associado eram salvos no conjunto de testes; caso não possuísem registro Qualis equivalente, então eram marcados como tal e também adicionados ao conjunto de testes.

Nas avaliações descritas a seguir, foram dadas como *query* ao sistema cada uma das publicações da coleção de testes, porém sem expor os resultados manualmente avaliados. Cada resposta retornada pelo sistema foi comparada com a respectiva resposta manualmente selecionada. Em caso das respostas serem idênticas, então o sistema avaliou corretamente a publicação; em caso de não serem idênticas, avaliou incorretamente. O caso de não haver registro Qualis equivalente à publicação foi considerada uma resposta correta quando o sistema não retornou nenhum resultado, e uma resposta incorreta caso contrário.

4.3.2 Métricas utilizadas

Uma vez definido o conjunto de teste, é preciso definir as métricas utilizadas na avaliação. Através da comparação das métricas é possível concluir se houve melhora em certos aspectos do sistema. No caso do SILQ, deseja-se melhorar a taxa de acerto, ou seja, maximizar o número de trabalhos corretamente avaliados. As métricas clássicas de avaliação de sistemas IR discutidas na Seção 2.2.3 foram consideradas.

As métricas de precisão e revocação foram descartadas por não se encaixarem com a forma de avaliação do sistema, baseada em *rank*. Conforme já discutido, estas métricas não são indicadas para sistemas deste tipo. Medidas mais indicadas nesse caso são *Precision at k* ($P@k$) e *R-Precision*. O algoritmo de avaliação do SILQ, porém, considera apenas o primeiro registro Qualis retornado para realizar *match* com o trabalho sendo avaliado (apenas o mais similar). Neste caso, a avaliação usando estas duas métricas devem considerar apenas o primeiro resultado, ou seja, $P@1$ e *R-Precision* com $|R| = 1$ (sendo R o número de registros relevantes para a *query*). Em ambos os casos, para cada *match* retornado pelo sistema, temos medidas com valor igual a 0, caso o sistema não tenha avaliado corretamente o trabalho, e 1 caso tenha avaliado corretamente. Têm-se, portanto, um simples valor *booleano* indicando se houve acerto ou não, para cada *query* submetida. Consi-

derando todo o conjunto de testes, pode-se somar o número de acertos e dividir pelo tamanho do conjunto, resultado um valor que indica a *taxa de acerto* do sistema. Este valor também é conhecido como *exatidão*¹ e foi a medida base escolhida para a avaliação experimental do sistema.

Outra medida utilizada em um primeiro momento foi a Média de Rank Recíproco (MRR). Conforme discutido, ela é particularmente interessante para sistemas que produzem uma lista de resultados ordenados por probabilidade de corretude, e, ao contrário da exatidão, é capaz de modelar o quão bem o sistema classificou o resultado correto, mesmo quando ele não foi classificado em primeiro lugar.

Por estas razões, as medidas de exatidão (ou taxa de acerto) e MRR foram escolhidas para as avaliações experimentais descritas nas próximas subseções.

4.3.3 Avaliação de *threshold* ideal

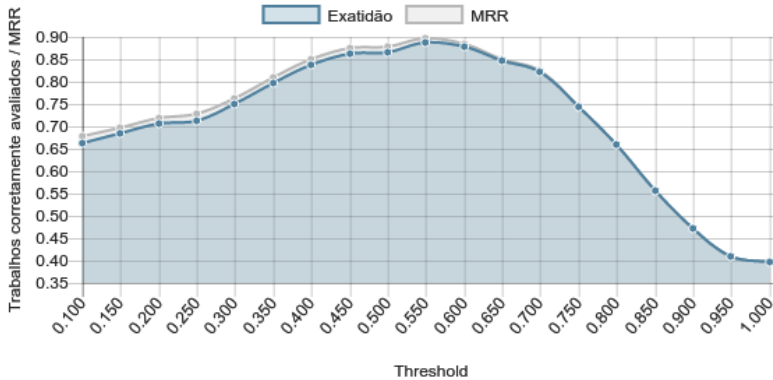
Um dos questionamentos levantados no início deste trabalho e que geralmente ocorre ao projetar sistemas de *data matching* baseados em similaridade, é o de qual *threshold* utilizar. Na primeira versão do sistema foi introduzido um controle de “nível de confiança” que permitia ao usuário controlar o *threshold* utilizado pelo algoritmo, conforme detalhado na Seção 3.1.2. O nível de confiança padrão, porém, foi fixado em 60% (equivalente ao *threshold* de valor 0.6). Este valor foi provavelmente escolhido de forma empírica pois observou-se que maximizava o número de resultados corretos, porém não foram realizados experimentos comprovando esta teoria.

Desta forma, para encontrar o valor de *threshold* ideal foi utilizado o conjunto de testes para avaliar o algoritmo inicial do SILQ 1, em um primeiro momento. O método utilizado foi o de avaliar via sistema cada uma das publicações do conjunto de testes e comparar com o resultado real, e repetir o processo variando o *threshold* a fim de observar as médias de exatidão e de rank recíproco (MRR). Os resultados foram agrupados no gráfico da Figura 8. A linha em azul claro representa a exatidão, ou seja, a taxa de trabalhos corretamente avaliados pelo sistema. A linha em cinza representa a média de rank recíproco (MRR), calculada conforme discutido na Seção 2.2.3.3.

O primeiro fenômeno que observamos ao avaliar o gráfico é o

¹O termo utilizado na literatura é *accuracy*, cuja tradução usual é *precisão*. Optou-se pelo uso do termo *exatidão*, no entanto, para evitar confusões com a métrica de precisão.

Figura 8 – Taxa de trabalhos corretamente avaliados e Média de Rank Recíproco (MRR) para diferentes *thresholds*



ponto de máximo por volta do valor 0.55 de *threshold*, que totaliza uma exatidão aproximada de 88%, e a tendência da exatidão baixar ao se afastar deste pico, para ambas as direções. Esse é um comportamento esperado pois valores de *threshold* baixos tendem a diminuir a exatidão do sistema por retornar resultados não relevantes para as *queries*, enquanto valores altos tendem a diminuir a exatidão por deixar de retornar resultados relevantes. Este ponto máximo trata-se, portanto, do *threshold* ideal para o caso de testes em questão.

Outra característica observada é a tendência do valor de MRR acompanhar o da exatidão, sendo sempre igual ou apenas um pouco superior em magnitude. Isso acontece pela forma com que o MRR é calculado, atribuindo valor de $1/r$ a cada avaliação, sendo r a posição em que o resultado real foi avaliado pelo sistema. Se o resultado foi corretamente avaliado, portanto, o valor de $1/1 = 1$ é atribuído ao resultado, o mesmo valor que seria atribuído à exatidão, já que o conjunto de valores possíveis para esta métrica é $\{0, 1\}$ para cada resultado (0 representando um erro e 1 representando um acerto). A semelhança dos valores, portanto, indica que houveram poucos casos em que o algoritmo retornou o resultado real em posições inferiores à primeira no *rank* de avaliação. Esta característica do valor de MRR permaneceu constante nos demais testes realizados neste trabalho, portanto omitiu-se o valor de MRR nas demais avaliações.

4.3.4 Avaliação dos algoritmos

Os algoritmos descritos na Seção 4.2 foram avaliados utilizando o mesmo processo descrito na Seção anterior. O algoritmo *trigrams* trata-se do método inicial utilizado pelo SILQ 1 e cuja análise de *threshold* ideal foi realizada na Seção anterior. O algoritmo $\text{fb}(t)$ foi testado variando t nos valores que obtiveram melhores resultados. Todas as análises foram realizadas com valor de *threshold* igual a 0.55. A Tabela 5 apresenta os resultados de cada teste.

Tabela 5 – Comparação da exatidão dos diferentes algoritmos testados (utilizando *threshold* de 0.55)

| Algoritmo | Exatidão |
|-----------------------|----------------|
| <i>trigrams</i> | 88.667% |
| <i>fb(1.00)</i> | 89.667% |
| <i>fb(0.90)</i> | 90.667% |
| <i>fb(0.80)</i> | 92.667% |
| <i>fb(0.70)</i> | 92.667% |
| <i>fb(0.60)</i> | 91.000% |
| <i>query_aliasing</i> | 93.333% |

A primeira tentativa de usar *feedback* de relevância na avaliação foi com o algoritmo *fb(1.00)*, que considera os resultados informados pelo usuário quando a *query* é idêntica a algum *feedback*. Houve uma melhora na taxa de acertos, porém de forma não tão significativa.

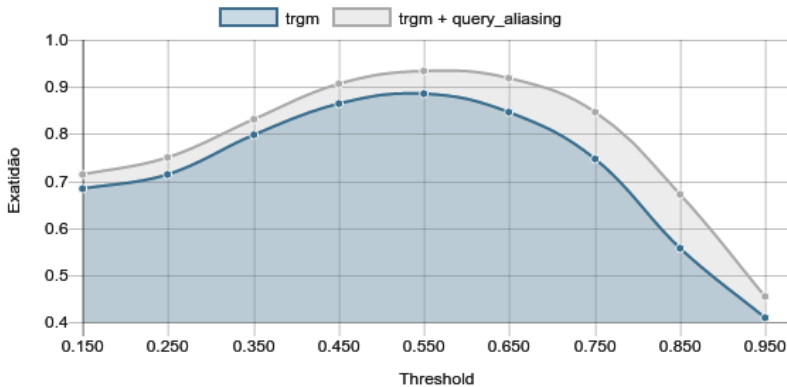
As variações que utilizam valores menores de t , porém, obtiveram melhores resultados, por serem capazes de identificar *queries* similares aos *feedbacks* já informados pelo usuário, mesmo quando este não julgou exatamente a *query* em questão. Um exemplo que demonstra este fato são os nomes de eventos “*IEEE International Symposium on Computer-Based Medical Systems*” e “*27th International Symposium on ComputerBased Medical Systems (CBMS)*”, extraídos de um mesmo currículo Lattes. É fácil notar que tratam-se do mesmo evento, porém o usuário informou a sigla e o número da edição no segundo, e nenhuma destas informações no primeiro (além de não utilizar o hífen em um dos casos). Caso o usuário tenha dado *feedback* para somente um dos casos, os algoritmos $\text{fb}(t)$ com valores de t inferiores a 1.0 são capazes de utilizar o mesmo *feedback* para ambos, apesar das *queries* não serem idênticas.

Valores de t muito baixos, porém, deterioram rapidamente a taxa de acerto pois consideram *feedbacks* similares entre eventos que não tem relação. Desta forma, existe também um “valor ideal” de t , que gira em torno de 0.7 a 0.8 conforme os testes realizados.

O algoritmo $\text{fb}(t)$, porém, pode cometer “injustiças” pois considera os *feedbacks* como resultados corretos, independente dos resultados primários retornados, caso sejam superiores ao *threshold* t . O algoritmo de *query_aliasing* resolve este problema inserindo o *feedback* no *ranking* junto com os demais resultados. O melhor resultado (aquele mais similar à *query*) será utilizado, independente da técnica usada para obtê-lo. Nos testes realizados, o algoritmo de *query_aliasing* obteve a melhor taxa de acerto, com uma média de 93.3% de trabalhos corretamente avaliados.

Foi realizada uma última avaliação comparativa entre o algoritmo *trgm* inicial e o novo que utiliza *query_aliasing*. A Figura 9 mostra a taxa de acerto média para ambos os algoritmos variando o *threshold* utilizado. Nota-se que o algoritmo que utiliza *feedback* de relevância obteve melhores resultados, para qualquer *threshold* utilizado, aumentando em aproximadamente 6% a taxa de acerto. O *threshold* ideal, porém, se manteve constante em 0.55 pois é dependente da função de similaridade.

Figura 9 – Comparação da taxa de acerto do algoritmo *trgm* e do *trgm* + *query_aliasing* para diferentes *thresholds*



5 CONCLUSÕES E TRABALHOS FUTUROS

O SILQ é um esforço coletivo para a automatização e consequente melhoria na qualidade de gestão de grupos de pesquisa e principalmente de Programas de Pós-Graduação. A atualização tecnológica e alimentação da base de dados da ferramenta são processos que devem ser constantemente realizados para mantê-la viável aos seus usuários.

Os objetivos iniciais de refatoração arquitetural com inclusão de melhorias de interface, novas funcionalidades e criação da camada de integração REST, foram alcançados. Tanto usuários como desenvolvedores de aplicações interessados no sistema se beneficiam dessa alteração. Também foram realizadas as inclusões de controles permitindo sugestões de resultados por parte dos usuários, e desenvolvidos dois algoritmos que utilizam tais informações para melhorar a taxa de acerto do sistema.

A hipótese de pesquisa levantada se mostrou verdadeira. De fato foi possível aumentar a precisão do sistema utilizando *feedback* de relevância dos usuários. Isso se mostrou verdade por meio dos experimentos realizados e relatados neste trabalho, e com o teste dos algoritmos propostos.

A avaliação de *threshold* mostrou que o valor ideal para o SILQ é de aproximadamente 0.55, muito próximo do “nível de confiança normal” estabelecido no trabalho anterior de (AGUIAR; COSTA, 2015), e que a taxa de acerto para este caso é de 88%.

Os algoritmos *fb(t)* e *query_aliasing*, baseados em *feedback* de relevância e similaridade textual, foram propostos e avaliados experimentalmente. Ambos resultaram em melhoria na taxa de acerto do sistema. O algoritmo *query_aliasing* foi preferido por resultar em uma melhor precisão e ser de mais fácil implementação, e está atualmente em uso na versão 2.3 do sistema. Através da alteração do *threshold* ideal e da inclusão do algoritmo baseado em *feedback* de relevância, a taxa de acerto média do sistema aumentou de 87% para 93.3%.

Esses resultados são relevantes não só por mostrarem que a taxa de acerto média do sistema aumentou, mas por estabelecerem uma medida base para trabalhos futuros. Os algoritmos utilizados neste trabalho são estratégias simples de uso de *feedback* de relevância construídas sobre a mesma função de similaridade textual do SILQ 1, o método *trigrams*, porém outros algoritmos e estratégias diferentes podem ser testados. Em especial, ambos algoritmos propostos consideram apenas

o conjunto de *feedbacks* de um usuário específico, de forma a evitar a necessidade do tratamento de *feedbacks* conflitantes ou divergentes de diferentes usuários.

Como trabalhos futuros pode-se sugerir os seguintes itens:

- Propor e analisar outros métodos de similaridade textual ou estratégias diferentes que possam melhorar ainda mais a precisão do SILQ. Alguns exemplos seriam o uso de *machine learning* para treino do algoritmo de classificação, uma estratégia conhecida por *learning to rank*; e o algoritmo de *Rocchio*, muito utilizada em sistemas que implementam *feedback* de relevância e baseado no modelo de espaço vetorial;
- Considerar *feedbacks* provindos de terceiros, ou seja, levar em conta *feedbacks* informados por outros usuários, mesmo que o atual não tenha informado *feedback* a respeito de uma *query* específica;
- Considerar nomes de eventos traduzidos. Comumente são utilizados nomes de periódicos ou eventos estrangeiros em inglês no currículo Lattes, porém em português no Qualis, ou vice-versa. Esses casos dificilmente são avaliados corretamente pelo sistema, já que o algoritmo atual é baseado em comparação por similaridade textual;
- Levantamento da produção por veículo de publicação. Considerando todos os currículos cadastrados no SILQ, seria possível levantar o número de publicações em cada periódico e evento cadastrado no Qualis;
- Gerar os valores de $I_{restrito}$ e I_{geral} automaticamente. Tratam-se de índices definidos pela CAPES e utilizados na avaliação de Programas de Pós-Graduação;
- Considerar, para fins de ponderação dos valores de estrato Qualis, os pesos considerados pela avaliação de Programas de Pós-Graduação realizada pela CAPES.

As mudanças relatadas neste trabalho já se encontram disponíveis na página oficial do SILQ: <http://silq.inf.ufsc.br/>, na versão 2.3 no momento de escrita deste trabalho, porém em constante evolução.

REFERÊNCIAS

- AGUIAR, F. N. M. de; COSTA, M. E. **SILQ - Sistema de Integração Lattes Qualis**. Florianópolis: Universidade Federal de Santa Catarina, Biblioteca Universitária, 2015.
- BAEZA-YATES, R. A.; RIBEIRO-NETO, B. **Modern Information Retrieval**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.
- BARBOSA, E. F.; MALDONADO, J. C.; VINCENZI, A. M. R. Introdução ao teste de software. **XIV Simpósio Brasileiro de Engenharia de Software**, 2000.
- BOEHM, B. W. **Software Engineering Economics**. Englewood Cliffs, NJ: Prentice Hall, 1981.
- BOOTH, D. et al. W3C Working Group Note, **Web Services Architecture**. fev. 2004. Disponível em: <<https://www.w3.org/TR/2004/NOTE-ws-arch-20040211>>.
- BOURQUE, P.; FAIRLEY, R. E. (Ed.). **SWEBOK: Guide to the Software Engineering Body of Knowledge**. Version 3.0. Los Alamitos, CA: IEEE Computer Society, 2014. ISBN 978-0-7695-5166-1. Disponível em: <<http://www.swebok.org/>>.
- BRODER, A. Z. et al. Syntactic clustering of the web. **Comput. Netw. ISDN Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 29, n. 8-13, p. 1157-1166, set. 1997. ISSN 0169-7552. Disponível em: <[http://dx.doi.org/10.1016/S0169-7552\(97\)00031-7](http://dx.doi.org/10.1016/S0169-7552(97)00031-7)>.
- CAPES. **Classificação da produção intelectual**. 2015. Disponível em: <<http://www.capes.gov.br/avaliacao/instrumentos-de-apoio/classificacao-da-producao-intelectual>>. Acesso em: 17/11/2015.
- CAPES. **História e missão**. 2015. Disponível em: <<http://www.capes.gov.br/historia-e-missao>>. Acesso em: 22/05/2016.
- CARVALHO, M. G. P. et al. Ppgi-sgpc sistema para gestão da produção científica. 2009. Disponível em:

<<http://www2.ic.uff.br/vanessa/papers/carvalho2009-sbbd-demos.pdf>>.

CESAR.EDU. **Qualis de Eventos e Periódicos**. 2014. Disponível em: <<http://www.cesar.edu.br/qualis/>>. Acesso em: 18/06/2016.

CNPQ. **Módulo Produção Bibliográfica**. 2015. Disponível em: <http://ajuda.cnpq.br/index.php/Módulo_Produção_Bibliográfica>. Acesso em: 17/11/2015.

CNPQ. **Sobre a Plataforma Lattes**. 2015. Disponível em: <<http://www.cnpq.br/web/portal-lattes/sobre-a-plataforma>>. Acesso em: 17/11/2015.

CNPQ. **Conselho Nacional de Desenvolvimento Científico e Tecnológico: Institucional**. 2016. Disponível em: <http://cnpq.br/apresentacao_institucional/>. Acesso em: 22/05/2016.

CNPQ. **Histórico**: História do surgimento da plataforma lattes. 2016. Disponível em: <<http://memoria.cnpq.br/web/portal-lattes/historico>>. Acesso em: 31/05/2016.

CRISPIM, B. do A. (Editorial) A importância da publicação de artigos científicos: Uma abordagem para área de ciência animal. **Journal of the Selva Andina Animal Science**, v. 1, n. 1, 2014.

DORNELES, C. F.; GONÇALVES, R.; MELLO, R. dos S. Approximate data instance matching: a survey. **Knowledge and Information Systems**, v. 27, n. 1, 2011. ISSN 0219-3116. Disponível em: <<http://dx.doi.org/10.1007/s10115-010-0285-0>>.

EUZENAT, J. Semantic precision and recall for ontology alignment evaluation. In: **Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)**, Hyderabad, India, January 6-12, 2007. [S.l.: s.n.], 2007. p. 348-353.

FIELDING, R. T.; TAYLOR, R. N. Principled design of the modern web architecture. **ACM Transactions on Internet Technology**, v. 2, p. 115-150, 2002.

FOWLER, M.; LEWIS, J. **Microservices**. 2014. Disponível em: <<http://martinfowler.com/articles/microservices.html>>. Acesso em: 07/06/2016.

JIA, W.; ZHOU, W. **Distributed Network Systems: From concepts to implementations**. Boston: Springer Science and Business Media Inc, 2005.

KELLY, D.; BELKIN, N. J. Reading time, scrolling and interaction: Exploring implicit sources of user preferences for relevance feedback. In: **Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 2001. (SIGIR '01), p. 408–409. ISBN 1-58113-331-6. Disponível em: <<http://doi.acm.org/10.1145/383952.384045>>.

KELLY, D.; TEEVAN, J. Implicit feedback for inferring user preference: A bibliography. 2003.

LIU, L.; ZSU, M. T. **Encyclopedia of Database Systems**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2009. ISBN 0387355448, 9780387355443.

MANNING, C. D.; RAGHAVAN, P.; SCHUTZE, H. **Introduction to Information Retrieval**. New York, NY, USA: Cambridge University Press, 2008. ISBN 0521865719, 9780521865715.

MCFEE, B.; LANCKRIET, G. Metric learning to rank. In: **Proceedings of the 27th annual International Conference on Machine Learning (ICML)**. [S.l.: s.n.], 2010.

MCTI. **Ministério da Ciência, Tecnologia e Inovação: Institucional**. 2016. Disponível em: <<http://www.mcti.gov.br/institucional>>. Acesso em: 22/06/2016.

MENA-CHALCO, J. P.; CESAR-JR, R. M. Scriptlattes: An open-source knowledge extraction system from the lattes platform. **Journal of the Brazilian Computer Society**, v. 15, n. 4, p. 31–39, 2009.

PINTO, R. de S. **WebQualis 2012**. 2012. Disponível em: <<http://qualis.renesp.com.br/index.php>>. Acesso em: 18/06/2016.

PRESSMAN, R. S. **Software Engineering: A Practitioner's Approach**. 5th. ed. [S.l.]: McGraw Hill, 2001. ISBN 0-07-365578-3.

RICHARDSON, L. **RESTful Web APIs**. Sebastopol, Calif: O'Reilly, 2013. ISBN 9781449358068.

RIPLEY, B. D.; HJORT, N. L. **Pattern Recognition and Neural Networks**. 1st. ed. New York, NY, USA: Cambridge University Press, 1995. ISBN 0521460867.

RUI, Y.; HUANG, T. S.; MEHROTRA, S. Content-based image retrieval with relevance feedback in mars: A new content-based image retrieval relevance feedback algorithm. 1997.

SAMMUT, C.; WEBB, G. I. **Encyclopedia of Machine Learning**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2011. ISBN 0387307680, 9780387307688.

SILVA, R. D. et al. Measuring quality of similarity functions in approximate data matching. **Journal of Informetrics**, 2007.

SPINK, A.; JANSEN, B.; OZMULTU, H. Use of query reformulation and relevance feedback by excite users. **Internet Research**, Emerald Group Publishing Ltd., v. 10, n. 4, p. 317–328, 2000. ISSN 1066-2243.

TAO, D. et al. Direct kernel biased discriminant analysis: A new content-based image retrieval relevance feedback algorithm. **IEEE Transactions on Multimedia**, v. 8, n. 4, 2006. ISSN 1520-9210.

WHITE, R. W.; JOSE, J. M.; RUTHVEN, I. Comparing explicit and implicit feedback techniques for web retrieval: Trec-10 interactive track report. In: VOORHEES, E.; HARMAN, D. (Ed.). **Proceedings of the Tenth Text Retrieval Conference (TREC-10)**. [S.l.]: NIST, 2001. p. 534–538.

WITTEN, I. H.; FRANK, E. **Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005. ISBN 0120884070.

APÊNDICE A – Artigo

Análise do uso de feedback de relevância no Sistema de Integração Lattes-Qualis (SILQ)

Carlos Bonetti¹

¹Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

carlosbonetti.mail@gmail.com

Abstract. *SILQ emerged in 2015 with the purpose of matching the vehicles qualified by Qualis with vehicles of publications registered on the Lattes curriculum of researchers, in order to automate the process of generating quality indicators of scientific production from these curriculums. The purpose of this work is a set of modifications to the system to include the most updated Qualis data, a study of the system's accuracy and the inclusion of controls allowing the users to suggest matchings, a technique named relevance feedback, proposing two new algorithms and an experimental analysis to evaluate their efficiency at the system.*

Resumo. *O SILQ surgiu em 2015 com o objetivo de realizar o matching automático dos veículos qualificados no Qualis com os veículos de publicações cadastradas no currículo Lattes de pesquisadores, a fim de automatizar o processo de geração de indicadores das produções contidas nesses currículos. Este trabalho propõe um conjunto de alterações no sistema para inclusão dos dados Qualis mais recentes, um estudo da taxa de acerto do sistema e a inclusão de controles que permitam ao usuário sugerir matching, técnica denominada feedback de relevância, propondo dois novos algoritmos e uma avaliação experimental para análise de sua eficácia no sistema.*

1. Introdução

A Plataforma Lattes, criada e mantida pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), é um sistema de informação responsável pela integração da base de dados de currículos, grupos de pesquisa e instituições. O Currículo Lattes se tornou o padrão nacional no registro da vida científica de estudantes, professores e pesquisadores e é hoje adotada por institutos e universidades de todo o país [CNPQ 2015b]. No Currículo Lattes pode-se inserir dados gerais do pesquisador, produção bibliográfica, orientações, citações, participações em eventos científicos entre outros dados. No módulo Produção Bibliográfica, por exemplo, é possível a inserção de artigos publicados ou aceitos para publicação em periódicos indexados pelo ISSN [CNPQ 2015a].

A qualidade da produção bibliográfica dos Programas de Pós-Graduação é classificada pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) através de um conjunto de procedimentos denominado Qualis. O Qualis define estratos de qualificação a partir da análise da qualidade dos veículos onde a produção científica de pesquisadores brasileiros foi divulgada, ou seja, periódicos e eventos científicos. Esta análise é realizada em um processo anual de atualização, sendo os veículos enquadrados

nos estratos indicativos de qualidade A1, A2, B1, B2, B3, B4, B5 e C, do maior para o menor peso, para cada área do conhecimento [CAPES 2015].

Apesar da Plataforma Lattes possuir um módulo de inclusão de publicações e permitir a definição do veículo onde este foi publicado, não há qualquer tipo de conexão entre os sistemas Lattes e Qualis, ou seja, o processo de avaliação de uma publicação (que é feita através da avaliação do veículo onde este foi publicado) deve ser realizado de forma manual.

O Sistema de Integração Lattes Qualis (SILQ) surgiu no ano de 2015, desenvolvido como Trabalho de Conclusão de Curso dos alunos Felipe Nedel Mendes de Aguiar e Maria Eloísa Costa do curso de Ciência da Computação da Universidade Federal de Santa Catarina (UFSC), orientados pela Prof. Carina F. Dorneles, da mesma instituição. O objetivo do sistema é a classificação automática da produção científica do currículo Lattes do pesquisador, através do *matching* por similaridade de dados extraídos do Qualis, usando uma interface web amigável [de Aguiar and Costa 2015].

A primeira versão do sistema foi finalizada em 2015 e desde então encontra-se disponível de forma pública e gratuita através do endereço <http://silq.inf.ufsc.br/>.

Um ponto de melhoria no sistema seria a avaliação experimental do algoritmo de classificação do SILQ para o estabelecimento de uma medida do grau de precisão da ferramenta. A partir disto, outras técnicas podem ser propostas e comparadas com as anteriores a fim de provar se elas resultaram em ganho de precisão.

Como o *matching* entre os dados do Qualis e do Lattes de pesquisadores é feito através de funções de similaridade, o resultado pode ser um falso positivo. Portanto, um ponto importante a ser trabalhado é permitir ao usuário informar ao sistema sugestões de resultados. Esta técnica, conhecida por *feedback* de relevância, exige alterar o algoritmo de classificação a fim de treiná-lo com as informações providas pelo usuário e melhorar os resultados de pesquisas similares subsequentemente realizadas. Também se faz necessária a mensuração da taxa de acerto do sistema após a inclusão desta técnica e avaliar se ela beneficia a precisão do algoritmo de classificação.

A hipótese de pesquisa a ser avaliada a partir desta ideia, portanto, é avaliar se o *feedback* de usuários pode melhorar a taxa de acerto do SILQ, ou seja, aumentar o número de trabalhos corretamente avaliados pelo sistema.

O objetivo geral deste trabalho é analisar o impacto que o uso de *feedback* de relevância tem na precisão dos resultados de avaliações realizadas pelo SILQ, efetuado sobre uma nova arquitetura da ferramenta que inclui a criação de API de integração com outros sistemas e a atualização da base de dados conforme as novas classificações Qualis.

2. Histórico e Visão Geral do SILQ 1

Este trabalho é uma continuação de [de Aguiar and Costa 2015], um Trabalho de Conclusão de Curso de alunos do curso de Ciência da Computação da UFSC, orientados pela Professora Carina F. Dorneles. O objetivo desse trabalho de 2015 era a criação de um sistema que deveria ser capaz de qualificar produções científicas, nas categorias artigos e trabalhos apresentados em eventos, por busca por similaridade de dados com os dados extraídos do WebQualis [de Aguiar and Costa 2015, p. 26-27]. Este objetivo foi alcançado

com a criação da primeira versão do Sistema de Integração Lattes-Qualis (SILQ), lançado no segundo semestre de 2015 e disponível no sítio <http://silq.inf.ufsc.br/>.

Apesar de estável e com sua função principal sendo desempenhada de forma satisfatória, o SILQ 1 deixou algumas lacunas e melhorias a serem desenvolvidas por trabalhos futuros. Segundo os próprios autores, “[...] o SILQ foi concebido para ser uma ferramenta de domínio público e vários projetos devem nascer a partir dele. A continuidade do projeto só tem a acrescentar ao mundo acadêmico [...]” [de Aguiar and Costa 2015, p. 79], o que motivou a criação do trabalho para a continuação da proposta original.

2.1. Como o SILQ realiza o *matching* Qualis-Lattes

O *matching* da produção científica de um currículo Lattes com o Qualis é realizado pelo SILQ com base na similaridade textual entre o título do evento de cada trabalho presente no currículo e o título do evento Qualis¹. O título do veículo onde o trabalho foi publicado, juntamente com seu ano e área de avaliação cadastrados no Lattes são dados como *query* para o sistema. O sistema busca sobre todo o conjunto de dados Qualis presentes na base de dados aquele com maior similaridade textual em relação à *query*. Se esta similaridade for maior do que um *threshold* pré-estabelecido, então o resultado é considerado um *match* válido e o estrato atribuído pelo Qualis a este evento é também atribuído ao trabalho do pesquisador.

Pode-se tomar o exemplo real de um trabalho qualquer extraído de um currículo Lattes cujo evento tenha sido especificado como “2016 IEEE 7th Latin American Symposium on Circuits & Systems (LASCAS)”. A Tabela 1 mostra os resultados retornados pelo SILQ junto com seus respectivos valores de similaridade textual.

Tabela 1. Resultados retornados pelo SILQ para a *query* “2016 IEEE 7th Latin American Symposium on Circuits & Systems (LASCAS)”

| Evento | Estrato | Similaridade |
|---|---------|--------------|
| IEEE Latin American Symposium on Circuits and Systems (LASCAS) | B5 | 0.87 |
| IEEE International Symposium on Circuits and Systems (ISCAS) | A1 | 0.48 |
| IEEE Latin American Robotics Symposium (LARS) | B4 | 0.45 |
| IEEE International Midwest Symposium on Circuits and Systems (MWSCAS) | B1 | 0.43 |
| Symposium on Asynchronous Circuits and Systems (ASYNC) | A2 | 0.40 |

O primeiro resultado retornado é o escolhido, por possuir a maior similaridade em relação à *query*. Neste caso, o trabalho é avaliado com o conceito B5, já que este é conceito Qualis atribuído ao evento.

¹O *matching* por similaridade é utilizado somente para eventos. Para periódicos, o *matching* é realizado através de comparação entre o ISSN informado no Lattes e nos registros Qualis.

O algoritmo de avaliação da primeira versão do SILQ retorna os resultados da Tabela 1 ao ser configurado para utilizar um “nível de confiança” de 40%. Esse nível de confiança é o *threshold* utilizado pelo algoritmo de classificação. Qualquer resultado cujo nível de similaridade em relação à *query* seja inferior ao nível de confiança utilizado não é retornado.

Figura 1. Diálogo de configurações de avaliação

Escolha as configurações de avaliação

Área de Atuação *

Ciência da Computação

As avaliações serão realizadas de acordo com a área selecionada.

Publicações de

2012

Até

2016

Nível de confiança

Normal (60%)

Estabelece um limite mínimo da qualidade da avaliação.

Iniciar Avaliação

O nível de confiança pode ser ajustado através das opções de avaliação, apresentadas quando o usuário requisita uma avaliação de currículo Lattes (Figura 1). Diminuir o nível de confiança (e em consequência o *threshold* do algoritmo de classificação) implica em obter mais resultados e classificar mais trabalhos, porém diminuir a precisão do algoritmo, já que resultados não relevantes serão retornados para as *queries* que não obtiveram bons resultados (resultados com nível de similaridade alto). A Seção 3.3 apresenta testes de validação do algoritmo que indicam o nível de precisão obtidos para cada valor de *threshold* utilizado, além de sugerir um nível de confiança ideal a ser utilizado para maximizar o número de trabalhos corretamente avaliados pelo sistema.

3. Uso de *Feedback* de relevância

Um dos itens de melhoria indicados por [de Aguiar and Costa 2015] após a realização da primeira versão do sistema seria “permitir que o usuário auxilie a ferramenta na qualificação”. Em sistemas de IR, esta característica é denominada *Feedback de Relevância*.

A ideia por trás desta técnica é permitir que o usuário julgue resultados iniciais retornados pelo sistema, classificando-os como relevantes ou não para a *query* vi-

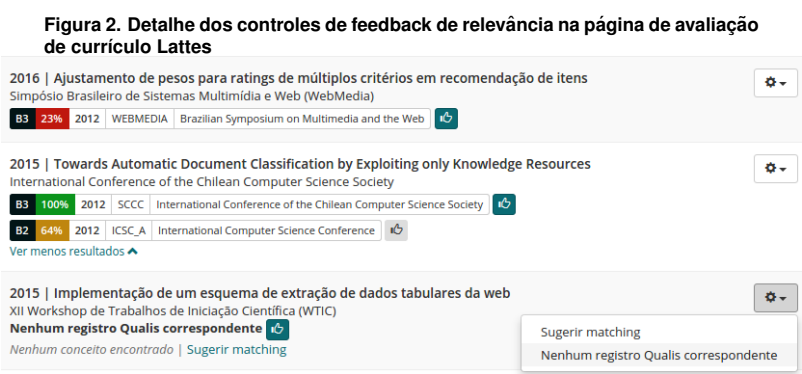
gente. O sistema então é capaz de utilizar esta informação para melhorar seu algoritmo de classificação e retornar melhores resultados para novas *queries*.

3.1. Obtenção de feedback

Em sistemas de IR que implementam *feedback* de relevância, tipicamente é permitido ao usuário julgar como relevante ou não relevante cada item retornado pelo sistema. Isso é natural pois o objetivo deste tipo de sistema é retornar a totalidade do conjunto de itens relevantes, cujo tamanho é variável dependendo da *query*. O SILQ, porém, é um caso específico em que existem somente 0 ou 1 item relevante para toda *query*. Isso acontece pois cada trabalho indicado no currículo Lattes de um pesquisador aparece qualificado no Qualis apenas uma única vez no ano em que o pesquisador o publicou. O objetivo do algoritmo de avaliação do SILQ é deduzir que registro Qualis é esse, caso exista na base de dados.

Para a implementação de *feedback* de relevância no SILQ, portanto, não é necessário o julgamento de cada item retornado pelo sistema em uma avaliação, mas apenas marcar *qual* dos itens da base de dados Qualis é um *match* correto para a *query* vigente. Desta forma, o usuário deve ser capaz de indicar o registro Qualis que deve ser considerado para cada trabalho avaliado. Também existe o caso especial em que não existe um registro Qualis correspondente ao trabalho, neste caso o usuário deve ser capaz de indicar que não existem *matches* corretos para o trabalho.

A Figura 2 mostra o exemplo de três trabalhos extraídos de um currículo Lattes e avaliados pelo SILQ que receberam *feedback* do usuário. O botão de “joinha” é utilizado para marcar o resultado que o usuário considera relevante para cada trabalho. Os botões grifados (com fundo azul) representam que o resultado já foi previamente marcado. Neste caso, o registro Qualis marcado como relevante é associado à *query* atual, que engloba o título, ano e área do trabalho avaliado, juntamente com o usuário que está realizando o julgamento. Também é possível marcar a opção “Nenhum registro Qualis correspondente” ou sugerir algum resultado não retornado previamente pelo sistema em “Sugerir matching” (mostrado no canto inferior direito da Figura exemplo).



Os *feedbacks* de relevância dados pelos usuários são salvos na base de dados SILQ para uso posterior pelo algoritmo de classificação, conforme descrito na Seção 3.2. Já que um trabalho qualquer pode ter no máximo um registro Qualis associado, é salvo somente um *feedback* por *query* por usuário. Desta forma, uma chave única é utilizada na tabela de *feedbacks*, dada pela dupla (título do trabalho, usuário).

3.2. Algoritmos de avaliação com feedback

Uma vez registrados os *feedbacks* dos usuários, passa-se a questionar de que forma utilizá-los para aumentar a taxa de acerto do sistema para futuras consultas. A Seção atual apresenta dois novos algoritmos que foram propostos e avaliados neste trabalho, e como foram implementados, enquanto a próxima Seção apresenta os resultados e comparações de exatidão dos mesmos.

3.2.1. Algoritmo $fb(t)$

Um das abordagens mais simples que podem ser usadas neste caso é utilizar o resultado marcado pelo usuário sempre que uma *query* idêntica ao do *feedback* seja submetida ao sistema. O algoritmo $fb(1.0)$, portanto, foi desenvolvido com base nesta ideia. O valor “1.0” presente no nome do algoritmo apenas indica que o *feedback* é considerado em detrimento de qualquer outro resultado dado pelo sistema quando a *query* submetida for 100% similar (ou seja, idêntica) à *query* do *feedback*.

Pode-se dar o exemplo real de um nome de evento extraído de um currículo Lattes cadastrado pelo pesquisador como “Software Engineering Knowledge Engineering”, no ano de 2009 e com área de avaliação Ciência da Computação. Ao avaliar tal trabalho, o sistema retorna a lista da Tabela 2.

Tabela 2. Resultados retornados pelo SILQ para a *query* “Software Engineering Knowledge Engineering”

| # | Evento | Similaridade |
|-----|---|--------------|
| 1 | Software Engineering and Data Engineering (SEDE) | 0.53 |
| 2 | International Conference on Software Engineering and Knowledge Engineering (SEKE) | 0.49 |
| 3 | Software Engineering and Applications (SEA_A) | 0.45 |
| ... | ... | ... |

Após analisar esta lista, o usuário submeteu um *feedback* ao sistema marcando o resultado #2 como o correto. Desta forma, utilizando o algoritmo $fb(1.0)$, toda *query* subsequente idêntica a “Software Engineering Knowledge Engineering” terá o resultado #2 retornado na primeira posição.

O $fb(1.0)$ considera apenas *feedbacks* que sejam idênticos à *query* submetida, *queries* similares não são consideradas. O usuário do exemplo anterior possui um outro trabalho cadastrado em seu currículo Lattes cujo título de evento é “Software Engineering

and Knowledge Engineering". Pode-se deduzir que o usuário quis se referir ao mesmo evento, porém o título não é idêntico ao exemplo anterior por causa do termo "and". Neste caso, o algoritmo $fb(1.0)$ não é capaz de deduzir que os dois casos se referem ao mesmo evento, apesar da semelhança entre eles. Uma modificação que pode ser realizada no algoritmo é utilizar uma função de similaridade entre novas *queries* submetidas ao sistema com aquelas anteriormente submetidas e que possuem *feedback* do usuário. Se a similaridade entre a nova *query* e algum dos *feedbacks* for maior do que certo *threshold de feedback*, então é provável que a nova *query* se refira ao mesmo evento do *feedback* anteriormente fornecido.

O algoritmo $fb(t)$ (para $0.0 \leq t \leq 1.0$) é uma generalização de $fb(1.0)$ que considera *feedbacks* cuja similaridade textual em relação à *query* seja maior que o *threshold* t . Por exemplo, $fb(0.75)$ irá considerar *feedbacks* cujo valor de similaridade textual em relação à *query* seja 0.75 ou superior. No exemplo anterior, ao submeter a nova *query* "Software Engineering and Knowledge Engineering" ao sistema, o algoritmo $fb(0.75)$ calcula a similaridade entre ela e os *feedbacks* anteriores fornecidos pelo usuário e encontra o *feedback* da primeira *query* "Software Engineering Knowledge Engineering" por ser 88% similar à *query* atual. Neste caso, por ter uma similaridade maior do que o *threshold* de 0.75 estipulado, o algoritmo retorna o mesmo evento marcado no *feedback* para a *query* atual (o evento #2 da Tabela 2).

O algoritmo $fb(t)$, entretanto, leva a outros questionamentos, já que utiliza a mesma técnica de *data-matching* que foi proposta a melhorar. Qual o valor de t (*threshold de feedback*) ideal? Qual o algoritmo de similaridade textual ideal para este caso? A Seção 3.3.4 apresenta testes de validação do algoritmo $fb(t)$ para diferentes valores de t .

O Algoritmo 1 é a representação em pseudocódigo de $fb(t)$. O parâmetro q representa a *query*, t é o valor de *threshold de feedback*, D é o conjunto de todos os documentos a serem pesquisados e F o conjunto de *feedbacks* fornecidos contendo as duplas (q_f, d) , *query* do *feedback* e documento dado como *feedback*, respectivamente, tal que $d \in D$. A variável m é o registro provindo de *feedback* com maior probabilidade de ser um *match* correto para a *query*, caso exista. A saída R é uma lista 0-indexada contendo os resultados da consulta, ordenada por ordem decrescente da probabilidade do resultado ser um *match* correto para a *query*. A função `trigram_sim` calcula a similaridade textual entre duas *strings* utilizando o método *trigrams* e retornando um valor no intervalo $[0, 1]$. A função `trigram_rank` é o algoritmo de avaliação da primeira versão do SILQ, que cria o rank R de similaridade a partir da comparação entre q e cada um dos documentos de D . A função `insert_rank_top` insere um registro no topo do *rank*, removendo itens duplicados previamente inseridos.

3.2.2. Algoritmo *query aliasing*

Uma adaptação de $fb(t)$ que mostrou-se de mais fácil implementação e que não gera o questionamento de qual valor de t utilizar, foi considerar as *queries de feedbacks* anteriormente fornecidos pelo usuário, da mesma forma que o $fb(t)$, porém inseri-las no *rank* de resultados de novas *queries* submetidas com base em seus valores de similaridade textual em relação à nova *query*, junto com os resultados previamente selecionados. Assim,

Algoritmo 1: $fb(t)$

Input : q, t, D, F
Output: R

```
1  $R \leftarrow \text{trigram\_rank}(q, D)$ 
2  $s_m \leftarrow -1$ 
3 for  $(q_f, d) \in F$  do
4    $s \leftarrow \text{trigram\_sim}(q, q_f)$ 
5   if  $s \geq t$  and  $s \geq s_m$  then
6      $m \leftarrow d$ 
7      $s_m \leftarrow s$ 
8   end if
9 end for
10 if  $m$  then
11    $\text{insert\_rank\_top}(R, m)$ 
12 end if
13 return  $R$ 
```

ao invés de escolhê-lo em detrimento dos demais, o evento marcado com *feedback* só é retornado se for mais bem ranqueado que os demais resultados.

Considerando os mesmos exemplos dados na Seção anterior, em que o usuário submete a nova *query* “Software Engineering and Knowledge Engineering” ao sistema, o algoritmo de *query_aliasing* realiza comparação textual entre a nova *query* e as *queries* anteriores que possuam *feedback*, da mesma forma que o $fb(t)$, encontrando a *query* “Software Engineering Knowledge Engineering”, com um valor de similaridade de 0.88. Ao contrário do $fb(t)$, o algoritmo de *query_aliasing* irá inserir o evento dado como *feedback* a esta *query* junto com a lista de resultados previamente encontrados apenas via similaridade textual, usando o valor de 0.88 para posicionamento no *ranking*. A Tabela 3 mostra o *ranking* retornado para este exemplo. Nota-se que o evento #2, marcado pelo usuário como correto, foi elevado no *ranking* por receber o novo valor de similaridade da comparação com o *feedback*.

Tabela 3. Resultados retornados pelo SILQ para a *query* “Software Engineering and Knowledge Engineering” utilizando *query_aliasing*

| # | Evento | Similaridade |
|-----|---|--------------|
| 2 | International Conference on Software Engineering and Knowledge Engineering (SEKE) | 0.88 |
| 1 | Software Engineering and Data Engineering (SEDE) | 0.62 |
| 3 | Software Engineering and Applications (SEA_A) | 0.53 |
| ... | ... | ... |

O valor de similaridade atribuído, porém, perde seu significado semântico pois não é mais a similaridade textual entre o título do evento do Lattes e do Qualis calculado

através do *trigrams*, mas um valor adimensional usado apenas para ordenação relativa dentro do *ranking*.

Desta forma, ao processar uma *query* q qualquer, o sistema processa um *rank* de resultados primários com base no algoritmo *trigrams* inicial. O *rank* é ordenado do resultado mais similar à q ao menos similar. Após esta etapa, ele também compara q com cada uma das *queries* anteriormente submetidas pelo usuário e que possuem *feedback* de relevância utilizando o mesmo algoritmo de similaridade textual. O resultado mais similar é inserido no *ranking* de resultados. Assim, se q é idêntico a um *feedback* já submetido pelo usuário, o evento deste *feedback* será retornado e inserido no topo do *ranking* de resultados, por ser 100% similar à *query*. Outros resultados similares, porém não idênticos, serão inseridos no *ranking* conforme seu valor de similaridade e só serão escolhidos em detrimento de outros resultados primários se seus valores de similaridade forem superiores a eles.

É como se, ao dar um *feedback* de relevância qualquer, o usuário criasse um *alias* (um apelido) ao resultado que está sugerindo. Assim, o sistema deve avaliar novas *queries* não só comparando-as com o nome real dos documentos, mas também com os apelidos dados a eles pelo usuário. Por este motivo o algoritmo foi chamado de *query_aliasing*. A avaliação deste algoritmo foi realizada e comparada com os demais na Seção 3.3.4.

O Algoritmo 2 é a representação em pseudocódigo do método proposto. O significado semântico das variáveis é equivalente ao do Algoritmo 1. A saída deste algoritmo é o próprio *rank* R , possivelmente contendo novos resultados devido à comparação com os *feedbacks* de F . O método *insert_rank* da linha 4 insere o registro d na lista ordenada R com um valor de *rank* s , preservando a ordenação da lista, de forma que o elemento em $R[0]$ seja aquele com maior valor de *rank* e, assim, o registro com maior probabilidade de ser um *match* correto para a *query*. A função também elimina itens duplicados, preservando aquele com maior valor de *rank*.

Algoritmo 2: *query_aliasing*

Input : q, D, F

Output: R

```

1  $R \leftarrow \text{trigram\_rank}(q, D)$ 
2 for  $(q_f, d) \in F$  do
3    $s \leftarrow \text{trigram\_sim}(q, q_f)$ 
4    $\text{insert\_rank}(R, s, d)$ 
5 end for
6 return  $R$ 
```

3.3. Avaliação experimental

Esta Seção apresenta os procedimentos realizados para avaliar as alterações promovidas no algoritmo de avaliação da nova versão do SILQ e se elas contribuíram para o aumento da taxa de acerto do sistema.

3.3.1. Conjunto de testes

O conjunto de testes utilizado para a avaliação do sistema foi criado a partir dos currículos Lattes de 33 pesquisadores do programa de pós-graduação em Ciência da Computação da Universidade Federal de Santa Catarina (UFSC).

Destes 33 currículos, 300 publicações foram selecionadas de forma aleatória e manualmente avaliadas: caso possuísem um registro Qualis equivalente então a publicação juntamente com o Qualis associado eram salvos no conjunto de testes; caso não possuísem registro Qualis equivalente, então eram marcados como tal e também adicionados ao conjunto de testes.

Nas avaliações descritas a seguir, foram dadas como *query* ao sistema cada uma das publicações da coleção de testes, porém sem expor os resultados manualmente avaliados. Cada resposta retornada pelo sistema foi comparada com a respectiva resposta manualmente selecionada. Em caso das respostas serem idênticas, então o sistema avaliou corretamente a publicação; em caso de não serem idênticas, avaliou incorretamente. O caso de não haver registro Qualis equivalente à publicação foi considerada uma resposta correta quando o sistema não retornou nenhum resultado, e uma resposta incorreta caso contrário.

3.3.2. Métricas utilizadas

Uma vez definido o conjunto de teste, é preciso definir as métricas utilizadas na avaliação. Através da comparação das métricas é possível concluir se houve melhora em certos aspectos do sistema. No caso do SILQ, deseja-se melhorar a taxa de acerto, ou seja, maximizar o número de trabalhos corretamente avaliados. Métricas clássicas de avaliação de sistemas IR foram consideradas.

As métricas de precisão e revocação foram descartadas por não se encaixarem com a forma de avaliação do sistema, baseada em *rank*. Conforme já discutido, estas métricas não são indicadas para sistemas deste tipo. Medidas mais indicadas nesse caso são *Precision at k* ($P@k$) e *R-Precision*. O algoritmo de avaliação do SILQ, porém, considera apenas o primeiro registro Qualis retornado para realizar *match* com o trabalho sendo avaliado (apenas o mais similar). Neste caso, a avaliação usando estas duas métricas devem considerar apenas o primeiro resultado, ou seja, $P@1$ e *R-Precision* com $|R| = 1$ (sendo R o número de registros relevantes para a *query*). Em ambos os casos, para cada *match* retornado pelo sistema, temos medidas com valor igual a 0, caso o sistema não tenha avaliado corretamente o trabalho, e 1 caso tenha avaliado corretamente. Têm-se, portanto, um simples valor *booleano* indicando se houve acerto ou não, para cada *query* submetida. Considerando todo o conjunto de testes, pode-se somar o número de acertos e dividir pelo tamanho do conjunto, resultado um valor que indica a *taxa de acerto* do sistema. Este valor também é conhecido como *exatidão*² e foi a medida base escolhida para a avaliação experimental do sistema.

Outra medida utilizada em um primeiro momento foi a Média de Rank Recíproco

²O termo utilizado na literatura é *accuracy*, cuja tradução usual é *precisão*. Optou-se pelo uso do termo *exatidão*, no entanto, para evitar confusões com a métrica de precisão.

(MRR). Conforme discutido, ela é particularmente interessante para sistemas que produzem uma lista de resultados ordenados por probabilidade de correteude, e, ao contrário da exatidão, é capaz de modelar o quão bem o sistema classificou o resultado correto, mesmo quando ele não foi classificado em primeiro lugar.

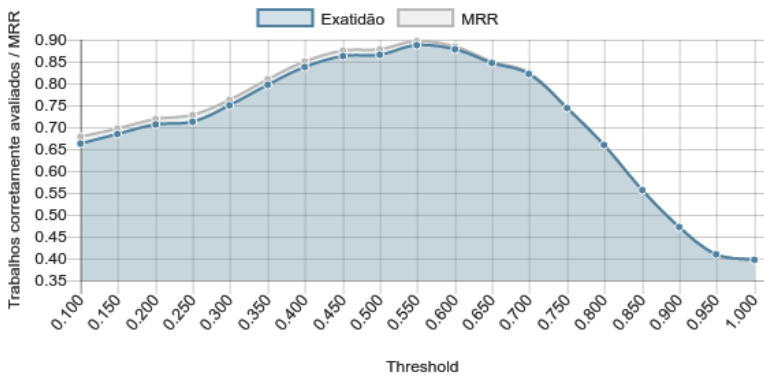
Por estas razões, as medidas de exatidão (ou taxa de acerto) e MRR foram escolhidas para as avaliações experimentais descritas nas próximas subseções.

3.3.3. Avaliação de *threshold* ideal

Um dos questionamentos levantados no início deste trabalho e que geralmente ocorre ao projetar sistemas de *data matching* baseados em similaridade, é o de qual *threshold* utilizar. Na primeira versão do sistema foi introduzido um controle de “nível de confiança” que permitia ao usuário controlar o *threshold* utilizado pelo algoritmo, conforme detalhado na Seção 2.1. O nível de confiança padrão, porém, foi fixado em 60% (equivalente ao *threshold* de valor 0.6). Este valor foi provavelmente escolhido de forma empírica pois observou-se que maximizava o número de resultados corretos, porém não foram realizados experimentos comprovando esta teoria.

Desta forma, para encontrar o valor de *threshold* ideal foi utilizado o conjunto de testes para avaliar o algoritmo inicial do SILQ 1, em um primeiro momento. O método utilizado foi o de avaliar via sistema cada uma das publicações do conjunto de testes e comparar com o resultado real, e repetir o processo variando o *threshold* a fim de observar as médias de exatidão e de rank recíproco (MRR). Os resultados foram agrupados no gráfico da Figura 3. A linha em azul claro representa a exatidão, ou seja, a taxa de trabalhos corretamente avaliados pelo sistema. A linha em cinza representa a média de rank recíproco (MRR).

Figura 3. Taxa de trabalhos corretamente avaliados e Média de Rank Recíproco (MRR) para diferentes *thresholds*



O primeiro fenômeno que observamos ao avaliar o gráfico é o ponto de máximo

por volta do valor 0.55 de *threshold*, que totaliza uma exatidão aproximada de 88%, e a tendência da exatidão baixar ao se afastar deste pico, para ambas as direções. Esse é um comportamento esperado pois valores de *threshold* baixos tendem a diminuir a exatidão do sistema por retornar resultados não relevantes para as *queries*, enquanto valores altos tendem a diminuir a exatidão por deixar de retornar resultados relevantes. Este ponto máximo trata-se, portanto, do *threshold* ideal para o caso de testes em questão.

Outra característica observada é a tendência do valor de MRR acompanhar o da exatidão, sendo sempre igual ou apenas um pouco superior em magnitude. Isso acontece pela forma com que o MRR é calculado, atribuindo valor de $1/r$ a cada avaliação, sendo r a posição em que o resultado real foi avaliado pelo sistema. Se o resultado foi corretamente avaliado, portanto, o valor de $1/1 = 1$ é atribuído ao resultado, o mesmo valor que seria atribuído à exatidão, já que o conjunto de valores possíveis para esta métrica é $\{0, 1\}$ para cada resultado (0 representando um erro e 1 representando um acerto). A semelhança dos valores, portanto, indica que houveram poucos casos em que o algoritmo retornou o resultado real em posições inferiores à primeira no *rank* de avaliação. Esta característica do valor de MRR permaneceu constante nos demais testes realizados neste trabalho, portanto omitiu-se o valor de MRR nas demais avaliações.

3.3.4. Avaliação dos algoritmos

Os algoritmos descritos na Seção 3.2 foram avaliados utilizando o mesmo processo descrito na Seção anterior. O algoritmo *trigrams* trata-se do método inicial utilizado pelo SILQ 1 e cuja análise de *threshold* ideal foi realizada na Seção anterior. O algoritmo $fb(t)$ foi testado variando t nos valores que obtiveram melhores resultados. Todas as análises foram realizadas com valor de *threshold* igual a 0.55. A Tabela 4 apresenta os resultados de cada teste.

Tabela 4. Comparação da exatidão dos diferentes algoritmos testados (utilizando *threshold* de 0.55)

| Algoritmo | Exatidão |
|-----------------------|----------------|
| <i>trigrams</i> | 88.667% |
| <i>fb(1.00)</i> | 89.667% |
| <i>fb(0.90)</i> | 90.667% |
| <i>fb(0.80)</i> | 92.667% |
| <i>fb(0.70)</i> | 92.667% |
| <i>fb(0.60)</i> | 91.000% |
| <i>query_aliasing</i> | 93.333% |

A primeira tentativa de usar *feedback* de relevância na avaliação foi com o algoritmo *fb(1.00)*, que considera os resultados informados pelo usuário quando a *query* é idêntica a algum *feedback*. Houve uma melhora na taxa de acertos, porém de forma não tão significativa.

As variações que utilizam valores menores de t , porém, obtiveram melhores resultados, por serem capazes de identificar *queries* similares aos *feedbacks* já informados pelo usuário, mesmo quando este não julgou exatamente a *query* em questão. Um exem-

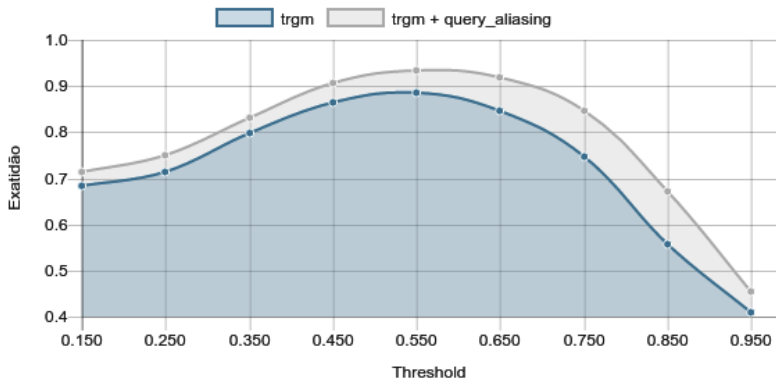
po que demonstra este fato são os nomes de eventos “*IEEE International Symposium on Computer-Based Medical Systems*” e “*27th International Symposium on ComputerBased Medical Systems (CBMS)*”, extraídos de um mesmo currículo Lattes. É fácil notar que tratam-se do mesmo evento, porém o usuário informou a sigla e o número da edição no segundo, e nenhuma destas informações no primeiro (além de não utilizar o hífen em um dos casos). Caso o usuário tenha dado *feedback* para somente um dos casos, os algoritmos $fb(t)$ com valores de t inferiores a 1.0 são capazes de utilizar o mesmo *feedback* para ambos, apesar das *queries* não serem idênticas.

Valores de t muito baixos, porém, deterioram rapidamente a taxa de acerto pois consideram *feedbacks* similares entre eventos que não tem relação. Desta forma, existe também um “valor ideal” de t , que gira em torno de 0.7 a 0.8 conforme os testes realizados.

O algoritmo $fb(t)$, porém, pode cometer “injustiças” pois considera os *feedbacks* como resultados corretos, independente dos resultados primários retornados, caso sejam superiores ao *threshold* t . O algoritmo de *query_aliasing* resolve este problema inserindo o *feedback* no *ranking* junto com os demais resultados. O melhor resultado (aquele mais similar à *query*) será utilizado, independente da técnica usada para obtê-lo. Nos testes realizados, o algoritmo de *query_aliasing* obteve a melhor taxa de acerto, com uma média de 93.3% de trabalhos corretamente avaliados.

Foi realizada uma última avaliação comparativa entre o algoritmo *trgm* inicial e o novo que utiliza *query_aliasing*. A Figura 4 mostra a taxa de acerto média para ambos os algoritmos variando o *threshold* utilizado. Nota-se que o algoritmo que utiliza *feedback* de relevância obteve melhores resultados, para qualquer *threshold* utilizado, aumentando em aproximadamente 6% a taxa de acerto. O *threshold* ideal, porém, se manteve constante em 0.55 pois é dependente da função de similaridade.

Figura 4. Comparação da taxa de acerto do algoritmo *trgm* e do *trgm* + *query_aliasing* para diferentes *thresholds*



4. Conclusões e trabalhos futuros

O SILQ é um esforço coletivo para a automatização e consequente melhoria na qualidade de gestão de grupos de pesquisa e principalmente de Programas de Pós-Graduação. A atualização tecnológica e alimentação da base de dados da ferramenta são processos que devem ser constantemente realizados para mantê-la viável aos seus usuários.

Os objetivos iniciais de refatoração arquitetural com inclusão de melhorias de interface, novas funcionalidades e criação da camada de integração REST, foram alcançados. Tanto usuários como desenvolvedores de aplicações interessados no sistema se beneficiam dessa alteração. Também foram realizadas as inclusões de controles permitindo sugestões de resultados por parte dos usuários, e desenvolvidos dois algoritmos que utilizam tais informações para melhorar a taxa de acerto do sistema.

A hipótese de pesquisa levantada se mostrou verdadeira. De fato foi possível aumentar a precisão do sistema utilizando *feedback* de relevância dos usuários. Isso se mostrou verdade por meio dos experimentos realizados e relatados neste trabalho, e com o teste dos algoritmos propostos.

A avaliação de *threshold* mostrou que o valor ideal para o SILQ é de aproximadamente 0.55, muito próximo do “nível de confiança normal” estabelecido no trabalho anterior de [de Aguiar and Costa 2015], e que a taxa de acerto para este caso é de 88%.

Os algoritmos *fb(t)* e *query_aliasing*, baseados em *feedback* de relevância e similaridade textual, foram propostos e avaliados experimentalmente. Ambos resultaram em melhoria na taxa de acerto do sistema. O algoritmo *query_aliasing* foi preferido por resultar em uma melhor precisão e ser de mais fácil implementação, e está atualmente em uso na versão 2.3 do sistema. Através da alteração do *threshold* ideal e da inclusão do algoritmo baseado em *feedback* de relevância, a taxa de acerto média do sistema aumentou de 87% para 93.3%.

Esses resultados são relevantes não só por mostrarem que a taxa de acerto média do sistema aumentou, mas por estabelecerem uma medida base para trabalhos futuros. Os algoritmos utilizados neste trabalho são estratégias simples de uso de *feedback* de relevância construídas sobre a mesma função de similaridade textual do SILQ 1, o método *trigrams*, porém outros algoritmos e estratégias diferentes podem ser testados. Em especial, ambos algoritmos propostos consideram apenas o conjunto de *feedbacks* de um usuário específico, de forma a evitar a necessidade do tratamento de *feedbacks* conflitantes ou divergentes de diferentes usuários.

Como trabalhos futuros pode-se sugerir os seguintes itens:

- Propor e analisar outros métodos de similaridade textual ou estratégias diferentes que possam melhorar ainda mais a precisão do SILQ. Alguns exemplos seriam o uso de *machine learning* para treino do algoritmo de classificação, uma estratégia conhecida por *learning to rank*; e o algoritmo de *Rocchio*, muito utilizada em sistemas que implementam *feedback* de relevância e baseado no modelo de espaço vetorial;
- Considerar *feedbacks* provindos de terceiros, ou seja, levar em conta *feedbacks* informados por outros usuários, mesmo que o atual não tenha informado *feedback* a respeito de uma *query* específica;

- Considerar nomes de eventos traduzidos. Comumente são utilizados nomes de periódicos ou eventos estrangeiros em inglês no currículo Lattes, porém em português no Qualis, ou vice-versa. Esses casos dificilmente são avaliados corretamente pelo sistema, já que o algoritmo atual é baseado em comparação por similaridade textual;
- Levantamento da produção por veículo de publicação. Considerando todos os currículos cadastrados no SILQ, seria possível levantar o número de publicações em cada periódico e evento cadastrado no Qualis;
- Gerar os valores de $I_{restrito}$ e I_{geral} automaticamente. Tratam-se de índices definidos pela CAPES e utilizados na avaliação de Programas de Pós-Graduação;
- Considerar, para fins de ponderação dos valores de estrato Qualis, os pesos considerados pela avaliação de Programas de Pós-Graduação realizada pela CAPES.

As mudanças relatadas neste trabalho já se encontram disponíveis na página oficial do SILQ: <http://silq.inf.ufsc.br/>, na versão 2.3 no momento de escrita deste trabalho, porém em constante evolução.

Referências

- CAPES (2015). Classificação da produção intelectual. <http://www.capes.gov.br/avaliacao/instrumentos-de-apoio/classificacao-da-producao-intelectual>. Acesso em 17/11/2015.
- CNPQ (2015a). Módulo produção bibliográfica. http://ajuda.cnpq.br/index.php/Modulo_Producao_Bibliografica. Acesso em 17/11/2015.
- CNPQ (2015b). Sobre a plataforma lattes. <http://www.cnpq.br/web/portal-lattes/sobre-a-plataforma>. Acesso em 17/11/2015.
- de Aguiar, F. N. M. and Costa, M. E. (2015). *SILQ - Sistema de Integração Lattes Qualis*. Florianópolis: Universidade Federal de Santa Catarina, Biblioteca Universitária.