Sistema de detecção de indícios de plágio para integração à plataforma *Open Journal Systems*

José Ferreira Dias Filho[1], Robson Silva Lopes[2]

[1]Instituto de Ciências Exatas e da Terra – Campus Universitário do Araguaia
Universidade Federal de Mato Grosso (UFMT) – Barra do Garças – MT – Brasil

[2]Instituto de Ciências Exatas e da Terra – Campus Universitário do Araguaia
Universidade Federal de Mato Grosso (UFMT) – Barra do Garças – MT – Brasil

jfilhobg@gmail.com, robsonsilvalopes@hotmail.com

Abstract. *This article discusses a technique that has proved very effective in calculating the similarity between text documents: using trigrams. His methodology shows instances of plagiarism in texts and are maximized with the help of search engines on the Internet. With this, three environments are integrated to meet the needs of an electronic magazine: OJS, the Hunter Plagiarists and search engines.*

Resumo. *Este artigo aborda uma técnica que tem se mostrado muito eficiente no cálculo da similaridade entre documentos de textos: o uso de trigramas. Sua metodologia aponta as ocorrências de plágio em textos e são maximizadas com o auxílio dos motores de busca da Internet. Com isso, são integrados três ambientes para atender às necessidades de uma revista eletrônica: o OJS, o Caçador de Plágios e os motores de busca.*

1. Introdução

O plágio pode ser visto como um problema que atinge a sociedade em geral, principalmente a comunidade científica. A Internet permite a difusão da informação com maior facilidade, e isso agrava ainda mais a questão do plágio, uma vez que muitas produções científicas são copiadas sem nenhuma autorização e/ou não têm sua autoria preservada, por exemplo.

Com o intuito de inibir a prática do plágio muitas instituições de ensino e pesquisa, por exemplo, as instituições de Ensino à Distância (EaD), têm adotado mecanismos (ferramentas) para detectá-lo em produções acadêmicas. Este cenário é abordado por Franco *et al.* (2008) e por Pertile *et al.* (2010), os quais fazem uma análise do uso de algumas ferramentas e propõem sua implantação num Ambiente Virtual de Aprendizagem (AVA).

Outro cenário diz respeito à dificuldade encontrada pelos avaliadores das revistas eletrônicas ao receberem um grande volume de produções científicas para análise no sentido de detectar plágio. Nesse sentido, faz-se necessário o desenvolvimento de uma ferramenta que facilite a detecção de plágio nessas produções.

Embora a Internet disponibilize várias ferramentas com essa finalidade, as técnicas utilizadas são bem diferentes e essas ferramentas não se integram a ambientes de indexação de documentos como o *Open Journal Systems* (OJS). Exemplos dessas ferramentas são Farejador, Plagius, Ferret, WebFerret, TurnitIn, iThenticate e outras.

O objetivo deste trabalho é implementar um *software* livre para detecção de plágios em arquivos de texto fundamentado no trabalho de Malcolm e Lane (2008), intitulado *Caçador de Plágios*, bem como de um *plug-in* que possibilite a integração do

Caçador de Plágios à plataforma de gerenciamento de periódicos livre e de código aberto denominado OJS, visto que a única ferramenta que permite essa integração é paga e possui código fechado: o iThenticate.

2. Princípio de funcionamento e arquitetura do Caçador de Plágios

Franco *et al.* (2008) citando Rosales *et al.* (2008) afirmam que ao se copiar uma obra de qualquer natureza ou partes dela sem que seja mencionada a respectiva fonte é caracterizada uma prática de plágio. A Lei nº 2.848, de 7 de dezembro de 1940, trata da violação do direito autoral, a qual define esta prática como crime.

Como percebemos esse cenário gera toda uma discussão no campo jurídico-filosófico-científico. Por isso, para evitar os plágios não só a Lei, mas também pesquisadores buscam criar soluções científicas e uma dessas soluções é o algoritmo Ferret. A base teórica do Caçador de Plágios é esse algoritmo que foi criado por Caroline Lyon e implementado por Bob Dickerson e James A. Malcolm, todos pesquisadores da Universidade de Hertfordshire no Reino Unido.

O Ferret tem funcionamento bem simplificado, uma vez que possui como entrada um conjunto de textos planos, os quais são fragmentados em *tokens* de três palavras adjacentes e classificados em ordem de série (*rank*) para calcular a similaridade entre esses textos. Os *tokens* também são conhecidos como trigramas. É importante ressaltar que os sinais de pontuação, números e símbolos em geral são descartados no processo de fragmentação dos textos. Essa etapa permitiu a Malcolm e Lane (2008) modelarem algumas funções.

A função $T(p) = p - 2$ determina a quantidade de trigramas em um texto, em que p é a quantidade de palavras contidas no texto. Já a função $P(n) = \frac{n \cdot (n-1)}{2}$ determina o total de pares de textos a serem comparados, em que n é a quantidade de textos submetidos ao algoritmo.

Os estudos de Lyon *et al.* (2001), citando as publicações de Shannon (1951), Manning e Schütze (1999), encontraram semelhanças com uma distribuição Zipfiana, em que são documentadas distribuições isoladas de palavras em inglês e outros idiomas. A distribuição Zipfiana é uma das leis utilizadas na bibliometria que está relacionada ao fato de que são usadas com muita freqüência uma quantidade pequena de palavras, embora um grande número tenha uso raro.

Lyon *et al.* (2001), afirmam que no *corpus* Brown, com aproximadamente 1 milhão de palavras, 40% ocorrem uma única vez. Os autores também mostram o percentual de trigramas que ocorrem uma única vez nos *corpora* TV News, Federalist Papers e Wall Street Journal. Seus estudos fundamentaram-se em um cenário empírico.

De acordo com a sequência em que foram citados, cada *corpus* possui aproximadamente 38 milhões, 183 mil e 985 mil palavras, e corresponde a 85%, 87% e 77% de ocorrência única de trigramas, respectivamente. Nesta perspectiva, a distribuição se torna mais dispersa para bigramas e trigramas. Isso levou Lyon *et al.* (2001) a concluírem que se uma palavra tem baixa possibilidade de ocorrer, então a possibilidade de que ela ocorra em conjunto com outras é muito menor, mesmo que os textos envolvidos tratem basicamente do mesmo assunto.

Para calcular a similaridade entre textos Lyon *et al.* (2001) fundamentam-se na publicação de Broder que busca classificar dois textos como “independentes” ou “similares” por meio do coeficiente de Jaccard.

De acordo com Meyer (2002, p. 7), o coeficiente de Jaccard é usado "na comparação entre o número de atributos comuns para um par de objetos e o número total de atributos envolvidos", cujo valor representativo está estritamente dentro do intervalo [0, 1] e é encontrado por meio da equação 1 abaixo:

$$S_{(T_1,T_2)} = \frac{a}{a+b+c}. \quad (1)$$

Nessa equação 1, $S_{(T_1,T_2)}$ representa o percentual de similaridade entre os textos T_1 e T_2, a é o número de elementos comuns aos dois conjuntos, b é o número de elementos exclusivos do primeiro conjunto e c é o número de elementos exclusivos do segundo conjunto.

Malcolm e Lane (2008) reescrevem a equação 1 transformando-a na equação 2 abaixo:

$$Similaridade = \frac{Número\ de\ trigramas\ comuns}{Número\ total\ de\ trigramas} = \frac{|A \cap B|}{|A \cup B|}. \quad (2)$$

Nessa equação 2, *A* e *B* representam dois conjuntos de trigramas que equivalem aos elementos citados na equação 1.

Nesse sentido, a eficácia do uso de trigramas é demonstrada por Lyon *et al.* (2001) ao investigar a construção de *n*-gramas. Os resultados obtidos pelos autores demonstraram que para $n > 3$ a sensibilidade para detecção de similaridades nos textos é fundamentalmente reduzida.

A figura 1 sintetiza a arquitetura do sistema.

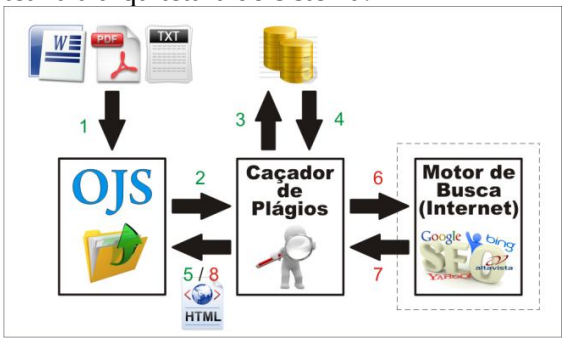

Figura 13. Arquitetura do sistema

Ao receber uma submissão de arquivo o OJS dispara a execução do Caçador de Plágios que busca e analisa os arquivos de textos armazenados localmente para a geração de um arquivo de resultados contendo o percentual de similaridade entre pares de arquivos.

Em seguida, o mesmo processo é colocado em execução com o auxílio dos motores de busca ao Caçador de Plágio, embora o programa original considere uma sutileza para a pesquisa de alguns trigramas. Como existe uma variedade de palavras muito comuns em uma dada língua (*top of words*) a busca pode retornar conteúdos irrelevantes. Devido a esse fato, todo trigrama que possuir qualquer dessas palavras deve ser descartado da busca. A cada busca de trigrama são armazenados os endereços dos dez resultados mais relevantes para o cálculo de sua frequência. Em seguida, o conteúdo dos dez endereços que possuem maior frequência são copiados para uma pasta local e submetidos ao cálculo de similaridade com o documento submetido ao OJS.

2.1. O OJS

A plataforma de gerenciamento de periódicos OJS foi desenvolvida em linguagem de programação PHP (*Hypertext Preprocessor*) com o objetivo de auxiliar na gerência de todo o processo de submissão e publicação de uma revista *online*.

Algumas dessas características dizem respeito à definição de papéis (autores, avaliadores e editores), a configuração dos requisitos, seções, processo de revisão, etc., do editor, a indexação do conteúdo, além de outras características que podem ser a ele atribuídas devido à extensão de sua aplicabilidade e à sua disponibilidade segundo a Licença Pública GNU.

Segundo PKP (2012), cerca de 11.500 instituições utilizam a plataforma OJS para publicação de revista *online.*

2.2. Implementação e integração do Caçador de Plágios ao OJS

O Caçador de Plágios foi implementado em linguagem de programação Java para integrar-se à plataforma OJS por meio de um *plug-in* que está sendo implementado em linguagem PHP e fundamentado no *plug-in* do iThenticate. O iThenticate é uma ferramenta com a mesma finalidade do Caçador de Plágios, embora seu uso esteja condicionado ao pagamento de uma mensalidade.

A Internet se insere nesse cenário quando o Caçador de Plágios utiliza-se dos motores de busca, por meio da Interface de Programação de Aplicações (*Application Programming Interface*[40] *– API*}), para localização de documentos semelhantes. Exemplos desses motores são o Google, Yahoo, Altavista, Cadê, etc.

3. Resultados e Considerações Finais

O protótipo do Caçador de Plágios possui diferenças sutis em relação ao Ferret original. Este substitui caracteres acentuados, numéricos e outros caracteres “estranhos” por um espaço em branco, o que proporciona a formação de um conjunto de trigramas completamente diferenciado em relação ao Caçador de Plágios e permite uma comparação mais precisa entre trabalhos de língua portuguesa. Uma tentativa de interfaceamento entre o motor de busca Google foi feita por meio do *Google Web Search API*, embora os dados retornados fossem ligeiramente desordenados em relação aos da página Internet (http://www.google.com.br). Isso levou-nos a utilizar o *Bing Search API* para que os resultados sejam retornados na ordem correta.

Faremos a integração entre o Caçador de Plágios e o OJS por meio de um *plug-in* que está em fase de desenvolvimento. Com isso, acreditamos alcançar melhores resultados que o Ferret ao implementarmos mais interfaces com outros motores de busca e criando uma base de dados de *top of words* da Língua Portuguesa.

Referências

FRANCO, Lucia R. H. R.; MILANEZ, José Renato Castro; SANTOS, Flávia Aparecida Oliveira. Implantação de um software detector de plágio para análise das questões dissertativas do ambiente virtual de aprendizagem Teleduc. *Revista Brasileira de Aprendizagem Aberta e a Distância*, São Paulo, v. 7, 2008. Disponível em: <http://www.abed.org.br/revistacientifica/Revista_PDF_Doc/2008/ARTIGO_17_RBAAD_2008_PESQUISA.pdf>. Acesso em: 04 jul. 2011.

[40] Conjunto de padrões e rotinas de um sistema para sua manipulação por programas de terceiros que não desejam saber os detalhes de implementação deste sistema.

LYON, Caroline; MALCOLM, James; DICKERSON, Bob. Detecting short passages of similar text in large document collections. In: ____. *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*. [S.l.]: Citeseer, 2001. p. 118–125. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.7.2630&rep=rep1&type=pdf>. Acesso em: 11 jan. 2012.

MALCOLM, James A.; LANE, Peter C. R. Efficient search for plagiarism on the web. *Proceedings of The International Conference on Technology Communication Education Kuwait 2008*, Computer Engineering Department - Kuwait University, p. 206–211, 2008. Disponível em: <https://uhra.herts.ac.uk/dspace/bitstream/2299/2549/1/kuwait-v09.pdf>. Acesso em: 04 jul. 2011.

MEYER, Andréia da Silva. *Comparação de coeficientes de similaridade usados em análises de agrupamento com dados de marcadores moleculares dominantes*. 118 f. Dissertação (Mestrado em Agronomia) — Universidade de São Paulo, Piracicaba, 2002. Disponível em: <http://www.teses.usp.br/teses/disponiveis/11/11134/tde-24072002-165250/publico/andreia.pdf>. Acesso em: 26 ago. 2011.

PERTILE, Solange de L. et al. Agente integrado a plataforma MLE-Moodle para detecção automática de indícios de plágio. In: ____. *Anais do XXI Simpósio Brasileiro de Informática na Educação*. João Pessoa: [s.n.], 2010. Disponível em: <http://www.ccae.ufpb.br/sbie2010/anais/Artigos_Resumidos_files/75380_1.pdf>. Acesso em: 04 jul. 2011.

PKP. A sample of journals using open journal systems. 2012. Disponível em: <http://pkp.sfu.ca/ojs-journals>. Acesso em: 15 jan. 2012.