

Uso de técnicas de pré-processamento textual e algoritmos de comparação como suporte à correção de questões dissertativas: experimentos, análises e contribuições

Ricardo L. F. de Ávila¹, José M. Soares²

¹ Centro Universitário Christus (Unichristus)

² Departamento de Engenharia de Teleinformática (DETI) - Universidade Federal do Ceará (UFC)

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

Resumo. Este trabalho apresenta um estudo sobre a adaptação de algoritmos de comparação textual combinados com técnicas de pré-processamento de textos para a correção de avaliações e exercícios dissertativos. Buscando contornar situações específicas de falso negativo e falso positivo, foram propostas algumas técnicas auxiliares como contribuição deste trabalho. Após a análise dos experimentos realizados, os resultados de índice de similaridade entre respostas indicam o uso da solução como suporte a correção de questões discursivas, podendo, ainda, ser aplicado na detecção de plágio e ser integrado a um ambiente virtual de ensino e aprendizagem.

Abstract. This paper presents a study of an adaptation of string-matching algorithms combined with techniques of preprocessing texts for correction of essay questions and evaluations. Seeking to neutralize specific situations of false negative and false positive, some techniques have been proposed as auxiliary contribution of this work. After analyze the experiments, the results of similarity index between responses indicates the use of the solution to support the correction of essay questions, and may also be applied in the detection of plagiarism and be integrated to a learning management system.

1. Introdução

Diferentes meios tecnológicos são utilizados para facilitar o processo de ensino e aprendizagem em diversos segmentos educacionais. Os computadores e softwares passaram a fazer parte do cotidiano de educadores em sala de aula, auxiliando na sistematização e acompanhamento dos educandos na resolução de diferentes métodos de avaliação do conhecimento.

Os recursos tecnológicos atualmente disponíveis ajudam a diminuir as dificuldades estabelecidas pela distância física entre educadores e educandos, possibilitando, por exemplo, o armazenamento, distribuição e acesso a notas de aula, atividades avaliativas e registro de notas, dentre outros tipos de informações. O simples uso de tais ferramentas pode mitigar os problemas de natureza organizacional e até mesmo colaborar com o aprendizado colaborativo.

Buscando melhorar as condições de ensino e aprendizagem em disciplinas com grande número de alunos que utilizam o suporte de um Ambiente Virtual de Ensino e Aprendizagem (AVEA), este trabalho apresenta um estudo sobre a adaptação de algoritmos de comparação textual combinados com técnicas de pré-processamento de textos para a correção de avaliações e exercícios dissertativos. Esse fato vem ao encontro do que afirmam Tavares *et al.* [2010], que reforçam que “especialmente no caso de turmas numerosas, a automatização ou semi-automatização da correção de exercícios propostos e o rastreamento das atividades realizadas pelos alunos por intermédio do computador podem oferecer um importante instrumento de análise e avaliação”.

O estudo realizado neste trabalho, faz uso de algoritmos de comparação textual para verificar a similaridade das respostas submetidas pelos alunos em relação a respostas padronizadas. Além de contribuir com o processo de correção de questões dissertativas, em que, no modelo tradicional, o educador deve avaliar a resposta de cada educando e pontuá-la conforme a sua proximidade do resultado esperado, ferramentas construídas por essa abordagem, por hipótese, podem, ainda, fornecer *feedback* durante a execução e a submissão das atividades, comparando a resposta inserida pelo aluno com padrões de resposta cadastrados pelo docente, exibindo, caso seja necessário, sugestões que podem ajudar a compreender melhor a questão.

2. Trabalhos Correlatos

O desafio deste trabalho encontra-se no desenvolvimento de uma ferramenta que sirva de apoio e facilite o processo de avaliação de questões dissertativas por parte do docente. A correção manual de questões dissertativas é uma tarefa que requer muito tempo e a disponibilidade da maioria dos educadores é limitada. Além disso, durante a correção de grande volume de provas, dificilmente o educador consegue manter o mesmo nível de rigidez para todas as questões.

Estudos nessa área tiveram seu início em meados de 1960, quando o sistema *Project Essay Grader* (PEG) foi desenvolvido para avaliar pequenas questões dissertativas [Page, 1967]. Com o advento de novas técnicas como Processamento de Linguagem Natural (PLN) e Extração da Informação (EI) na década de 90, foram restabelecidas as pesquisas e novas ferramentas foram desenvolvidas, a exemplos de E-Rater e IEA [Hearst, 2000].

Sphair [2006] desenvolveu trabalho de pesquisa relativo ao AVEA Eureka, verificando seu impacto no ensino aprendizagem, além de aferir aspectos sobre o seu potencial e a sua mediação nesse processo, e as metodologias que deverão ser implementadas pelos docentes para a utilização deste ambiente virtual como apoio às atividades no ambiente presencial. O ambiente disponibiliza, dentre outros instrumentos avaliativos, o uso de questões dissertativas, para os quais devem ser escritos o enunciado da questão e um conjunto de respostas possíveis. Essas respostas são apenas para facilitar o processo de correção, uma vez que não há a correção automática pelo sistema para este tipo de questão.

O uso de técnicas de pré-processamento de texto para melhorar os resultados de classificação e a mineração textos vem sendo explorado há alguns anos. Os trabalhos relacionados a seguir foram desenvolvidos nesse contexto.

Rodrigues [2012] desenvolveu em seu trabalho um sistema que avalia respostas curtas de questões dissertativas. Para isso, o sistema classifica a pergunta do professor por tipo (quem, o quê, onde, qual, quando e como) para em seguida, de acordo com o tipo de questão, definir pontuações associadas à resposta. Diferentes técnicas de pré-processamento foram utilizadas com o objetivo de melhorar o texto antes de efetuar as comparações. Os resultados obtidos mostraram que há uma boa correlação entre a avaliação do instrutor e a avaliação efetuada pelo sistema, alcançando 0,78 de correlação entre os escores médios fornecidos pelos professores.

O trabalho de [Azevedo, Behar e Reategui, 2011] apresenta os resultados de uma análise qualitativa das contribuições textuais registradas por alunos em fóruns de discussão. O estudo aplica técnicas de mineração de textos, como a remoção de *stopwords*, e grafos para identificar quais alunos redigiram mensagens relevantes e quais não o fizeram. Os resultados obtidos sugerem que é possível ter subsídios para avaliar os discentes que necessitam de maior auxílio e motivá-los para discutir os conceitos importantes que fazem parte do tema em debate.

Neste trabalho, em contribuição aos que foram apresentados nesta seção, propõe-se uma ferramenta que seja capaz de reunir as seguintes características: (i) adaptar o comportamento de algoritmos de busca e comparação textual, permitindo que as comparações sejam realizadas entre todas as *strings* que compõem duas frases; (ii) utilizar técnicas de pré-processamento de textos para melhorar os resultados das comparações; (iii) permitir a ordenação das técnicas de pré-processamento para avaliar os ganhos e perdas de índice de similaridade; (iv) apresentar o quantitativo de cada técnica de pré-processamento, o índice de similaridade obtido, o tempo de execução e o resultado da transformação das duas frases. Essas características foram essenciais para a melhora dos resultados obtidos, conforme pode ser visto na seção 4 deste artigo.

3. Algoritmos de comparação e técnicas de pré-processamento textual

Para os propósitos deste trabalho, os algoritmos Força-Bruta [Ziviani, 2010], Boyer-Moore [Boyer e Moore, 1977], KMP [Knuth, Morris e Pratt, 1977], Levenshtein (*edit distance*) [Levenshtein, 1966] e Rabin-Karp [Karp e Rabin, 1987], chamados *string-matching* [Cormen, Leiserson e Rivest, 2009], foram adaptados em seu comportamento quanto à 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.

Além da adaptação de algoritmos de comparação, foram utilizadas técnicas de pré-processamento de textos com a finalidade de melhorar o desempenho dos índices de similaridade entre as frases comparadas. A descrição dessas técnicas é enumerada a seguir, sendo cada uma delas identificada por uma sigla de maneira a facilitar a descrição dos experimentos discutidos neste trabalho:

- Substituir caracteres acentuados (RAP): Caracteres acentuados são substituídos por seus respectivos caracteres sem acentuação.
- Remover caracteres inválidos (RCI): Caracteres como aspas, colchetes, parênteses, dentre outros, são retirados.
- Remover palavras repetidas (RPR): Evitar a comparação desnecessária de uma palavra duplicada várias vezes.
- Aplicar *uppercase* (AUC): Evitar que palavras com o mesmo significado iniciadas com caracteres em maiúsculo venham a ser diferenciadas de uma palavra semelhante iniciada com caracteres em minúsculo.
- Remover de *stopwords* (RSW): Palavras como artigos, advérbios, pronomes, preposições, dentre outras, irrelevantes para a finalidade deste trabalho, são retiradas afim de evitar que venham a influenciar de forma negativa a comparação das frases. Mais detalhes sobre esta técnica podem ser vistos em [Luhn, 1966].
- *Stemming* (SOR): Serve para reduzir as variantes morfológicas das palavras, como formas singulares, plural e conjugações verbais, para a sua raiz ou radical, retirando sufixos e prefixos. Outra informação sobre esta técnica e como utilizá-la em outras situações estão em [Moens, 2000].

Para avaliar a capacidade dos algoritmos de busca e comparação e das demais técnicas de pré-processamento, foi desenvolvida uma ferramenta utilizando a linguagem Java que pode ser configurada a partir de arquivos codificados em XML. Com isso, foi possível testar individualmente a combinação de múltiplas técnicas de pré-processamento de texto, além de verificar eventuais ganhos ou perdas de eficiência com a modificação na ordem de aplicação de algumas delas. Dessa forma, foi possível verificar os percentuais de similaridade com o uso ou não de cada técnica, avaliando, também, o impacto de uma sobre outra.

A figura 1 exibe a interface da ferramenta desenvolvida, apresentando a tela de seleção dos algoritmos e das técnicas de pré-processamento de texto.

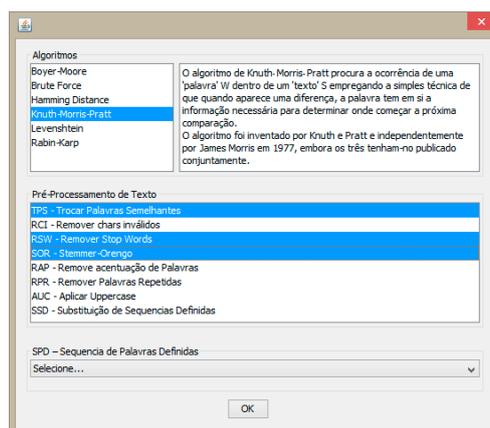


Figura 1: Tela de seleção de algoritmo e configuração de técnicas de pré-processamento.

Através da ferramenta desenvolvida, foram realizados testes de similaridade nas seguintes condições:

- Somente com algoritmo de busca e comparação
- Com algoritmo de busca e comparação e técnicas de pré-processamento de texto encontradas na literatura
- Com troca de palavras semelhantes
- Com sequência de palavras definidas
- Com realimentação de respostas

Os três últimos itens da lista de testes são contribuições deste trabalho para mitigar problemas não tratáveis com as técnicas de pré-processamento ou com os algoritmos de busca. A descrição das técnicas utilizadas é apresentada a seguir:

- Trocar palavras semelhantes (TPS): Palavras semelhantes ou sinônimos são substituídos para um único termo correspondentes com o intuito de melhorar os índices de similaridade. Isso ocorre quando duas palavras, apesar de não possuírem, em sua essência, significados idênticos, no contexto, apontam exatamente para o mesmo objeto manifestando uma relação de equivalência. O objetivo é montar uma base de dados contendo as palavras comumente utilizadas no domínio avaliado para melhorar o desempenho dos índices de similaridade entre as frases comparadas.
- Substituir sequencia pré-definida (SSD): Essa nova técnica, não encontrada na literatura e sugerida nesse trabalho, foi desenvolvida, inicialmente, para verificar se a inversão de palavras dentro de uma sentença, como por exemplo, “*emissor para o receptor*” e “*receptor para o emissor*” poderia influenciar no índice da taxa de similaridade. O seu funcionamento consiste em percorrer uma frase (sentença de palavras) em busca de uma sequencia de palavras pré-definidas e, caso encontre, a substituir por outra sequencia.
- A opção de realimentação com mais de um padrão de resposta foi desenvolvida permitindo que o docente faça o cadastro de uma ou várias respostas padrão. Isso permite que sejam evitadas injustiças na correção das questões, possibilitando, ainda, uma melhor avaliação dos critérios de julgamento realizados ao longo do processo, podendo, até mesmo ser utilizada para comparar as respostas fornecidas entre os alunos para a detecção de plágio.

4. Experimentos realizados

Para efetuar os experimentos, foram utilizadas duas bases de dados com as seguintes características:

- Primeira amostra com 3 questões dissertativas de avaliação realizada por 10 estudantes, totalizando, 30 respostas para análise. Prova aplicada na disciplina de sistemas e aplicações multimídia do curso de graduação em engenharia de teleinformática.
- Segunda amostra com 10 questões dissertativas de avaliação realizada por 13 estudantes, totalizando 130 respostas para análise. Prova aplicada na disciplina de desenvolvimento de aplicações para a web do curso de graduação em engenharia de teleinformática.

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 no trabalho proposto. Para comparar as respostas, o docente da disciplina forneceu uma resposta padrão para cada questão.

Em particular, pretende-se testar um conjunto de questões previamente corrigidas e comparar os resultados obtidos pela ferramenta desenvolvida com a nota atribuída a cada texto analisado.

Tratando-se de uma primeira abordagem ao problema, o estudo será realizado em dois níveis distintos: avaliação da similaridade somente com algoritmos de busca e comparação e avaliação da similaridade com algoritmos com técnicas de pré-processamento de texto.

O problema que se coloca no primeiro caso é determinar quais algoritmos devem ser utilizados para a correção de questões dissertativas. No segundo caso, determinar quais técnicas de pré-processamento são capazes de melhorar os resultados dos índices de similaridade, além de verificar a melhor ordem de execução das mesmas.

Um fator importante que distingue os dois casos e que poderá ter influência decisiva nos resultados é o tamanho dos textos e a sua influência nos resultados.

4.1. Resultados de similaridade com algoritmos de busca e comparação textual

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. Como o algoritmo Levenshtein calcula a distância de cada palavra, os seus índices acabaram ficando, na maioria dos casos utilizados neste trabalho, com índices superiores aos demais. Por ter apresentando melhores índices de similaridade, o algoritmo Levenshtein foi adotado em todos os demais experimentos.

Tabela 1. Similaridade utilizando algoritmos de busca e comparação textual

Questão 01	Boyer-Moore	Rabin-Karp	Levenshtein	Brute Force	Knuth-Morris-Pratt
	Similaridade	Similaridade	Similaridade	Similaridade	Similaridade
Aluno 01	32,58	32,58	36,93	32,58	32,58
Aluno 02	43,82	43,82	55,89	43,82	43,82
Aluno 03	44,94	44,94	57,81	44,94	44,94
Aluno 04	44,94	44,94	56,76	44,94	44,94
Aluno 05	41,57	41,57	53,47	41,57	41,57
Aluno 06	46,06	46,06	55,74	46,06	46,06
Aluno 07	44,94	44,94	57,68	44,94	44,94
Aluno 08	50,56	50,56	53,42	50,56	50,56
Aluno 09	31,46	31,46	31,16	31,46	31,46
Aluno 10	42,69	42,69	46,89	42,69	42,69

4.2. Resultados de similaridade com técnicas de pré-processamento textual encontradas na literatura

A Tabela 2 demonstra os resultados dos experimentos com o uso de algoritmos de busca e comparação em conjunto com técnicas de pré-processamento de texto. Conforme pode ser visto, cada técnica de pré-processamento de texto resultou em diferentes índices de similaridade. Tendo como base a resposta do Aluno 07, destacado em negrito, em todos os casos o uso das técnicas de pré-processamento de texto obtiveram melhores índices de similaridade.

Tabela 2. Similaridade com algoritmo proposto e técnicas de pré-processamento

Questão 02	Levenshtein	Levenshtein + Rem. acentuação	Levenshtein + Rem. caracteres inválidos	Levenshtein + Rem. Palavras repetidas	Levenshtein + Rem. stopw ords	Levenshtein + Stemmer Orengo	Levenshtein + Aplicar uppercase
Aluno 01	50,69	51,07	50,88	52,34	47,64	50,60	51,34
Aluno 02	42,50	42,81	42,38	44,05	41,12	43,90	42,91
Aluno 03	56,50	56,76	56,86	56,86	55,46	56,88	56,84
Aluno 04	50,04	50,61	50,70	52,67	50,96	50,93	50,51
Aluno 05	50,88	51,52	50,88	49,67	48,85	50,76	51,34
Aluno 06	39,16	39,91	39,14	41,84	42,13	39,54	39,48
Aluno 07	64,14	64,80	64,29	64,05	63,29	64,74	64,54
Aluno 08	42,13	42,34	42,12	42,85	40,97	40,34	42,66
Aluno 09	33,95	34,39	34,47	40,05	35,52	33,97	34,39
Aluno 10	61,92	62,47	62,17	63,64	62,71	61,99	62,62

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 possíveis¹, sendo apresentados na Tabela 3 os resultados obtidos.

Podemos evidenciar em relação ao resultado das permutações, que:

- a técnica de *stemming* (SOR), quando aplicada no início do pré-processamento, não alcançou os melhores índices de similaridade;
- a técnica de aplicação de *uppercase* (AUC) não influenciou nos resultados, independentemente da ordem que ocorreu nas permutações. Isso se deve devido à técnica SOR aplicar *lowercase* durante as suas transformações;
- os melhores índices de similaridade ocorreram quando a técnica de remoção de caracteres inválidos (RCI) foi aplicada antes da técnica SOR; e,
- a técnica substituição de caracteres acentuados (RAP) deve ser aplicada após a SOR, uma vez que o *stemming* possui regras que utilizam a acentuação das palavras.

Tabela 3. Resultado dos testes com permutação das técnicas de pré-processamento

Permutações	Questão 2 Aluno 07	Permutações	Questão 2 Aluno 07
AUC+RAP+RCI+SOR	66,48	RCI+AUC+SOR+RAP	65,63
AUC+RAP+SOR+RCI	65,63	RCI+RAP+AUC+SOR	66,72
AUC+RCI+RAP+SOR	66,72	RCI+RAP+SOR	66,72
AUC+RCI+SOR+RAP	66,48	RCI+RAP+SOR+AUC	66,72
AUC+SOR+RAP+RCI	65,71	RCI+SOR+AUC+RAP	65,71
AUC+SOR+RCI+RAP	65,71	RCI+SOR+RAP	65,71
RAP+AUC+RCI+SOR	66,72	RCI+SOR+RAP+AUC	65,71
RAP+AUC+SOR+RCI	65,63	SOR+AUC+RAP+RCI	65,71
RAP+RCI+AUC+SOR	66,72	SOR+AUC+RCI+RAP	65,71
RAP+RCI+SOR	66,72	SOR+RAP+AUC+RCI	65,71
RAP+RCI+SOR+AUC	66,72	SOR+RAP+RCI	65,71
RAP+SOR+AUC+RCI	65,63	SOR+RAP+RCI+AUC	65,71
RAP+SOR+RCI	65,63	SOR+RCI+AUC+RAP	65,71
RAP+SOR+RCI+AUC	65,63	SOR+RCI+RAP	65,71
RCI+AUC+RAP+SOR	66,72	SOR+RCI+RAP+AUC	65,71

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 $[(4!) + (3!) = 30]$.

4.3. Resultados de similaridade com técnicas de pré-processamento textual propostas

É apresentado, a seguir, o funcionamento passo-a-passo das técnicas de pré-processamento de texto, com o objetivo de melhorar o entendimento sobre as mesmas e comprovar as modificações realizadas. As técnicas ainda não utilizadas, e propostas neste trabalho, também são apresentadas nesta série de experimentos.

Para os experimentos, foram utilizadas as seguintes frases como padrão de resposta:

- Padrão de resposta: “A mensagem PATH é orientada do emissor para o receptor, construindo o caminho através do qual o fluxo deve passar, armazenando em cada nó o endereço do nó anterior.”
- Resposta do aluno: “A mensagem path é orientada do emissor para o receptor e sua função é ir guardando os endereços IP de cada nó.”

Somente com o algoritmo Levenshtein a similaridade entre as frases, na sua escrita original, ficou em 68,98%. A Tabela 4 exhibe o resultado de cada transformação e os seus respectivos índices de similaridade. Com base no resultado das transformações, pode-se inferir que:

- utilizando a técnica de trocar palavras semelhantes (TPS), houve aumento no índice de similaridade ao substituir “guardando” por “armazenado” e “caminho” por “fluxo”. De acordo com os experimentos realizados, esta técnica deve ser a primeira a ser aplicada nas permutações, pois a mesma não deve sofrer as modificações das demais técnicas;
- a técnica de remover *stopwords* (RSW), nesse experimento, foi a que mais melhorou os índices de similaridade, aumentando-o em 4,55%, comprovando a sua utilidade para esse tipo de aplicação;
- a técnica de remover palavras repetidas (RPR), além de melhorar os índices de similaridade, também auxiliou em diminuir o tempo de execução em milissegundos do algoritmo. Isso se deve ao fato de a quantidade de palavras nas duas frases terem diminuído, reduzindo o número de comparações.

Tabela 4. Passo-a-passo da execução das técnicas de pré-processamento textual

Permutações	Padrão de resposta	Resposta do aluno	Similaridade	Tempo de execução (ms)
TPS	A mensagem PATH é orientada do emissor para o receptor, construindo o caminho através do qual o fluxo deve passar, armazenando em cada nó o endereço do nó anterior.	A mensagem path é orientada do emissor para o receptor e sua função é ir armazenando os endereços IP de cada nó.	71,27	1,281
TPS+RCI	A mensagem PATH é orientada do emissor para o receptor construindo o fluxo através do qual o fluxo deve passar armazenando em cada nó o endereço do nó anterior	A mensagem path é orientada do emissor para o receptor e sua função é ir armazenando os endereços IP de cada nó	72,22	1,251
TPS+RCI+RSW	mensagem PATH é orientada emissor receptor construindo fluxo fluxo deve passar armazenando no endereço no anterior	mensagem path é orientada emissor receptor função é ir armazenando endereços IP no	76,77	1,144
TPS+RCI+RSW+SOR	mens path é orient emis recep constru flux flux dev pass armazen no endereç no anterior	mens path é orient emis recep func é ir armazen endereç ip no	77,33	1,407
TPS+RCI+RSW+SOR+RAP	mens path e orient emis recep constru flux flux dev pass armazen no enderec no anterior	mens path e orient emis recep func e ir armazen enderec ip no	78,67	1,385
TPS+RCI+RSW+SOR+RAP+RPR	mens path e orient emis recep constru flux dev pass armazen no enderec anterior	mens path e orient emis recep func ir armazen enderec ip no	80,00	0,993

Outro ponto positivo em relação à técnica TPS desenvolvida, é que a mesma possui um custo computacional de execução baixo em relação a outras técnicas de pré-

processamento de texto que executam a mesma tarefa, como, por exemplo, o uso de *Thesaurus*, principalmente por ser executado localmente, utilizando um arquivo XML customizável. Além disso, o acesso a um *Thesaurus* teria que ser feito com o uso de Internet, consumindo um serviço disponibilizado na Web, aumentando o tempo de espera e, provavelmente, o de execução.

A proposta apresentada neste trabalho tem o mesmo potencial de padronização de textos que façam o uso de um *Thesaurus*, guardadas as devidas proporções. Quanto mais palavras forem cadastradas no arquivo de configuração XML, melhores serão as chances de melhorar os índices de similaridade, uma vez que ocorrerá uma padronização do vocabulário dentro das frases comparadas.

4.4. Considerações sobre realimentação de respostas consideradas certas

A ferramenta desenvolvida permite que seja inserido mais de um padrão de resposta, através da leitura de arquivos de textos, permitindo, dessa forma, verificar o índice de similaridade, até mesmo, entre as respostas fornecidas pelos alunos. Esta funcionalidade não é considerada uma técnica de pré-processamento de texto, uma vez que não altera o conteúdo das frases fornecidas.

Os resultados realizados com esta funcionalidade, proposta neste trabalho, mostraram o potencial da mesma para a comparação de várias respostas ao mesmo tempo, sendo indicada a sua adoção para a avaliação dos índices de similaridade de questões dissertativas.

Essa funcionalidade serve para avaliar, ainda, outras características:

- Alimentando com dois ou mais padrões de resposta – Sendo inserido mais de um padrão de resposta pelo docente, certamente os índices de similaridade poderão ser melhorados, desde que as respostas sejam elaboradas para cobrir novos termos e palavras mais próximos do vocabulário dos alunos.
- Realimentando com uma resposta considerada correta fornecida por um aluno – Comparando a resposta de um aluno com as dos demais estudantes de uma turma, pode-se verificar a proximidade das palavras e termos utilizados por eles.
- Detecção de plágio – Efetuando a comparação das respostas de todos os alunos entre eles, caso os índices de similaridades sejam muito altos, pode ser indicativo de algum tipo de plágio (ou, como se diz no jargão popular, “cola” ou “pesca”). Pode-se, ainda, nessa mesma perspectiva, usar a técnica para procurar a similaridade entre respostas incorretas.

4.5. Considerações sobre a sequência de palavras definidas (SSD)

A proposta dessa técnica de pré-processamento de texto foi de criar um mecanismo de ajuste que permita ao professor inserir termos ou sequências de palavras que possuam significado coerente quando escritas em conjunto (ex.: “o IP não é válido” é oposto a dizer “o IP é válido”, embora o conjunto de palavras da segunda expressão seja completamente encontrado na primeira). Dessa forma, ao inserir um termo, o docente indicará se o mesmo irá atribuir um match positivo ou negativo a questão.

Para facilitar o entendimento da aplicação desta técnica, é apresentado abaixo o enunciado de uma questão, seu padrão de resposta e respostas fornecidas pelos alunos.

Enunciado da questão: Uma aplicação de rede precisa utilizar os serviços de uma camada específica da pilha de protocolos, ainda que essa comunicação seja intermediada por outra aplicação. No caso do TCP/IP, que camada é essa?

Resposta padrão: A camada de transporte.

Resposta do Aluno 01: A camada de aplicação vai utilizar os serviços da camada de transporte para fazer a comunicação.

Resposta do Aluno 02: Transporte

Resposta do Aluno 03: Camada de transporte.

Resposta do Aluno 04: Camada de transporte.

Resposta do Aluno 05: Camada de rede (camada 3).

Resposta do Aluno 06: Camada de transporte.

Resposta do Aluno 07: Camada de transporte.

Resposta do Aluno 08: Transporte.

Resposta do Aluno 09: Transporte.

Resposta do Aluno 10: Camada de Transporte.

O resultado do experimento é apresentado na Tabela 5. Com o uso da técnica de sequência de palavras definidas (SSD) foram feitos ajustes que permitiram que as respostas dos Alunos 01, 02, 08 e 09 melhorassem o índice de similaridade, nesse caso, por serem respostas corretas. Já a resposta do Aluno 05, mesmo errada, ficou com um índice de 63,64%. O SSD ajustou sua similaridade para 0,00%.

Tabela 5. Resultados com uso da técnica de sequência de palavras definidas

Respostas	Permutação	
	Sem SSD	Com SSD
Aluno 01	41,18	100,00
Aluno 02	83,33	100,00
Aluno 03	100,00	100,00
Aluno 04	100,00	100,00
Aluno 05	63,64	0,00
Aluno 06	100,00	100,00
Aluno 07	100,00	100,00
Aluno 08	83,33	100,00
Aluno 09	83,33	100,00
Aluno 10	100,00	100,00

5. Conclusões e Trabalhos Futuros

A principal contribuição deste trabalho foi validar e 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 uma parte do 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 índices de similaridade alcançados indicam que a solução proposta pode ser utilizada como ferramenta de apoio na correção de questões dissertativas. O desenvolvimento da ferramenta utilizada nos testes representa uma contribuição do presente trabalho. A análise da massa de dados utilizada neste trabalho só foi possível porque a ferramenta foi desenvolvida seguindo os objetivos especificados na metodologia de testes proposta.

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. Deve-se, ainda, ser levado em consideração que a avaliação de classificadores de textos é realizada experimentalmente e não analiticamente, principalmente devido à subjetividade e forma não estruturada dos dados [Sebastiani, 2002].

Como continuidade deste trabalho, alguns pontos pendentes ainda deverão vir a ser considerados para a melhoria da pesquisa realizada e da ferramenta desenvolvida, como os que se seguem:

- executar testes de exemplos mais complexos e testes exaustivos de todas as estratégias de análise dos dados, visando a corrigir eventuais falhas e tornar a ferramenta mais robusta;
- inserir novos algoritmos de comparação, podendo, inclusive, ser utilizadas outras tecnologias como Avaliação Assistida pelo Computador (CAI) desenvolvidas até o momento, como a Análise da Semântica Latente (ASL), Processamento de Linguagem Natural (PLN) e Inteligência Artificial (IA), entre outros;
- utilizar um *Thesaurus* em português do Brasil, quando disponível, para verificar se é possível melhorar os índices de similaridade;
- 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 AVEA, como o Moodle, por exemplo;
- realizar outros estudos experimentais, com amostras maiores e em outros contexto (por exemplo, em disciplinas de cursos de graduação em ciência da computação e/ou administração) para verificar a aplicabilidade e acurácia da ferramenta proposta com maior significância estatística.

Referências Bibliográficas

- Azevedo, B. F. T.; Behar, P. A. e Reategui, E. B. (2011) “Análise das mensagens de fóruns de discussão através de um software para mineração de textos.” In: Anais do XXII Simpósio Brasileiro de Informática na Educação, p. 20-29.
- 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.
- Cormen, T., Leiserson, C. e Rivest, R. (2009) “Introduction to algorithms”. 3rd Edition, The MIT Press, Pages 985-1002.
- Hearst, M. A. (1992). “Automatic Acquisition of Hyponyms from Large Text Corpora”. In: Proceedings of the Fourteenth International Conference on Computational Linguistics. Nantes, France.
- 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.
- Luhn, H. P. (1966). “Keyword-in-context index for technical literature.” *American Documentation*, 11(4):288–295.
- Moens, Marie-Francine. (2000) “Automatic indexing and abstracting of document texts”. *Artificial Intelligence and Law*, The Kluwer International Series on Information Retrieval Vol. 6, Pages 343-347.
- Page, E. B. (1967). “Grading essays by computer: Progress report.” *Proceedings of the 1966 Invitational Conference on Testing*. Princeton, pp. 87-100.
- Rodrigues, F., Araújo, L. (2012) “Automatic Assessment of Short Free Text Answers.” In *Proceedings of the 4th International Conference on Computer Supported Education – CSEDU 2012 – Porto, Portugal*, 16-18 April, pp. 50-57.
- Sebastiani, F. (2002) “Machine Learning in Automated Text Categorization.” *ACM Computing Surveys* 34 (1), p. 1–47.
- Sphair, M. J. K. (2006) “Um estudo sobre o Eureka e seu impacto no processo de ensino/aprendizagem”. *Dissertação (Mestrado em Educação)*, Pontifícia Universidade Católica do Paraná, Curitiba: PUC-PR.
- Tavares, D. A. B. *et. al.* (2010) “Integração do ambiente WIMS ao Moodle usando Arquitetura Orientada a Serviços e Compilação Automática de Médiatextos.” *RENOTE - Revista Novas Tecnologias na Educação*, v. 8, n. 3, Porto Alegre: UFRGS.
- Ziviani, N. (2010) “Projeto de Algoritmos com Implementações em Pascal e C”. São Paulo: Pioneira Thomson Learning, 3ª Edição.