Concepção de ferramenta de apoio à correção de questões dissertativas com base na adaptação de algoritmos de comparação e busca textual combinados com técnicas de pré-processamento de textos

Ricardo L. F. de Ávila[1], José M. Soares[2]

[1, 2]Departamento de Engenharia de Teleinformática (DETI)

Universidade Federal do Ceará (UFC) – Fortaleza, CE – Brasil

ricardo.lims@gmail.com, marques@ufc.br

Abstract. *This paper presents the development of a support tool for correction of essay questions in a virtual learning system, using an adaptation of string-matching algorithms combined with techniques of preprocessing texts. The tests showed, in the best case, similarity indices of 71.42%. Some answers were classified as partially correct or incorrect showing 47.62% and 18.46%, respectively. The results show that with the new mechanisms such as the use of feedback showing tips to improve responses, the algorithm can be used as a support tool for correction of essay questions.*
Keywords*: string-matching algorithms, preprocessing texts, similarity, stemming.*

Resumo. *Este trabalho apresenta o desenvolvimento de uma ferramenta de apoio para a correção de questões dissertativas em ambiente virtuais de aprendizagem, com a adaptação de algoritmos de comparação de frases combinados com técnicas de pré-processamento de textos. Os testes realizados apontaram, no melhor caso, índices de similaridade de 71,42%. As questões que foram corrigidas como parcialmente corretas ou incorretas alcançaram 47,62% e 18,46%, respectivamente. Os resultados sugerem que, com a adição de novos mecanismos, como o uso de feedback com a apresentação de dicas para melhorar as respostas, o algoritmo pode ser utilizado como ferramenta de apoio na correção de questões dissertativas.*
Palavras-Chave*: algoritmos de comparação de frases, pré-processamento de textos, similaridade, radicalização.*

1. Introdução

Nos dias atuais, os meios tecnológicos são utilizados para facilitar o processo de ensino e aprendizagem em todos os segmentos educacionais. Computadores, *softwares*, lousas digitais, *tablets* passaram a fazer parte do cotidiano dos educadores, auxiliando na sistematização do ensino, no acompanhamento dos educandos e também na correção de instrumentos de avaliação do conhecimento.

É consenso que os recursos tecnológicos atualmente disponíveis ajudam, sobremaneira, a diminuir a distância física entre educadores e educandos, possibilitando, por exemplo, o armazenamento, a distribuição e o acesso aos conteúdos da aula, as atividades avaliativas e ao registro de notas, dentre outras informações acadêmicas,

mitigando problemas de natureza organizacional escolar e colaborando para o aprendizado colaborativo.

Buscando melhorar as condições do trabalho docente no que diz respeito às correções de avaliações e de exercícios, este trabalho apresenta o desenvolvimento de uma ferramenta de apoio àqueles que utilizam o suporte de ambientes virtuais de ensino de aprendizagem (AVEAs), fazendo uso de algoritmos de comparação de frases para atestar a similaridade das respostas submetidas pelos alunos.

Essa ferramenta de apoio automatizará o processo de correção de questões dissertativas, o qual o educador avalia as respostas de cada educando, pontuará essas respostas conforme a sua proximidade do resultado esperado e possibilitará o *feedback* durante a execução e submissão das atividades, comparando a resposta inserida pelo aluno com o padrão de resposta cadastrado pelo docente, exibindo, caso seja necessário, dicas que podem ajudar a compreender melhor a questão.

Crê-se que essa ferramenta facilitará, também, a celeridade na divulgação dos resultados, com mais transparência e a qualquer tempo. Para tanto, serão utilizados algoritmos de comparação de *strings* e algumas técnicas de pré-processamento de texto com a o intuito de identificar o índice de similaridade entre a resposta do aluno e o padrão de resposta cadastrado pelo docente.

Esse artigo está disposto da seguinte forma: a próxima seção detalha o processo de reconhecimento de padrões em textos, as aplicações mais comuns quanto ao seu uso e uma breve descrição dos algoritmos utilizados de comparação de texto utilizados nesse trabalho; a seção 3 apresenta uma compilação dos trabalhos de literatura relacionados ao uso dos algoritmos de reconhecimento de padrões em textos em diferentes áreas de estudo; a seção 4 apresenta a metodologia proposta, detalhando os processos de pré-processamento de textos utilizados para melhorar o desempenho os índices de similaridade entre as frases comparadas; a seção 5 mostra a avaliação dos resultados obtidos com a solução proposta, detalhando os índices de similaridade obtidos nos testes realizados; e, por fim, na seção 6 são apresentadas as conclusões desta proposta e os trabalhos futuros.

2. Reconhecimento de Padrões em Textos

O processo de reconhecimento de padrões em textos é uma dificuldade que vem sendo estudada por parte de vários pesquisadores da área de ciência da computação. Objetiva-se definir se duas ou mais instâncias de dados (*strings*, árvores, tuplas, etc.) reproduzem o mesmo objeto do mundo real (SILVA, STASIU, ORENGO E HEUSER, 2007).

Para encontrar ocorrências de um padrão em um texto, os algoritmos chamados *string-matching* (CORMEN, LEISERSON E RIVEST, 2009), são utilizados como componente importante para a resolução de vários problemas que surgem em diversas áreas do conhecimento, dentre os quais podem ser destacados:

- análises de exames médicos para identificação da existência de tumores em tomografias, por exemplo;
- análises agrícolas, meteorológicas, geológicas, geográficas etc., de imagens capturadas por aerofotogrametria ou por satélites;
- biologia computacional para o reconhecimento de subsequências de nucleotídeos que compõem o DNA;

- processamento de textos, quando se busca encontrar as ocorrências de determinadas palavras.

Essa diversidade de utilização, porém, desencadeou algumas dificuldades de reconhecimento: a da aproximação dos padrões; a da maior subsequência repetida e a de padrões em duas ou três dimensões.

Considerando, portanto, essa variedade de dificuldades, este trabalho se restringiu, especificamente, a: (1) tratar do problema de reconhecimento de padrões unidimensionais, ou seja, de identificar as palavras iguais entre duas frases, tratado no corpo deste artigo, como reconhecimento de padrões em texto, e (2) utilizar os seguintes algoritmos para comparação dos textos:

- **Força-bruta (*brute-force*):** O algoritmo força-bruta (ZIVIANI, 2010) consiste em enumerar todos os possíveis candidatos de uma solução e verificar se cada um deles satisfaz a solução do problema. Sendo considerado um algoritmo de implementação simples, sempre encontrará uma solução, caso ela exista.
- **Boyer-Moore:** Dado um padrão P relativamente grande e um alfabeto Σ também razoavelmente grande, como normalmente acontece na maioria dos textos comuns, o algoritmo proposto por Boyer-Moore (BOYER E MOORE, 1977) pode ser considerado mais eficiente que o algoritmo Força-Bruta por realizar suas comparações da direita para a esquerda, fazendo o uso de duas novas variáveis que são usadas para calcular o novo deslocamento. Essas duas novas variáreis são conhecidas como heurística do mau-caractere e a heurística do bom-sufixo.
- **Knuth-Morris-Pratt (KMP):** O algoritmo KMP (KNUTH, MORRIS E PRATT, 1977) trabalha em tempo linear utilizando uma função prefixo que serve para extrair informações do seu próprio texto. A proposta é ignorar comparações inúteis e ser mais eficaz do que os demais algoritmos de *string-matching* como, por exemplo, o Força-Bruta.
- **Levenshtein (*edit distance*):** A distância Levenshtein (LEVENSHTEIN, 1966) ou distância de edição entre duas *strings* é dada pelo menor número de operações de edição necessárias para transformar a primeira *string* na segunda. As possíveis operações de edição são a inserção, substituição e deleção, onde a cada uma delas é atribuído um custo igual a ***1***.
- **Rabin-Karp:** O algoritmo de Rabin-Karp (KARP E RABIN, 1987) tem característica probabilística, utilizando como parâmetro duas *strings* ao qual se deseja comparar. Caso a *string* seja encontrada, a função retorna o número de caracteres iguais, caso contrário o valor ***0*** será retornado. O algoritmo surgiu com a ideia de comparar o *hash* da *string* padrão com o *hash* de *sub-strings* do texto. Em geral, a ideia é aparentemente simples, porém é necessário utilizar uma função *hash* que gere um *hash* diferente para cada *sub-string*. Essa função *hash* pode, por exemplo, utilizar os códigos *ASCII* para cada caractere, mas tomando-se cuidado com o suporte multi-linguagem.

Inicialmente, buscou-se identificar, dentre os algoritmos apresentados, aquele que exibisse os melhores resultados durante a comparação da resposta do aluno com a sua respectiva resposta padrão, fornecida pelo docente. Os algoritmos de reconhecimento de

padrões em textos sofreram basicamente uma única alteração no seu comportamento quanto à sua unidade básica de comparação. A comparação que antes era feita entre todos os caracteres de duas *strings*, tratando cada *string* como se fosse uma única palavra, passou a ser feito entre todas as palavras que compõem as duas frases, permitindo, dessa maneira, aumentar a chance de acertos e melhorar o índice de similaridade.

Os testes e a análise dos resultados realizados podem ser vistos na seção 5 deste artigo.

3. Trabalhos Relacionados

Algumas pesquisas foram realizadas para as diferentes aplicações de algoritmos de reconhecimento de padrões em texto. Pode-se citar o trabalho de Naradhipa e Purwarianti (2011) que utiliza o algoritmo de Levensthein em uma das etapas do tratamento de textos de um método para classificar os sentimentos de internautas em mídias sociais por meio do que é escrito em língua indonésia nas mensagens.

Essas mensagens foram classificadas em quatro classes: neutro (informativas, saudações etc.); pergunta; sentimento positivo e sentimento negativo. Em resultados experimentais foi obtido um índice de precisão de 86,66% utilizando o algoritmo *Support Vector Machine* (SVM) como método de classificação.

Brodanac, Budin e Jakobovic (2011) apresentam um trabalho que utiliza o algoritmo Rabin-Karp e uma maneira de paralelizá-lo, melhorando o seu desempenho em sistemas multiprocessados de descoberta de sequencia exata e todas as ocorrências de uma *string* em outra *string*. Um dos segmentos da ciência onde este tipo de pesquisa é aplicado em um nível substancial é a biologia, especialmente no segmento sobre as cadeias de DNA. Dentre os diferentes algoritmos que poderiam ser utilizados para a solução apresentada, o de Rabin-Karp mostrou ser o mais eficaz.

Um sistema de anotação semiautomático para habilitar pesquisa semântica é apresentado por Liu, Chen, Jaine Chen (2009), capturando os dados textuais web usando o algoritmo Knuth-Morris-Pratt (KMP). A solução proposta visa obter melhores resultados nas pesquisas textuais, uma vez que a notação semântica reduz a quantidade de termos inadequados obtidos. Os resultados alcançados apresentaram uma melhora significativa nas buscas realizadas em sistemas como o Google, Yahoo e Bing em comparação ao simples uso de palavras-chaves.

Wang, ZhuGe e Wang (2010) utilizam uma versão adaptada do algoritmo Boyer-Moore no desenvolvimento de um sistema de auditoria e segurança de rede. O trabalho proposto combina os pontos fortes do algoritmo em sua versão clássica, melhorando, durante o processo de comparação, a distância dos saltos e a sua velocidade em tempo de execução. Essas mudanças, de acordo com os resultados apresentados, aceleraram a velocidade de comparação e melhoraram a eficiência do acerto, alcançado, no melhor caso, o índice de 94,03% de acerto na identificação das regras utilizadas.

Esses trabalhos contemplam importantes contribuições para o uso de algoritmos de comparação e busca textual, porém nenhum deles trata especificamente da solução proposta no presente artigo e que seja capaz de reunir as seguintes características: (i) utilizar técnicas de pré-processamento de textos; (ii) efetuar a comparação individual de todas as palavras entre duas *strings*; (iii) ser utilizado para a correção de questões

dissertativas, três características essenciais para a melhora dos resultados obtidos, conforme pode ser visto na seção 5 deste artigo.

4. Metodologia Proposta

Além da adaptação dos algoritmos citados na seção 2, foram utilizadas algumas técnicas de pré-processamento de textos com a finalidade de melhorar o desempenho dos índices de similaridade entre as *strings* comparadas, que foram:

- **a remoção de caracteres inválidos**: caracteres como aspas, colchetes, parênteses, dentre outros, foram retirados;
- **a exclusão de palavras repetidas**, para evitar a comparação desnecessária de uma palavra duplicada várias vezes;
- **a aplicação do *uppercase***, para impedir que palavras com o mesmo significado iniciadas com caracteres em maiúsculo venham a ser diferenciadas de uma palavra semelhante iniciada com caracteres em minúsculo;
- **a retirada de *stopwords***, para prevenir que palavras como artigos, advérbios, pronomes, preposições, dentre outras, irrelevantes para a finalidade deste trabalho, venham a influenciar de forma negativa a comparação das frases. Mais detalhes sobre esta técnica podem ser vistos em (LUHN, 1966);
- **a substituição de caracteres acentuados**. Os caracteres acentuados foram substituídos por seus respectivos caracteres sem acentuação;
- ***a stemming***: para reduzir as variantes morfológicas das palavras, como formas singulares, plural e conjugações verbais, para a sua raiz ou radical, retirando sufixos e prefixos (MOENS, 2000). Para o processo de aplicação da técnica de *stemming*, foi utilizado o algoritmo proposto por Porter (1980) e adaptado para a língua portuguesa por Orengo e Hyuck (2001) e a implementação *opensource* do algoritmo de Orengo para o idioma português, denominado PTStemmer (PTSTEMMER, 2012).

Para identificar dentre os algoritmos considerados aquele que possibilitou os melhores resultados nos testes de similaridade entre frases, foi desenvolvida uma ferramenta que possibilitou testar individualmente a utilização de todas as técnicas de pré-processamento de texto, permitindo, ainda, a seleção de um conjunto delas, além da sua ordem de execução. Dessa forma, foi possível verificar os percentuais de acerto com o uso ou não de uma técnica específica, evitando, que uma técnica causasse impacto sobre outra. Por exemplo, caso a técnica de substituir caracteres acentuados seja realizada antes da técnica de *stemming*, os resultados poderão ser comprometidos, visto que a retirada da acentuação irá prejudicar as regras de redução das palavras para a sua raiz ou radical.

5. Avaliação da Solução Proposta

Foi utilizada a amostra de 3 questões dissertativas de uma avaliação realizada por 10 estudantes, totalizando, então, 30 respostas para análise. Foram observadas as respostas dadas nas questões, todas devidamente digitadas e salvas em um arquivo de texto. As questões foram corrigidas e pontuadas pelo docente da disciplina, permitindo comparar a sua proximidade com os resultados obtidos na solução proposta. Para comparar as respostas, o docente da disciplina forneceu uma resposta padrão para cada questão.

Inicialmente foram utilizados apenas os algoritmos de reconhecimento de padrões em textos adaptados para a solução proposta. De posse do padrão de resposta de cada questão dissertativa e das respostas individuais dos alunos, foram realizadas as comparações para identificar o algoritmo com melhor resultado.

Verificando os resultados apresentados na tabela 1, percebem-se que, com exceção do algoritmo Levenshtein, as taxas de similaridade de cada resposta fornecida pelos alunos, para a mesma questão, foram semelhantes. Isso se deve ao fato de a adaptação dos algoritmos seguirem a mesma regra, onde cada palavra de uma frase é comparada com todas as palavras da outra frase. Ao encontrar uma palavra igual, o algoritmo acrescenta um acerto (*match*) que posteriormente será utilizado para calcular o percentual de similaridade entre as frases.

O algoritmo Levenshtein teve o pior resultado para a adaptação proposta por possuir como característica principal a retirada dos espaços em branco entre palavras, inviabilizando o uso desse algoritmo para a solução proposta.

Tabela 1: Resultados obtidos com os algoritmos de comparação e busca textual.

Questão 01	Nota atribuída pelo docente	Boyer-Moore		Rabin-Karp		Levenshtein		Brute Force		Knuth-Morris-Pratt	
		Sim.	ms	Sim.	ms	Sim.	ms	Sim.	ms	Sim.	ms
Aluno 01	1,00	44,58	1,28	44,58	4,23	30,65	9,65	44,58	2,05	44,58	5,87
Aluno 02	1,00	43,82	1,53	43,82	3,75	30,46	1,45	43,82	0,63	43,82	3,08
Aluno 03	1,00	44,94	1,45	44,94	3,39	31,67	1,46	44,94	0,41	44,94	1,66
Aluno 04	1,00	46,07	0,75	46,07	3,15	32,36	1,34	46,07	0,56	46,07	1,97
Aluno 05	1,00	41,57	1,36	41,57	4,66	29,60	1,25	41,57	0,72	41,57	3,40
Aluno 06	1,00	47,19	1,55	47,19	7,50	28,40	1,46	47,19	0,49	47,19	2,69
Aluno 07	1,00	44,94	1,28	44,94	5,51	30,81	2,81	44,94	0,46	44,94	2,72
Aluno 08	0,50	24,72	0,69	24,72	1,55	13,43	0,41	24,72	0,26	24,72	2,98
Aluno 09	0,25	21,35	0,66	21,35	2,31	12,74	0,51	21,35	0,39	21,35	4,78
Aluno 10	0,75	42,70	0,76	42,70	2,54	24,96	1,73	42,70	0,53	42,70	2,54

Fonte: elaborada pelo autor.

Foi constatado, ainda, que as respostas dos alunos 01 a 07, consideradas pelo docente da disciplina corretas, obtiveram índices de similaridades entre 41,57 e 47,19%. A resposta do aluno 08, que recebeu meio ponto, alcançou 24,72% de similaridade. Já o aluno 09, que recebeu um quarto da pontuação, obteve 21,35% de similaridade. Com exceção do aluno 10, com um índice de 42,70% de similaridade, podendo ter recebido a pontuação máxima, podemos aferir que os índices alcançados apresentam um padrão, em que é possível indicar a possibilidade a adoção dos algoritmos de comparação e busca textual para a correção de questões dissertativas.

Após a confirmação que os algoritmos de reconhecimento de padrões em textos adaptados para a solução proposta obtiveram os mesmos resultados comparados entre si, foram realizadas simulações utilizando as técnicas de pré-processamento de textos. Os testes individuais de cada técnica de pré-processamento em conjunto com o algoritmo Boyer-Moore foram realizados e os resultados obtidos mostram que o uso de algumas técnicas de pré-processamento podem melhorar as comparações entre frases.

O critério para a seleção do algoritmo Boyer-Moore foi devido ao seu baixo custo computacional, obtendo tempos de execução na faixa de 0,66 a 1,53 milissegundos, conforme pode ser visto na figura 2. Além disso, por efetuar as suas comparações da

direita para a esquerda, pode ser considerado mais eficiente que o algoritmo Força-bruta, que também obteve baixos índices de custo computacional (BOYER E MOORE, 1977).

Conforme pode ser visto na tabela 2, cada técnica de pré-processamento de texto resultou em diferentes índices de similaridade. Os melhores resultados obtidos foram com as técnicas Stemmer Orengo e aplicação do *uppercase*, que alcançaram 67,23% de similaridade entre frases com a resposta do aluno 10. Os demais resultados, em ordem decrescente de similaridade são: remoção de caracteres inválidos com 66,39%, remoção de acentuação com 62,18%, remoção de *stopwords* com 47,30% e remoção de palavras repetidas com 43,24%. É importante reforçar que as simulações foram feitas entre o padrão de resposta fornecido pelo docente da disciplina e as respostas individuais dadas pelos alunos.

Tabela 2: Resultados Iniciais Obtidos com o Algoritmo Boyer-Moore.

Questão 03	Nota atribuída pelo docente	Boyer-Moore		Boyer-Moore + Rem. acentuação		Boyer-Moore + Rem. caracteres inválidos		Boyer-Moore + Rem. palavras repetidas		Boyer-Moore + Rem. stopwords		Boyer-Moore + Stemmer Orengo		Boyer-Moore + Aplicar upercase	
		Sim.	ms	Sim.	ms	Sim.	ms	Sim.	ms	Sim.	ms	Sim.	ms	Sim.	ms
Aluno 01	1,00	44,54	1,03	46,22	0,76	49,58	0,47	28,38	0,30	20,27	0,33	51,26	0,42	49,58	0,34
Aluno 02	2,00	43,70	1,52	45,38	0,53	46,22	0,39	25,68	0,26	29,73	0,25	49,58	0,40	49,58	0,43
Aluno 03	2,00	49,58	1,22	49,58	0,56	52,10	0,63	32,43	0,30	35,14	0,21	53,78	0,45	52,94	0,37
Aluno 04	2,00	48,74	1,37	50,42	0,71	51,26	0,37	29,73	0,28	32,43	0,24	55,46	0,42	53,78	0,43
Aluno 05	2,00	41,18	0,91	41,18	0,59	42,86	0,82	21,62	0,19	21,62	0,19	44,54	0,32	47,06	0,28
Aluno 06	2,00	47,06	0,75	47,06	0,45	49,58	0,45	27,03	0,21	35,14	0,16	54,62	0,23	52,94	0,23
Aluno 07	2,00	55,12	1,44	55,12	0,73	58,27	0,86	38,75	0,73	37,66	0,49	61,42	0,77	58,27	0,68
Aluno 08	2,00	47,06	0,62	47,06	0,59	50,42	0,35	28,38	0,23	33,78	0,22	50,42	0,22	50,42	0,21
Aluno 09	2,00	31,09	0,61	35,29	0,33	34,45	0,36	18,92	0,24	17,57	0,18	36,97	0,25	37,82	0,23
Aluno 10	2,00	62,18	0,86	62,18	0,84	66,39	0,47	43,24	0,39	47,30	0,30	67,23	0,35	67,23	0,29

Fonte: elaborada pelo autor.

Em todos os testes realizados foi averiguado que o uso das técnicas de remoção de *stopwords* e remoção de palavras repetidas não ajudaram a melhorar os resultados de índice de similaridade. Na verdade, os deixou piores, principalmente, ao se comparar com os valores alcançados somente com o algoritmo Boyer-Moore. Diante disso, decidiu-se não utilizar essas técnicas nas rodadas de testes seguintes.

Empós, optou-se por aplicar o uso de três ou quatro técnicas de pré-processamento de texto em conjunto em busca de melhores resultados. Arquitetou-se um plano de testes que garantiu o uso das técnicas de processamento, em grupos de no máximo quatro técnicas, levando-se em consideração, ainda, a ordem de execução.

Os testes foram feitos em todas as ordens possíveis[1], sendo apresentados na figura 1 os melhores resultados obtidos. Verificou-se que a ordem das técnicas influenciou nos resultados.

[1] Ao todo foram realizados 30 testes para esgotar todas as possibilidades possíveis para um conjunto de três ou quatro técnicas. O cálculo para comprovar a quantidade total foi [(4!) + (3!) = 30]

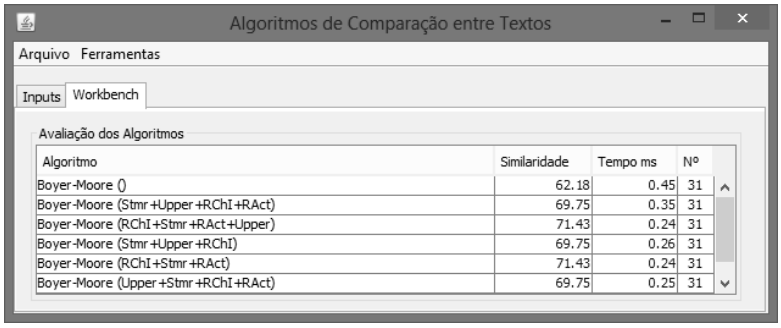

Figura 1: Resultados Finais Obtidos com o Algoritmo Boyer-Moore

Comparando o resultado inicial de 62,18%, utilizando somente o algoritmo Boyer-Moore, com o resultado final de 71,43%, com o uso das técnicas de pré-processamento de texto, foi constatada uma melhora nos índices de similaridade obtidos, sugerindo que o algoritmo proposto pode ser utilizado para apoiar a correção de questões dissertativas. Dentre as sequências testadas, a composta por remoção de caracteres inválidos, stemmer Orengo e remoção de acentuação foi a que exigiu o menor custo computacional.

Confrontando os índices de similaridade obtidos com a ferramenta proposta com os *scores* (pontos) alcançados pelos alunos nas questões utilizadas para o teste, foi verificado que os alunos que obtiveram a pontuação máxima na questão (2 pontos) também tiveram altos índices de similaridade, alcançando, no melhor caso, 64,76%. No caso dos alunos que tiraram 1 (um) ponto na questão, o índice de similaridade foi, no melhor caso, de 47,62%. Já os alunos com a questão corrigida como errada não passaram dos 18,46% de índice de similaridade. Esses índices podem ser visualizados na tabela 3.

Tabela 3: Resultados Finais Obtidos com o Algoritmo Boyer-Moore.

Questão 02	Nota atribuída pelo docente	Boyer-Moore + Rem. caracteres inválidos + Stemmer Orengo + Rem. acentuação	
		Sim.	ms
Aluno 01	1,00	43,81	0,38
Aluno 02	0,00	10,77	0,22
Aluno 03	0,00	16,92	0,27
Aluno 04	0,00	18,46	0,19
Aluno 05	0,00	13,85	0,27
Aluno 06	0,00	12,31	0,12
Aluno 07	2,00	64,76	0,32
Aluno 08	2,00	56,19	0,33
Aluno 09	1,00	47,62	0,30
Aluno 10	0,00	15,24	0,11

Fonte: elaborada pelo autor.

6. Conclusões e Trabalhos Futuros

A principal contribuição deste trabalho é avaliar o uso de algoritmos de comparação e busca textual combinados com técnicas de pré-processamento de textos, identificando a possibilidade de utilizar essa solução em um AVEA. Uma ferramenta dessas poderá automatizar parcialmente o processo de correção de questões dissertativas, possibilitando, ainda, o *feedback* durante a execução e submissão das atividades, comparando a resposta inserida pelo aluno com o padrão de resposta cadastrado pelo

docente, exibindo, caso seja necessário, dicas que podem ajudar a compreender melhor a questão.

Os resultados alcançados durante os testes, comparando respostas reais de uma avaliação com o padrão de resposta fornecido pelo docente da disciplina, foram promissores, chegado ao índice de similaridade de 71,43%, no melhor caso, em uma questão considerada correta. Além disso, as respostas que foram corrigidas como parcialmente corretas alcançaram, no melhor caso, 47,62% de índice de similaridade e as respostas corrigidas como incorretas não passaram de 18,46% de índice de similaridade.

Os índices de similaridade alcançados indicam que a solução proposta pode ser utilizada como ferramenta de apoio na correção de questões dissertativas. Porém, para garantir que as respostas dadas fiquem mais próximas do padrão de resposta esperado, a opção de oferecer dicas quando o índice de similaridade estiver abaixo de um percentual pré-determinado, poderão melhorar o entendimento do enunciando da questão e também a resposta dada pelo aluno. Além disso, poderão ser feitos levantamentos da quantidade de tentativas que o aluno realizou para alcançar um bom índice de similaridade. Pode-se, ainda, permitir que o tutor / docente adicione novas respostas padrões.

É importante esclarecer que a proposta apresentada não contempla a análise semântica ou conteúdo das frases, utilizando apenas as palavras de cada frase para avaliar a proximidade da resposta dada pelo aluno com o padrão de resposta fornecido pelo docente. Em alguns casos, terão que ser criadas regras que identifiquem, por exemplo, se é esperado uma negação ou uma afirmação dentro da resposta, podendo, para isso, serem cadastradas ou destacadas as palavras-chave aguardadas para uma resposta aceitável.

Como sugestão de melhorias para a solução proposta, visando aperfeiçoar, dentre outros itens, os índices de similaridade, a sua integrabilidade e a metodologia de ensino e aprendizagem adotada, pode-se

- possibilitar o cadastro de palavras-chave, aumentando o índice de similaridade e a identificação de termos importantes necessários para a validação da resposta fornecida pelo aluno;
- desenvolver a ferramenta proposta de forma que a mesma seja oferecida em uma arquitetura orientada a serviços (*Service Oriented Architecture* - SOA), possibilitando a sua integração com o uso de *Web Services* (WS) a um AVA; (PAPAZOGLOU, 2007; SAKAI, 2012);
- aplicar um questionário de pesquisa quantitativa com o objetivo de investigar, junto aos docentes, as seguintes representações consideradas relevantes para o processo de avaliação da solução proposta: a correção de questões dissertativas em seu modelo tradicional de aplicação (com o uso de papel e caneta, por exemplo); a confecção de padrão de resposta; a ordem de correção das questões; o momento de correção das questões; a formulação do enunciado das questões; a resposta esperada ou as partes essenciais da resposta esperada; os critérios de atribuição dos pontos, valores atribuídos a diferentes níveis de resposta; e permissão ou não de falhas na construção das respostas (ALVES-MAZZOTTI, 2002).

Referências

ALVES-MAZZOTTI, A. J. e GEWANDSZNAJDER, F. (2002) **O método nas ciências naturais e sociais** - Pesquisa quantitativa e qualitativa. São Paulo: Pioneira Thompson Learning, pp. 147-176.

BOYER, R.S. E MOORE, J.S. (1977) **A Fast String Searching Algorithm**. Comm. ACM (New York, NY, USA: Association for Computing Machinery) 20 (10): 762–772.

BRODANAC, P.,BUDIN, L. e JAKOBOVIC, D. (2011) **Parallelized Rabin-Karp method for exact string matching**. Information Technology Interfaces (ITI), Proceedings of the ITI 2011 33rd International Conference, pp.585-590, 27-30 June 2011.

CHARRAS, C. e LECROQ, T. (1998) **Sequence comparison**. Disponível em: <http://www-igm.univ-mlv.fr/~lecroq/seqcomp/index.html>. Acesso em: 15 Mar. 2012.

CORMEN, T., LEISERSON, C. e RIVEST, R. (2009) **Introduction to algorithms**. 3rd Edition, The MIT Press, Pages 985-1002.

FRANCO, L., MILANEZ, J. e SANTOS, F. (2008). **Implantação de um software detector de plágio para análise das questões dissertativas do ambiente virtual de aprendizagem TelEduc**. Brazilian Review of Open And Distance Learning, Volume 7.

GROSSI, M. A. (2005) **Avaliando a Performance das Funções de Similaridade através da Revocação e Precisão.** Projeto de Diplomação (Bacharelado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre.

KARP, R.M. E RABIN, M.O. (1987) **Efficient randomized pattern-matching algorithms**. IBM Journal of Research and Development 31 (2): 249–260.

KNUTH, D. MORRIS, J.H. E PRATT, V.J. (1977) **Fast pattern matching in strings**. SIAM Journal on Computing 6 (2): 323–350.

LEVENSHTEIN, V. I. (1966) **Binary codes capable of correcting deletions, insertions and reversals.** Soviet Physics Doklady, [S.l.], v. 10, n. 8, Pages 707-710.

LIU, C.-H., CHEN, H.-C., JAIN, J.-L. E CHEN, J.-Y.(2009) **Semi-Automatic Annotation System for OWL-Based Semantic Search Complex, Intelligent and Software Intensive Systems**, 2009.CISIS '09.International Conference, pp.475-480, 16-19 March 2009.

LUHN, H. P. (1966). **Keyword-in-context index for technical literature**. American Documentation, 11(4):288–295.

MANBER, U. (1989) **Introduction to Algorithms** : a Creative Approach. Addison-Wesley Pub Co, 1st Edition.

MOENS, M.-F. (2000). **Automatic Indexing and Abstracting of Document Texts**. (The Kluwer International Series on Information Retrieval 6). Kluwer Academic Publishers: Boston, Pages 81-84

NARADHIPA, A.R. E PURWARIANTI, A. (2011) **Sentiment classification for Indonesian message in social media**. Electrical Engineering and Informatics (ICEEI), 2011 International Conference, pp.1-4, 17-19 July 2011.

ORENGO, V. M. E HUYCK, C. (2001) **A Stemming Algorithm for the Portuguese Language**. Proceedings of SPIRE'2001 Symposium on String and Information Retrieval, Laguna de San Raphael, Chile, November 2001.

PAPAZOGLOU, M. P. e HEUVEL, W.-J. (2007) **Service Oriented Architectures: approaches, technologies and research issues**. The VLDB Journal, v. 16, p. 389-415.

Porter, M.F. (1980). **An Algorithm for Suffix Stripping**. Program, 14(3): 130–137

PTSTEMMER – **A Java Stemming Toolkit for the Portuguese Language**. Disponível em: <http://code.google.com/p/ptstemmer>. Acessado em 11 de Abril de 2012.

SAKAI – **Collaborative and Learning Environment for Education**. Disponível em: <https://confluence.sakaiproject.org/display/WEBSVCS/Home>. Acessado em 16 Abril de 2012.

SEDGEWICK, R E WAYNE, K. (2011) **Algorithms**. Addison-Wesley Professional, 4th edition.

SILVA, R., STASIU, R., ORENGO, V. E HEUSER, C. (2007) **Measuring quality of similarity functions in approximate data matching**. Journal of Informetrics, Volume 1, Issue 1, January 2007, Pages 35–46.

WANG, X., ZHUGE, B. E WANG, W. (2010) **A BM Algorithm Oriented on Network Security Audit System.** e-Business and Information System Security (EBISS), 2nd International Conference, pp.1-4, 22-23 May 2010.

ZIVIANI, N. (2010) **Projeto de Algoritmos com Implementações em Pascal e C**. São Paulo: Pioneira Thomson Learning, 3ª Edição.