Utilização do Caminhamento Aleatório na Identificação de Características de Documentos na Língua Portuguesa

Vagner Francisco Le Roy Júnior[1]

Ana Paula Ladeira[2]

Resumo: Devido ao grande volume de textos armazenados, a área de mineração de textos vem sendo foco de inúmeras pesquisas que visam a classificação automática de documentos. O presente trabalho tem como objetivo avaliar o método do caminhamento aleatório na definição dos pesos dos termos de textos da língua portuguesa. Esta técnica utiliza a co-ocorrência dos termos como medida de dependência entre as características das palavras. Um grafo não direcionado é utilizado, sendo que a pontuação de cada vértice é calculada em função da probabilidade de ser encontrado. Os resultados obtidos com o caminhamento aleatório foram comparados com os apresentados por técnicas tradicionais, e demonstraram que o método de caminhamento aleatório se mostrou bastante eficaz no processo de classificação de documentos.

Abstract: *Due to a great amount of web-stored texts, the text mining area has been coming through a series of studies in order to optimize the automatic classification of texts. In this context, this study is aimed at testing a technique to assess the weight of terms named random-walk and applying it to Portuguese language texts. This technique uses the co-occurrence of the terms as a measure between the words' features. Considering this purpose, it uses an undirected graph and each vertex score is calculated according to the probability of being found by a hiker. To provide the results, the CETENFolha dataset was utilized as well as the Weka tool, used in the classification of texts. The results obtained were compared to the traditional weight of terms assessing techniques and showed that the random-walk method is quite effective to this purpose.*

1 Departamento de Ciências Exatas, UNI-BH, Caixa Postal 9999 {vagnerleroy@gmail.com}

2 Departamento de Ciências Exatas, UNI-BH, Caixa Postal 9999 {aladeira.unibh@gmail.com}

1 Introdução

O grande volume de dados gerados na WEB e principalmente, a forma como os documentos são armazenados nas empresas, normalmente de forma não estruturada, impulsionou estudos focados em pesquisar, organizar e categorizar esses documentos [1]. Pesquisas mostram que 80% dos dados de uma organização e da web estão armazenados em formato textual [3;13].

Neste contexto, a mineração de textos (*Text Mining*) surgiu com o objetivo de obter informações relevantes e descobrir conhecimento significativo em documentos textuais [9]. Em outras palavras, pode ser considerada uma extensão da área de mineração de dados (*Data Mining*) com o enfoque na análise de textos [1].

Geralmente, o processo de mineração de textos é composto por três grandes etapas: pré-processamento dos dados, análise e extração do conhecimento e pós-processamento [9].

A etapa de pré-processamento efetua a limpeza dos dados para as etapas seguintes e é dividida em três fases: correção ortográfica, remoção de *stopwords* (eliminação de artigos, por exemplo) e *stemming* (substituição da palavra por seu radical, ou *stem*). Na análise e extração do conhecimento, os algoritmos de mineração de textos são aplicados para a obtenção dos resultados. No pós-processamento, os resultados são avaliados, ou seja, são interpretados e validados [9].

O presente trabalho se concentra na etapa de análise e extração do conhecimento com enfoque na avaliação dos pesos dos termos, isto é, decidir se um termo é ou não relevante. As técnicas tradicionais de avaliação dos termos trabalham com dois grupos: a frequência do termo no documento (*tf*), que conta quantas vezes o termo aparece e nos dá uma medida de quão bem o termo descreve o conteúdo do documento, e a frequência inversa do documento (*idf*), que quantifica um termo como fator discriminatório para todos os documentos da coleção [10].

Entretanto, essas técnicas tendem a falhar em não avaliar a dependência que as palavras do texto exercem uma nas outras, ou seja, descartam o efeito global que um termo exerce no documento inteiro [5]. Os algoritmos de caminhamento aleatório desenvolvidos por Hassan, Mihalcea e Banea [5] propõem uma maneira de determinar os pesos dos termos baseado no caminhamento aleatório, ao invés da escolha aleatória. Para isso, eles imaginam um leitor percorrendo o texto de forma aleatória, onde a importância do termo no documento é determinada pela probabilidade do leitor o encontrar.

Portanto, a técnica de Caminhamento Aleatório determina que tanto a localização do termo quanto o seu contexto seja levado em consideração [5]. Ela é aplicada em um grafo onde cada termo do texto corresponde a um vértice e a importância do termo é decidida usando a informação global calculada recursivamente a partir do grafo inteiro.

Como as variações linguísticas causam um efeito drástico na implementação de algoritmos de processamento de textos, não se pode afirmar que os resultados obtidos por uma implementação que utiliza documentos em uma língua se repetirão quando muda-se o idioma dos documentos. Isto obriga a estabelecer novos parâmetros específicos para o idioma e consequentemente, testar e comparar os resultados [1].

Sendo assim, o objetivo geral deste artigo é aplicar o algoritmo de caminhamento aleatório para calcular os pesos dos termos em documentos da língua portuguesa, comparando os resultados com os obtidos utilizando as técnicas usuais, tais como *tf* e *idf*.

Na Seção 2 é apresentada toda a fundamentação teórica utilizada, desde as técnicas tradicionais de avaliação de pesos dos termos, passando pela explicação do método de caminhamento aleatório, e terminando nos algoritmos de classificação. A Seção 3 aborda toda a metodologia e os passos para a obtenção dos resultados explicitados na Seção 4. Por último, a Seção 5 contém a conclusão e trabalhos futuros.

2 Fundamentação Teórica

O Modelo de Espaço Vetorial proposto por Salton e Buckley [11] é a forma mais comum de representar documentos em mineração de textos [4]. Neste modelo, calcula-se os pesos através de dois conjuntos: a importância do termo no documento e a importância global do termo.

Nas seções a seguir, serão apresentadas tanto algumas métricas conhecidas de importância de pesos, como os algoritmos de classificação, que foram usados nos experimentos do presente trabalho.

2.1 Frequência do Termo (*TF*) e Frequência Inversa do Documento (*IDF*)

A importância do termo no documento pode ser determinada pela frequência do termo (*tf*), contabilizando quantas vezes o mesmo apareceu no texto. Já a importância global do termo, levando-se em consideração toda coleção de documentos, é determinada pela frequência inversa do termo, comumente referenciada como frequência inversa do documento (*idf*) [6]. O peso de cada termo é determinado pela Equação (1).

$$tf_{i,j} \times idf_i = tf_{i,j} \times \log(D / df_i) \quad \textbf{(1)}$$

Onde df_i, é o número de documentos contendo o termo i; D é o número de documentos da coleção e tf_i é a frequência do termo. tf_i pode ser calculado a partir da Equação (2).

$$tf_{i,j} = \frac{freq_{i,j}}{\max_l \cdot feq_{l,q}} \quad (2)$$

Onde $tf_{i,j}$ é a frequência do termo k_i no documento d_j ; $freq_{i,j}$ é a frequência em que o termo k_i aparece no documento d_j ; max_l . $feq_{l,q}$ é a frequência máxima de um termo qualquer no texto da consulta. Para calcular o peso dos termos na consulta, Salton e Buckley[11] sugerem uma modificação na Equação (2):

$$tf_{i,j} \times idf_j = \left(0{,}5 + \frac{0{,}5\, freq_{i,q}}{\max l \cdot freq_{1,q}} \right) \times \log(D / df_i) \quad (3)$$

2.2 Algoritmos de Caminhamento Aleatório

O algoritmo de caminhamento aleatório basei-se na noção de voto ou recomendação. Quando um vértice é ligado a outro por meio de uma aresta, um voto é computado a ambos. Quanto mais votos o vértice tiver, mais importante ele é para o documento [6].

Entre vários algoritmos de caminhamento, o PageRank é utilizado com sucesso em diversas aplicações, incluindo redes sociais, análise de links da web, análise de citações, e por último, processamento de texto [5].

Dado um Grafo $G = (V,E)$, seja $In(V_a)$ o conjunto de vértices que apontam para V_a (antecessores), e $Out(V_a)$ o conjunto de vértices para que V_a aponta (sucessores). A pontuação utilizando PageRank, associada ao vértice V_a, é calculada a partir de uma função recursiva que integra o resultado dos antecessores (Equação (4)).

$$S(V_a) = (1-d) + d \times \sum_{V_b \in In(V_a)} \frac{S(V_b)}{|Out(V_b)|} \quad (4)$$

Onde d é um parâmetro que varia entre 0 e 1. A pontuação de cada vértice é calculada a cada iteração, levando-se em conta o novo peso do vizinho. O algoritmo converge quando a taxa de erro de todos os vértices fica abaixo de um limite pré-estabelecido.

Assim, a pontuação do vértice é obtida em função da probabilidade de ser encontrado durante a execução do algoritmo, determinando, assim, sua importância no grafo. Esta é a ideia central do caminhamento aleatório.

Apesar do algoritmo de caminhamento aleatório ser utilizado para diversos fins, no presente trabalho o mesmo foi aplicado no processamento de textos, como feito no método de TextRank [5]. O algoritmo de TextRank tem sido aplicado a três tarefas de processamento de linguagem natural, sendo elas: sumarização de documentos, resolução da ambiguidade de

palavras e extração de palavras chaves em textos. A vantagem deste modelo está na representação global do contexto e como a co-ocorrência de recursos pode propagar por todo contexto e afetar a outras características distantes [5].

Hassan, Mihalcea e Banea [5] propõem uma abordagem similar à usada na aplicação do TextRank na extração de palavras-chave, que deriva os pesos dos termos utilizando uma representação gráfica que mostra a co-ocorrência de dependências entre palavras no texto.

2.3 Modelos de Caminhamento Aleatório

Hassan, Mihalcea e Banea [5] apresentam métodos de caminhamento aleatório para atribuição de pesos aos termos de um documento baseados em variações do PageRank com adição de informação e variáveis na Equação (4), dentre eles:

Rw_o: método original representado na Equação (4) em que usa-se um grafo não orientado com uma constante de fator de amortecimento (*dumping factor*), seguindo rigorosamente a fórmula tradicional do PageRank.

$Rw_{e.idf}$: este modelo representa uma abordagem do grafo não orientado que utiliza a versão dos pesos das arestas do PageRank com uma variável do fator de amortecimento. O peso de cada aresta é calculado a partir da seguinte equação:

$$E_{v1,v2} = tf \cdot idf_{v_1} \times tf \cdot idf_{v_2} \quad (5)$$

Sendo que $E_{vi,v2}$ é o peso da aresta que conecta o vértice 1 ao vértice 2 e $tf \cdot idf_{v_1} \times tf \cdot idf_{v_2}$ é a frequência do termo multiplicada pela frequência inversa do documento representados nos vértices 1 e 2, respectivamente. Eles sugerem o cálculo do fator de amortecimento em função dos pesos das arestas de entrada, como segue:

$$d_{E_{v1,v2}} = E_{v1,v2} / E_{\max} \quad (6)$$

Onde $d_{Ev1,v2}$ é o fator de amortecimento e E_{max} é o maior peso de uma aresta no grafo. Quanto maior este valor, mais importante é o termo para o documento. Assim, a pontuação de cada vértice é determinada pela equação a seguir:

$$s'(v_1) = \frac{(1-d)}{|N|} + \sum_{V2 \in iN(V1)} C \times \frac{d_{E_{v1,v2}} \times s(v_2)}{|Out(v_2)|} \quad (7)$$

Onde *s´(v1)* é a pontuação do vértice, *N* é o número total de nós e *C* é uma constante escalar. Este modelo integra tanto a localidade do termo quanto sua relação com o contexto do documento.

Com o objetivo de dar um maior poder discriminatório a um termo que pode identificar o documento, utilizou-se, para o cálculo de *tf-idf*, uma variação da Equação (1), proposta por Sun, He e Chen [12] e utilizada por Islam e Islam [6], que acrescenta o modelo de ganho de informação:

$$tf_{i,j} \times idf_i = \frac{tf_{i,j} \times \log\left(\frac{D}{df_i} + 0.01\right) \times IG_i}{\sqrt{\sum_{t \in d}\left[tf_{i,j} \times \log\left(\frac{D}{df_i} + 0.01\right) \times IG_i\right]^2}} \quad (8)$$

Onde $tf_{i,j}$ é a frequência do termo i no documento j, idf_i é a frequência inversa do documento para o termo i, df_i é o número de documentos que contém o termo i, e D é o número de documentos da coleção. O resultado de $tf_{i,j} \times idf_i$ reflete a importância do termo i em toda a coleção de documentos. IG_i é o ganho de informação referente a um termo i, e é calculado a partir da seguinte equação:

$$IG_{termo(i)} = H(D) - H(D \mid term(i)) \quad (9)$$

$$H(D) = -\sum_{t \in D} P(t) \times \log_2 P(t) \quad (10)$$

$$P(d) = \frac{N_t}{\sum_{t \in D} N_t} \quad (11)$$

Onde *term(i)* é um termo ti em um documento, N_t é o número de termos t em um documento D, $P(t)$ é a probabilidade de um termo t aparecer em um documento, e $H(D)$ é a entropia do documento D. O termo $H(D|term(i))$ pode ser calculado como segue:

$$H(D \mid term(i)) = H(D \mid t_i) \quad (12)$$

$$H(D \mid t_i) = -\sum_{t \in D} P(t, t_i) \times \log_2 P(t \mid t_i) \quad (13)$$

$$P(t, t_i) = \frac{n_1}{N} \quad (14)$$

$$P(t \mid t_i) = \frac{n_1}{n_2} \quad (15)$$

Onde n_1 é o número de documentos que contêm ambos t_i e t, n_2 é o número de documentos que contêm t_i, e N é o número de documentos.

2.4 Algoritmos de Classificação

Para determinar o percentual de instâncias classificadas corretamente e obter as medidas de precisão (*precision*) e revocação (*recall*), foi utilizado o algoritmo de classificação de texto Naive Bayes. Precisão mede a fração de documentos recuperados que são relevantes e revocação mede a fração de documentos relevantes que foram recuperados [2].

A ideia central de um classificador de textos Naive Bayes é estimar a probabilidade de uma determinada categoria de um documento, usando probabilidades conjuntas de palavras e documentos [5]. As probabilidades são calculadas pela contagem da frequência de cada valor de característica para as instâncias dos dados contidos no treino [8].

Vale destacar que, para avaliar a eficácia dos métodos de atribuição dos pesos dos termos, é necessário que os documentos utilizados nos experimentos estejam classificados, ou seja, é necessário conhecer a categoria na qual eles se encaixam.

3 Metodologia

Nesta seção é discutida a metodologia aplicada no presente trabalho para a obtenção dos resultados. A implementação foi desenvolvida utilizando a plataforma Java[3]. Nas próximas seções são apresentados o processo de preparação dos dados para a etapa de análise e extração do conhecimento, incluindo o pós-processamento feito na classificação de textos para a obtenção dos resultados. O corpus de documentos utilizado para os experimentos é apresentado na Seção 3.4.

3.1 Etapa de Pré-Processamento

Nesta etapa é feita a limpeza dos dados, ou seja, a aplicação de algoritmos de remoção de *stopwords*, *stemming* e correção ortográfica. Posteriormente, as palavras são "tokenizadas" e só são adicionados ao grafo os substantivos e os adjetivos. Para evitar o crescimento excessivo do grafo em decorrência de todas as possíveis combinações de sequência consistindo de mais de uma unidade léxica, só são adicionadas palavras únicas.

[3] Disponível em: www.oracle.com/technetwork/java/index.html

Para os testes, foram criados dois corpus de documentos com base no corpus da CETENfolha que será apresentado posteriormente. O corpus C1 contém 900 textos divididos em seis categorias, sendo elas: desportos, sociedade, veículos, informativa, economia e agricultura. O corpus C2 contém 900 textos divididos em quatro categorias: agricultura, desportos, informática e veículos.

A partir do corpus estabelecido, todas as palavras foram removidas, exceto os substantivos e os adjetivos. Em seguida, os textos foram submetidos à ferramenta Pretext[4] para remoção de *stopwords* e *stemming*. Após, as palavras foram adicionadas ao grafo.

3.2 Etapa de Análise e Extração do Conhecimento

Os termos são comumente modelados como sendo nós. Os termos que possuem co-ocorrência dentro de uma janela de tamanho N são, então, ligados por arestas. Para um determinado termo, as palavras que aparecem nas suas proximidades são consideradas termos dependentes. Isto é modelado ligando o termo aos seus dependentes através de um conjunto de arestas. A Figura 2 mostra um exemplo de grafo modelado para o texto da Figura 1, assumindo uma janela de tamanho 2 como a utilizada neste trabalho.

pt govern brasil pesquis datafolh supreendent postur radical esmag maior eleit pt govern fern henriqu cardos temp ditadur solidez pt lul part discurs oposi prioridad nov govern prioridad PT

Figura 1. Texto de Exemplo

[4] Disponível em: http://sites.labic.icmc.usp.br/pretext2/#1

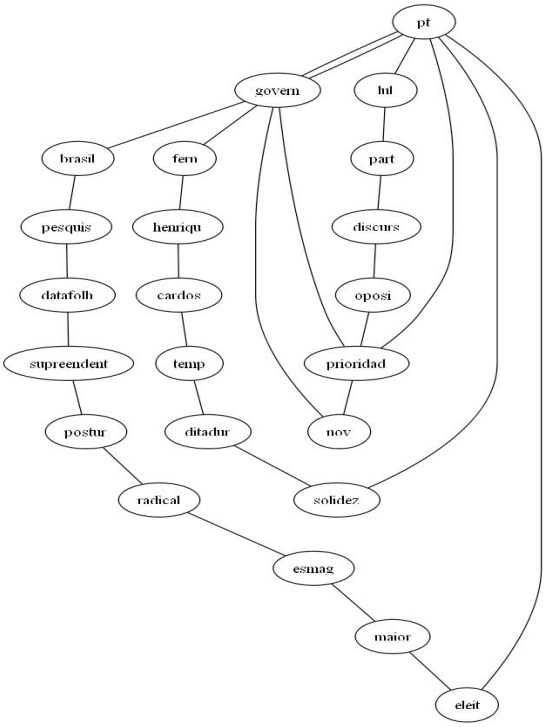

Figura 2. Grafo de Exemplo

Após todas as palavras terem sido adicionadas ao grafo, aplica-se o algoritmo de caminhamento aleatório (Equação (4)) até o valor limite de convergência atingir 0.0001 para todos os vértices. A constante *d* foi estipulada em 0.85, como a utilizada por Hassan, Mihalcea e Banea [5]. Deste modo, obtêm-se Rw_o.

Em seguida, calcula-se $Rw_{e.idf}$ através da Equação (7). A constante *C* é comumente configurada em 0.95, o peso de cada aresta é calculado através da Equação (5), e o valor do fator de amortecimento $d_{Ev1,v2}$ é calculado através da Equação (6).

Para efeitos comparativos, algoritmos de *tf*, *tf-idf* e algumas de suas variações também foram implementados: *tf* representa a frequência do termo dentro do documento, ou seja, quantas vezes a palavra ocorreu dentro do texto; *tf-idf* representa o método da Equação (1); $tf\text{-}idf_{norm}$ representa a Equação (9), porém, sem a adição do ganho de informação(*IG*), e $tf\text{-}idf_{norm.ig}$ representa a Equação (9) por completa.

3.3 Etapa de Pós-Processamento

Por fim, para obter os resultados, os arquivos contendo os resultados de cada método de avaliação foram submetidos ao algoritmo de classificação Naive Bayes, com validação cruzada (cross-validation 10 folds), através da ferramenta open-source de data mining Weka[5] - um software livre, desenvolvido em Java, e que contêm uma série de algoritmos de classificação já implementados. A Fig. 3 demonstra um resumo de todo o processo.

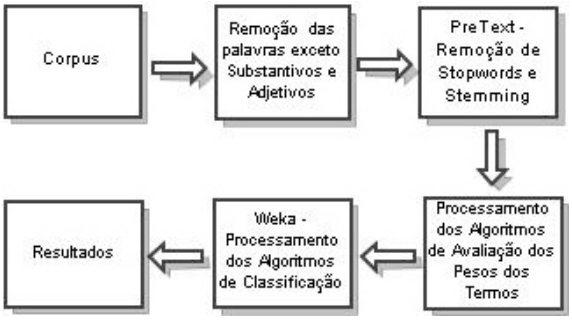

Figura 3. Fluxograma do processo

3.4 Corpus de Documentos

O CETENFolha[6] (Corpus de Extratos de Textos Eletrônicos NILC/Folha de São Paulo), é um corpus de cerca de 24 milhões de palavras em português brasileiro, criado pelo projeto Processamento Computacional do Português com base nos textos do jornal Folha de São Paulo, que fazem parte do corpus NILC/São Carlos, compilado pelo Núcleo Interinstitucional de Linguística Computacional (NILC).

Este corpus contém cerca de 34.000 textos divididos em onze categorias, sendo elas: política brasileira e internacional, desportos, sociedade, economia, cultura, opinião, agricultura, veículos, informática e não determinada.

[5] Disponível em: www.cs.waikato.ac.nz/ml/weka/
[6] Disponível em: www.linguateca.pt/cetenfolha

4 Resultados Obtidos

Esta seção contém os resultados obtidos com os experimentos. Na Tabela 1 são apresentados os resultados obtidos aplicando-se os algoritmos de análise dos pesos dos termos no corpus C1. Para esse corpus de seis categorias, é possível observar uma pequena diferença entre os resultados.

Tabela 1. Percentual de instâncias classificadas corretamente no corpus C1

tf	*tf-idf*	$tf\text{-}idf_{norm}$	$tf\text{-}idf_{norm.ig}$	rw_o	$rw_{e.idf}$
69.7	69.6	72.5	*72.6*	72.3	*68.1*

O método de $tf\text{-}idf_{norm.ig}$ apresentou resultados melhores, classificando corretamente 72,6% das instâncias. No entanto, os valores apresentados não são suficientes para afirmar superioridade. Apesar disso, vale destacar que o método original do caminhamento aleatório (rw_o) apresentou resultados melhores que o método de *tf-idf*. O método de $rw_{e.idf}$ se demonstrou o pior nos experimentos.

A Tabela 2 apresenta as medidas de precisão, revocação e área da curva Roc, para cada categoria do corpus C1.

Tabela 2. Precisão, Revocação e área da curva Roc por categoria do corpus C1

	tf	*tf-idf*	$tf\text{-}idf_{norm}$	$tf\text{-}idf_{norm.ig}$	rw_o	$rw_{e.idf}$
			Agricultura			
Precision	0.747	0.728	0.781	0.774	0.792	0.721
Recall	0.727	0.733	0.807	0.8	0.76	0.707
ROC Area	0.873	0.872	0.929	0.929	0.905	0.901
			Desportos			
Precision	0.742	0.748	0.732	0.733	0.697	0.702
Recall	0.747	0.753	0.8	0.787	0.767	0.787
ROC Area	0.891	0.894	0.92	0.926	0.933	0.91

Tabela 2. Precisão, Revocação e área da curva Roc por categoria do corpus C1 (cont.)

	tf	*tf-idf*	*tf-idf$_{norm}$*	*tf-idf$_{norm.ig}$*	*rw$_o$*	*rw$_{e.idf}$*
Economia						
Precision	*0.614*	*0.612*	*0.633*	*0.629*	*0.603*	*0.587*
Recall	*0.573*	*0.567*	*0.587*	*0.587*	*0.627*	*0.56*
ROC Area	*0.802*	*0.804*	*0.848*	*0.847*	*0.848*	*0.835*
Informática						
Precision	0.779	0.789	0.749	0.754	0.77	0.75
Recall	0.758	0.752	0.839	0.846	0.852	0.765
ROC Area	0.902	0.902	0.933	0.936	0.939	0.939
Sociedade						
Precision	0.53	0.534	0.607	0.612	0.595	0.507
Recall	0.58	0.58	0.453	0.473	0.48	0.46
ROC Area	0.814	0.814	0.829	0.836	0.845	0.792
Veículos						
Precision	*0.784*	*0.778*	*0.802*	*0.813*	*0.865*	*0.797*
Recall	*0.8*	*0.793*	*0.867*	*0.867*	*0.853*	*0.813*
ROC Area	*0.924*	*0.923*	*0.94*	*0.943*	*0.959*	*0.939*

Os resultados apresentados na Tabela 2 mostram que a categoria Economia foi a que apresentou os piores resultados na classificação, seguida pela categoria de Sociedade. Por outro lado, a categoria de veículos foi a que apresentou os melhores resultados, seguida de Informática, Desportos e Agricultura.

A Tabela 3 mostra os resultados dos algoritmos de análise dos pesos dos termos sobre o corpus C2. Para um corpus de quatro categorias, os resultados da Tabela 3 mostram, novamente, uma pequena diferença entre os resultados.

Tabela 3. Percentual de instâncias classificadas corretamente no corpus C2

tf	*tf-idf*	*tf-idf$_{norm}$*	*tf-idf$_{norm.ig}$*	*rw$_o$*	*rw$_{e.idf}$*
85.1	*85.1*	90.5	90.3	*90.6*	86.1

O método original do caminhamento aleatório apresentou resultados melhores, classificando 90,6% das instâncias corretamente. Mais uma vez, a diferença entre os métodos não foi representativa: as variações do método de *tf-idf* também apresentados resultados muito bons.

O método original de *tf* e *tf-idf* apresentaram os piores resultados, 85,1% das instâncias classificadas corretamente, seguidos do método de caminhamento aleatório $rw_{e.idf}$, com 86,1% de acerto.

A Tabela 4 apresenta as medidas de precisão, revocação e área da curva Roc, por categoria para o corpus C2.

Tabela 4. Precisão, Revocação e área da curva Roc por categoria do corpus C2

	tf	$tf\text{-}idf$	$tf\text{-}idf_{norm}$	$tf\text{-}idf_{norm.ig}$	rw_o	$rw_{e.idf}$
			Agricultura			
Precision	*0.827*	*0.825*	*0.902*	*0.898*	*0.902*	*0.872*
Recall	*0.827*	*0.836*	*0.898*	*0.898*	*0.898*	*0.787*
ROC Area	*0.923*	*0.927*	*0.973*	*0.968*	*0.97*	*0.961*
			Desportos			
Precision	0.837	0.836	0.89	0.894	0.893	0.858
Recall	0.867	0.862	0.902	0.898	0.893	0.916
ROC Area	0.943	0.942	0.966	0.967	0.963	0.965
			Informática			
Precision	0.862	0.858	0.898	0.897	0.911	0.835
Recall	0.835	0.835	0.902	0.897	0.915	0.857
ROC Area	0.927	0.926	0.958	0.958	0.972	0.949
			Veículos			
Precision	*0.879*	*0.887*	*0.932*	*0.924*	*0.92*	*0.881*
Recall	*0.876*	*0.871*	*0.92*	*0.92*	*0.92*	*0.884*
ROC Area	*0.947*	*0.95*	*0.97*	*0.969*	*0.972*	*0.959*

Os resultados da Tabela 4 demonstram que a categoria de Veículos foi, novamente, a mais bem classificada, seguida da categoria de Desportos. A categoria de Informática apresenta o terceiro melhor resultado, e a categoria de Agricultura apresentou o pior resultado entre as quatro categorias.

5 Conclusão

Neste trabalho, foi apresentado um estudo sobre a técnica de caminhamento aleatório aplicada à definição dos pesos dos termos, utilizando um corpus de textos da língua portuguesa.

A interpretação dos resultados sugere que, para um corpus contendo quatro categorias, o algoritmo original de caminhamento aleatório (rw_o) apresenta o melhor desempenho. Já para o corpus contendo seis categorias, o algoritmo de *tf-idf* normalizado com ganho de informação ($tf\text{-}idf_{norm.ig}$) se mostrou o melhor. Porém, a diferença entre eles é muito pequena (três décimos).

Conclui-se, assim, que o algoritmo de caminhamento aleatório cumpre o seu propósito, e se mostra bastante eficaz quando aplicado a textos em português. Como trabalhos futuros, espera-se, além de adotar outros critérios de inicialização da pontuação do vértice, estudar e aplicar as outras variações do caminhamento aleatório, como o baseado em co-ocorrência de bi-grama, proposto por Hassan, Mihalcea e Banea [5], e o baseado em cálculo de similaridade semântica, sugerido por Islam e Islam [6].

Referências

[1] Christian Aranha, Emannuel Passos. A Tecnologia de Mineração de Textos. *In RESI -Revista Eletrônica de Sistemas de Informação*, n. 2, Rio de Janeiro, 2006.

[2] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, Boston, MA, May 1999.

[3] Hsinchun Chen. Knowledge Management Systems: A Text Mining Perspective. *In International Conference of Asian Digitial Libaries*, 4., Bangalore, India, 2001.

[4] Nelson F. F. Ebecken, Maria Célia S. Lopes, Myrian C. de A. Costa. *Sistemas Inteligentes: Fundamentos e Aplicações*. Manole, São Paulo, 2003.

[5] Samer Hassan, Rada Mihalcea, Carmen Banea. Random Walk Term Weighting for Improved Text Classification. *In Workshop on Graph Based Methods for Natural Language Processing*, 2., 2006.

[6] Rafiqul lslam, Rakibul Islam. An Effective Term Weighting Method Using Random Walk Model for Text Classification. *In International Conference On Computer And Information Technology*, 11., 2008.

[7] Jianyi Liu, Jinghua Wang. Keyword Extraction Using Language Network. *In Natural Language Processing And Knowledge Engineering*, 2007.

[8] Antônio Cardoso Martins, João Miguel Marques, Paulo Dias Costa. Estudo Comparativo de Três Algoritmos De Machine Learning na Classificação de Dados Electrocardiográficos. Trabalho (Mestrado em Informática Médica) – Universidade do Porto, Porto, 2009.

[9] Leda de Oliveira Monteiro, Igor Ruiz Gómez, Thiago Oliveira. Etapas do Processo de Mineração de Textos – uma abordagem aplicada a textos em Português do Brasil. *In Workshop de Computação e Aplicações*, 26., 2006.

[10] Flaviana Regis de Oliveira. Modelos Clássicos de Recuperação de Informação: Uma análise Comparativa. 2004. 44 f. Monografia (Bacharelado em Ciência da Computação) – Centro Universitário de Belo Horizonte, Belo Horizonte, 2004.

[11] Gerard Salton, Christopher Buckley. Term-Weighting Approaches in Automatic Text Retrieval. *In Information Processing & Management,* vol. 24, pages 513-523, 1988.

[12] Yue-Heng Sun, Pilian He, Zhi-Gang Chen. An Improved Term Weighting Scheme for Vector Space Model. In Proceedings of the Third International Conference on Machine Learning and Cybernetics, 2004.

[13] Ah-Hwee Tan. Text Mining: The state of the art and the challenges. *In Proceedings Pakdd'99 Workshop on Knowledge Discovery From Advanced Databases*, pages 65-70, 1999.