Revista Brasileira de Computação Aplicada, Julho, 2020

DOI: 10.5335/rbca.v12i2.10561
Vol. 12, Nº 2, pp. 28–36
Homepage: seer.upf.br/index.php/rbca/index

ISSN 2176-6649

ARTIGO ORIGINAL

O uso de redes neurais para classificar artigos em revisões sistemáticas

Use of neural networks to classify papers in systematic reviews

Erico de Souza Veriscimo [1], João Luiz Bernardes Júnior [1], Luciano Antonio Digiampietri [1]

[1]University of São Paulo

*ericoveriscimo@usp.br; jlbernardes@usp.br; digiampietri@usp.br

Recebido: 18/01/2020. **Revisado:** 19/05/2020. **Aceito:** 07/06/2020.

Resumo

O número de alunos e de titulados no Brasil vem aumentando a cada ano. Este crescimento é extremamente necessário, pois a pesquisa é fundamental para o desenvolvimento de um país e grande parte da pesquisa mundial é desenvolvida com participação de alunos de pós-graduação. Tipicamente, uma pesquisa se inicia com uma revisão da literatura e, caso o objetivo seja conhecer o estado da arte de um determinado assunto por meio de um processo bem formulado e reprodutível, a revisão sistemática pode ser utilizada. Porém, revisões como a sistemática tendem a ser bastante rigorosas, demoradas e cansativas de ser realizadas manualmente. O objetivo deste trabalho é auxiliar na classificação dos trabalhos como a serem incluídos ou excluídos de uma revisão sistemática por meio de uma rede neural Multilayer Perceptron (MLP) maximizando a leitura dos trabalhos que interessam para a pesquisa. Foram realizados testes com dois conjuntos de dados e os resultados foram comparados com os produzidos por outros dois classificadores. A MLP teve o melhor resultado entre os métodos testados nos dois conjuntos de dados, correspondendo a uma boa escolha para este tipo de tarefa.

Palavras-Chave: revisão sistemática, classificação, MLP

Abstract

The number of graduate students in Brazil is increasing every year. This growth is extremely necessary because research is fundamental to the development of a country and much of the world research is developed with the participation of graduate students. Typically, research begins with a literature review, and if the goal is to know the state of the art of a particular subject through a well-formulated and reproducible process, a systematic review can be used. However, reviews such as the systematic one tend to be quite rigorous, time-consuming and hard to be performed manually. This paper's goal is to develop a method to automatically assist in the classification of papers to be included or excluded in a systematic review through a Multilayer Perceptron (MLP) neural network, maximizing the reading of papers that are of interest to the research. The proposed solution was evaluated with two datasets and the results were compared with those produced by two other classifiers. MLP had the best result among the methods tested in both datasets, corresponding to a good choice for this type of task.

Keywords: systematic review, classification, MLP

1 Introdução

Segundo a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) o número de alunos de pós-graduação no Brasil vem aumentando a cada ano (CAPES, 2005) e em 2014 o número de alunos ultrapassou 120 mil. O número de titulados também evoluiu para mestrado acadêmico, mestrado profissional e doutorado (CAPES, 2017). Este fato é muito importante e traz inúmeros benefícios para o país visto quanto a pesquisa é fundamental para o desenvolvimento de uma sociedade.

Tipicamente, a pesquisa científica inicia-se com uma revisão da literatura, procurando fontes, embasamento e o estado da arte sobre os temas envolvidos. Há uma revisão da literatura cujo método é mais rigoroso, denominada revisão sistemática (RS), que tem como objetivo coletar, extrair e analisar trabalhos de um determinado tema a fim de encontrar o estado da arte do assunto pesquisado, investigando possíveis desafios e lacunas a serem alvos de novas pesquisas (Galvão et al., 2004).

A condução de uma RS é realizada seguindo rigorosamente um protocolo. Neste protocolo são definidas todas as etapas da revisão, como: Quais fontes de dados serão pesquisadas, qual será a *string* de busca, quais idiomas serão aceitos, quais serão os critérios para incluir ou excluir trabalhos, quais são as perguntas de pesquisa que se deseja responder e quais serão os dados extraídos de cada trabalho (Kitchenham et al., 2009).

Atualmente, não é raro que a busca por artigos nas fontes (bibliotecas digitais) com a *string* definida retorne centenas ou até milhares de trabalhos. Com base nesta lista de artigos retornados, o passo seguinte realizado pelo pesquisador é efetuar a leitura do título e resumo de todos os trabalhos retornados, classificando-os previamente em incluídos ou excluídos na revisão. Os trabalhos previamente classificados como incluídos são então lidos na íntegra e os critérios de inclusão e exclusão são novamente aplicados. Ao final deste processo obtém-se a lista final de artigos que efetivamente serão incluídos na revisão sistemática. Concluída esta etapa, a etapa seguinte consiste em extrair as informações de interesse dos artigos e produzir uma síntese dos resultados.

Este processo de RS é bastante trabalhoso e demorado, além disso, uma pessoa pode não seguir um padrão de classificação dos artigos em vista do cansaço, ou pode até, involuntariamente, ir modificando seu padrão de classificação por conta do conhecimento adquirido com a leitura dos trabalhos.

Neste cenário, o objetivo deste trabalho é auxiliar na automatização do primeiro filtro da RS utilizando uma Rede Neural Perceptron Multicamadas (MLP) para classificar os trabalhos em incluídos ou excluídos e comparar os resultados com outros dois classificadores: NaiveBayes (Murphy et al., 2006) e DecisionStump (Fürnkranz, 2017).

Destaca-se que, apesar do foco em revisões sistemáticas, a solução proposta no presente trabalho pode também ser utilizada para ajudar a indicar trabalhos potencialmente interessantes para uma revisão tradicional.

Desta forma, imagina-se como cenário de uso da presente proposta um usuário que pretende realizar uma revisão da literatura (sistemática ou não) e que ao consultar uma ou mais bibliotecas digitais recebe uma lista de diversos trabalhos potencialmente interessante. A partir do momento que o usuário indique que ao menos um trabalho é interessante e ao menos um não é, a abordagem proposta é capaz de classificar os demais trabalhos como potencialmente interessantes (que devem ser incluídos na revisão) ou não. Conforme será discutido ao longo do presente trabalho, é possível utilizar a abordagem para a classificação dos trabalhos ou para a produção de uma pontuação a ser usada para um ranqueamento da "importância" dos artigos. Assim como para qualquer problema tradicional de classificação, quanto maior o número de instâncias de treinamento (artigos classificados pelo usuário ou para os quais o usuário indicar que concorda com a classificação do algoritmo) maior será a chance do classificador produzir um modelo mais adequado aos interesses do usuário.

O restante deste artigo está organizado da seguinte forma. A Seção 2 contém a apresentação de conceitos básicos e trabalhos correlatos. A Seção 3 descreve os conjuntos de dados, modelagem do problema e estratégias utilizadas. Na Seção 4 são apresentados e discutidos os resultados. Por fim, a Seção 5 traz as conclusões e direcionamentos para trabalhos futuros.

2 Trabalhos Correlatos

Alguns trabalhos auxiliam na automatização da RS de diversas maneiras e em diferentes etapas. O trabalho de Molléri and Benitti (2015) é direcionado para o gerenciamento das etapas da RS, criando uma ferramenta para apoiar todas as fases do processo. A ferramenta é baseada na web e auxilia no gerenciamento e na condução da revisão em três fases distintas: fase 1: Planejando a revisão; fase 2: Realização da revisão; fase 3: Produção do relatório da revisão. A Fig. 1 ilustra a interface da ferramenta.

Outra ferramenta que apoia a condução deste tipo de revisão é o StArt (Hernandes et al., 2012), que também organiza a RS em três etapas: planejamento, execução e sumarização. No planejamento o usuário deve especificar o protocolo da RS, na execução são realizadas as tarefas de identificação dos trabalhos, seleção e extração dos dados. Por fim, na sumarização, a ferramenta facilita o acesso às informações extraídas durante a tarefa de extração, provendo algumas figuras e estatísticas básicas sobre os dados extraídos ou classificados pelo usuário, e fornece um editor de texto para ajudar em uma primeira versão do documento de resumo, produzindo, de maneira automática, gráficos sobre a extração, origem dos trabalhos e aplicação dos critérios de inclusão e exclusão. A Fig. 2 ilustra alguns dos gráficos produzidos automaticamente após a classificação dos artigos pelo usuário e extração das informações de interesse.

No trabalho de Al-Zubidy et al. (2014) é apresentada uma revisão das ferramentas existentes que auxiliam na condução de revisões sistemáticas. São descritos

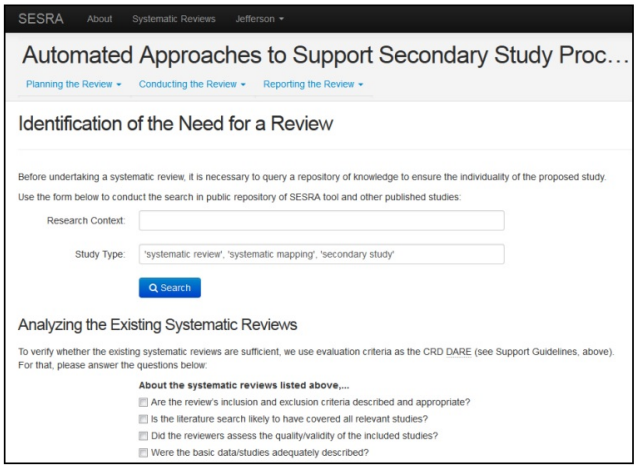


Figura 1: Interface da ferramenta (Molléri and Benitti, 2015)

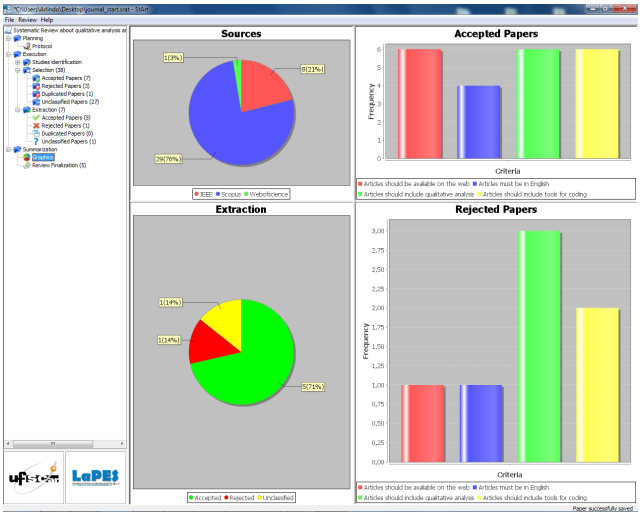


Figura 2: Cópia da Tela de Sumarização da Ferramenta StArt (Hernandes et al., 2012)

recursos, limitações para cada uma e o tipo de classificação usado pelas ferramentas. Destaca-se que a maior parte das ferramentas auxilia no processo de revisão principalmente organizando as etapas e os campos ou informações que devem ser preenchidas pelos usuários. Há, porém, poucas atividades que são executadas de maneira automática ou semi-automática pelas diferentes ferramentas.

Brito and Digiampietri (2013) apresentam um estudo sobre sistemas de recomendação de conteúdo por meio de uma revisão da literatura. Foi elucidado que o método de recomendação mais utilizado nos trabalhos é o híbrido, de forma a reunir as vantagens dos métodos de aprendizado baseado em conteúdo e colaborativo. Outra informação destacada pelos autores é que, em geral, os modelos utilizados na etapa de aprendizagem do sistema são *K-Nearest Neighbor* (KNN) (Soucy and Mineau, 2001), *Naive Bayes* (Vaidya and Clifton, 2004) e que os algoritmos *Collaborative Filtering* (CF) (Sarwar et al., 2001) e *Content-Based Filtering* (CBF) Salter and Antonopoulos (2006) trouxeram resultados mais eficazes, dentre os utilizados. Uma das possíveis aplicações de recomendação de conteúdo é a recomendação de artigos em uma revisão da literatura.

Mortenson and Vidgen (2016) propõem avaliar a literatura existente com relação ao impacto, estrutura e conteúdo. Segundo os autores, esta abordagem oferece maior validade acadêmica às revisões da literatura, pois é possível avaliar a relevância e a importância das fontes de dados que são frequentemente avaliadas nas revisões de literatura.

Uma revisão sistemática (Feng et al., 2017) sobre técnicas e ferramentas de mineração de texto para revisões sistemáticas da literatura analisou 32 estudos que descrevem o uso de técnicas de mineração de texto para apoiar o processo de RS. Neste contexto, os autores identificaram quatro aplicações principais das técnicas de mineração de texto:

- Mineração de texto visual (VTM) - VTM é um tipo de técnica de mineração de dados visual aplicada aos textos. Uma grande quantidade de textos é organizada em uma hierarquia visual ou mapa;
- Estratégia de pesquisa - tenta fornecer uma interface de consulta unificada para recuperar documentos de diferentes bancos de dados digitando uma única consulta para pesquisa;
- Classificação automatizada de documentos / textos – fornece uma classificação dos trabalhos geralmente em estudos relevantes e irrelevantes;
- Síntese de documentos - é o processo de criação automática de um resumo que objetiva manter os pontos mais importantes do documento original.

Os autores ainda relataram que a seleção (classificação) e validação automáticas de trabalhos disponíveis em uma RS continua sendo um desafio.

O trabalho de Budhi et al. (2017) não foi aplicado a revisões sistemáticas, porém está associado a classificação de produtos por texto. Os textos relacionados aos produtos são classificados em três categorias: positivo, negativo ou neutro por meio de comentários de usuários. Foram comparados 13 algoritmos de classificação

e o que obteve o melhor resultado foi o algoritmo MLP com medida-F (*F-score*) de 91,13%, conforme apresentado na Fig. 3.

Classifier Name	Experiment Type	Max. Acc. (%)	Average (%)			
			Acc	Prec	Rec	F1
Randomized Decision Trees	A	80.43	79.64	80.13	79.64	79.86
	B	74.57	73.74	74.08	73.74	73.89
	C	69.19	68.04	68.20	68.04	68.09
Gradient Boosting	A	86.98	86.64	86.42	86.64	85.15
	B	82.57	82.28	82.46	82.28	81.39
	C	78.52	78.23	75.85	78.23	74.55
Random Forest	A	87.95	87.19	86.73	87.19	86.86
	B	83.04	82.20	82.36	82.20	82.26
	C	79.27	78.52	76.48	78.52	76.19
Bagging (DT)	A	87.15	86.38	86.38	86.38	86.37
	B	82.16	81.35	81.71	81.35	81.48
	C	78.29	77.45	75.62	77.45	76.05
Ada Boost (DT)	A	87.26	86.75	86.07	86.75	85.99
	B	82.79	82.31	82.02	82.31	81.94
	C	77.88	77.59	74.30	77.59	74.55
Bagging (LSVM)	A	90.58	89.26	89.14	89.26	89.14
	B	86.96	85.40	85.34	85.40	85.33
	C	82.26	80.83	79.10	80.83	79.37
Ada Boost (LSVM)	A	90.57	89.26	89.14	89.26	89.14
	B	86.95	85.40	85.34	85.40	85.33
	C	82.25	80.83	79.10	80.83	79.36
Bagging (LR)	A	91.16	90.36	90.12	90.36	90.18
	B	87.54	86.65	86.53	86.65	86.55
	C	83.19	82.27	80.11	82.27	80.60
Ada Boost (LR)	A	89.51	88.78	88.80	88.78	88.78
	B	85.69	84.78	84.81	84.78	84.79
	C	80.56	79.44	78.49	79.44	78.89
Bagging (MLP)	A	92.11	91.21	91.08	91.21	91.13
	B	88.81	87.47	87.42	87.47	87.44
	C	84.27	83.39	82.00	83.39	82.45

Figura 3: Comparação dos classificadores de texto (Budhi et al., 2017)

De acordo com o observado na literatura correlata, selecionar trabalhos automaticamente em uma RS ainda é um desafio em aberto e, entre os classificadores baseados em texto, o algoritmo MLP se mostrou capaz de obter bom desempenho, superior aos algoritmos com os quais foi comparado. Desta forma, este trabalho busca auxiliar na primeira etapa de seleção de trabalhos de uma revisão sistemática utilizando MLP.

3 Materiais e Métodos

Esta seção descreve os conjuntos de dados, modelagem do problema e estratégias utilizadas.

3.1 Conjuntos de dados

Dois conjuntos de dados reais (curados manualmente) foram utilizados para a avaliação da solução proposta.

O conjunto de dados A (*dataset A*) foi fornecido pela professora Dra. Sarajane M. Peres e corresponde a dados de um mapeamento sistemático sobre mineração de processos.

O conjunto de dados B (*dataset B*) foi fornecido pelos autores deste artigo e é oriundo de uma revisão sistemática sobre avaliação da experiência do usuário em interação tridimensional.

Os dados contêm o título e o resumo dos trabalhos (textuais) e a classificação numérica do trabalho: um para positivo (incluído) e zero para negativo (excluído).

As Tabelas 1 e 2 descrevem algumas características destes conjuntos de dados.

Tabela 1: Características do conjunto de dados A

Conjunto de dados	A
Atributos	textuais
Valores faltando ?	não
Número de instâncias	3883
Classe (1 - sim ou 0 - não)	está incluído?
Instâncias positivas (1 - sim)	730
Instâncias negativas (0 - não)	3153

Tabela 2: Características do conjunto de dados B

Conjunto de dados	B
Atributos	textuais
Valores faltando ?	não
Número de instâncias	57
Classe (1 - sim ou 0 - não)	está incluído ?
Instâncias positivas (1 - sim)	21
Instâncias negativas (0 - não)	36

3.2 Modelagem da tarefa

Para a utilização de algoritmos de classificação tradicionais sobre os dados é necessário representar o texto do título e resumo de uma maneira estruturada. Ao longo desta seção é descrito o pré-processamento dos dados brutos, de forma a poderem ser utilizados como entrada para o algoritmo MLP. Conforme será apresentado, o pré-processamento é realizado de maneira automatizada.

Inicialmente, o título e o resumo foram concatenados em um único texto e a pontuação foi removida.

Com um único texto sem pontuação, foi realizada a remoção de *stop-words* em inglês, ou seja, foram retiradas palavras como: *the*, *a*, *of* entre outras. Após este processamento foi realizado uma radicalização (ou, mais especificamente, a remoção de sufixos) com o algoritmo Porter Stemmer (Willett, 2006), para que as palavras fiquem em seu modo raiz, por exemplo, *work*, *worked* e *working* são representadas como *work* após a execução do algoritmo. Estas etapas de pré-processamento são bastante comuns em mineração de textos e são importantes para a redução da dimensionalidade do problema, isto é, palavras que potencialmente não possuem muito significado agregado são excluídas (*stop-words*) e diferentes variações de uma mesma palavra são representadas como uma única palavra (radicalização).

Após as etapas descritas, cada texto foi representado como um *bag of words*. Isto é, cada texto foi represen-

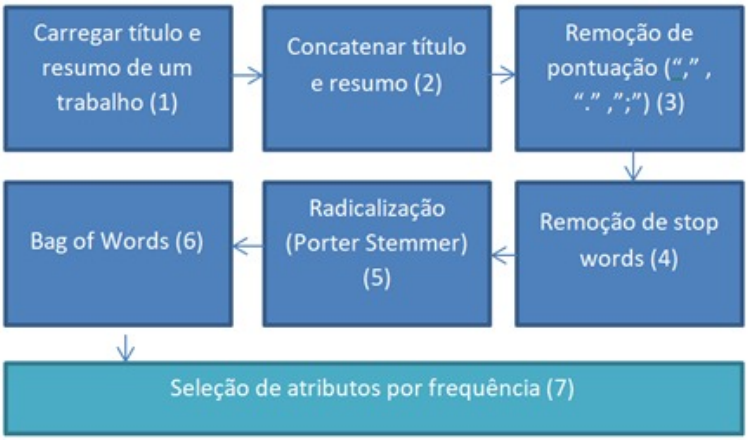


Figura 4: Etapas do pré-processamento e representação dos dados

tado como uma matriz binária de características, na qual, cada “palavra” (após remoção de *stop-words* e radicalização) corresponde a uma característica ou atributo e o valor 1 indica que o respectivo texto contém essa palavra e o valor zero indica que não contém a palavra.

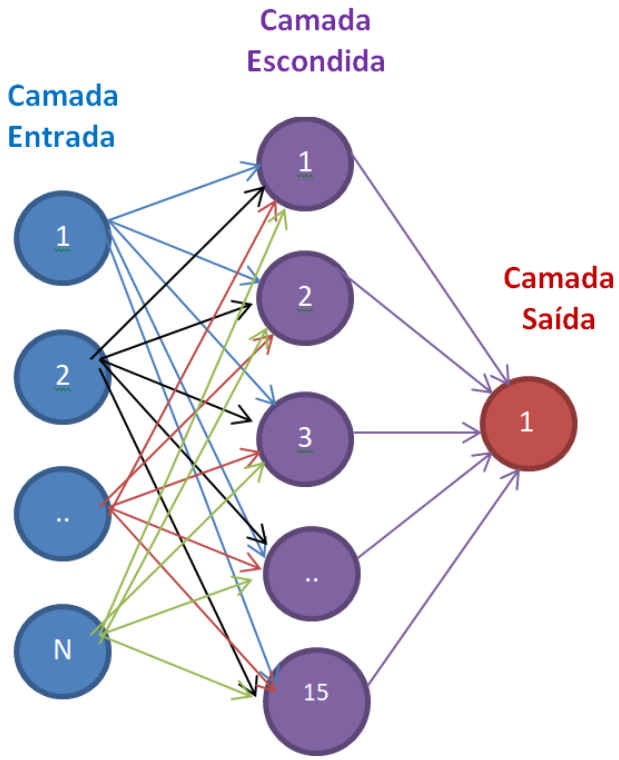
Adicionalmente, para reduzir novamente a dimensionalidade do problema excluindo as características (ou atributos) menos frequentes, foi realizada uma seleção de atributos por frequência sendo que os atributos que aparecem em mais de 100 trabalhos no conjunto de dados A e de 5 trabalhos no conjunto de dados B permaneceram no conjunto de dados e os outros foram excluídos. Estes limiares de frequência foram estabelecidos empiricamente com base na distribuição de frequência das palavras nos dois conjuntos de dados, de forma que cada conjunto de dados não tivesse mais de 100 características (ou atributos). Este processo de pré-processamento foi automatizado para garantir que o mesmo conjunto de dados seja utilizado nos diferentes classificadores. Outras alternativas são apresentadas nas considerações finais. A Fig. 4 representa todas as etapas de pré-processamento descritas.

3.3 Estratégia de parametrização

Para cada atributo (após o pré-processamento) foi criado um neurônio de entrada. Quinze neurônios foram adicionados na camada escondida e apenas um neurônio na camada de saída. A Fig. 5 ilustra a estrutura de neurônios da rede.

O ponto de parada para o treinamento foi estabelecido como a execução de 2.500 épocas ou quando o algoritmo atinge um erro médio quadrático menor que 0,0001, o que ocorrer primeiro. A taxa de aprendizado foi fixada em 0,3 e como função de ativação dos neurônios foi utilizada a função logística. Estes valores foram atribuídos empiricamente, sem haver uma busca exaustiva por parâmetros que maximizem o desempenho do algoritmo para os conjuntos de dados testados.

Figura 5: Estrutura dos neurônios da MLP



3.4 Estratégia de avaliação

Para a avaliação da classificação foi utilizada validação cruzada (Browne, 2000) com 10 subconjuntos para o conjunto A e 4 subconjuntos para o conjunto B.

A solução proposta, baseada em MLP, foi comparada com duas das estratégias mais encontradas na literatura correlata baseada em mineração de textos: o uso de algoritmos bayesianos e o uso de árvores de decisão.

Como os conjuntos de dados são desbalanceados, a acurácia não é um bom critério para avaliação, deste modo para avaliar os classificadores serão usadas: matriz de confusão, precisão, sensibilidade e medida-F (*F-score*). Estas medidas são definidas a seguir:

$$\textit{Matriz de confusão} = \frac{TP|FN}{FP|TN}$$

$$\textit{Precisão} = \frac{TP}{TP+FP}$$

$$\textit{Sensibilidade} = \frac{TP}{TP+FN}$$

$$\textit{medida } F = \frac{2 * \textit{precisão} * \textit{sensibilidade}}{\textit{precisão} + \textit{sensibilidade}}$$

Sendo: TP a contagem de predições positivas que o classificador acertou (*true positive*), FN é a contagem de predições negativas que o classificador errou (*false negative*), FP é a contagem de predições positivas que o classificador errou (*false positive*) e por fim TN é a contagem de predições negativas que o classificador acertou (*true negative*).

Adicionalmente, a curva ROC (Fawcett, 2006) será utilizada para se identificar o melhor classificador.

4 Resultados e Discussão

Neste trabalho foi investigado o desempenho dos classificadores MLP, Rede NaiveBayes e a árvore de decisão DecisionStump, com os dois conjuntos de dados apresentados na Seção 3.1. A parametrização da MLP foi apresentada na Seção 3.3. Para os outros algoritmos, foram utilizadas as implementações disponíveis no software Weka (Holmes et al., 1994) com configuração padrão dos parâmetros.

A função de ativação da MLP é uma função logística, deste modo o retorno da camada de saída não é um valor binário. Sendo assim, é preciso escolher um valor para ser o limiar transformando então a saída em binário (1 para incluído e 0 para excluído). Este cálculo é bastante simples, a partir de um limiar L e sendo S o valor de saída da rede, a conversão ocorre da seguinte maneira:

1 = se S $\geq$ L
0 = caso contrário

Uma das vantagens de se ter um valor contínuo ao invés de um binário como saída do algoritmo de classificação é a possibilidade de escolher um limiar que privilegia alguma medida específica (precisão, revocação, medida F, etc) ou mesmo abre-se a possibilidade de usar esse valor como uma possível pontuação sobre adequabilidade de cada um dos artigos avaliados na revisão. Isto é, artigos com maior pontuação são, provavelmente, mais interessantes à revisão.

A escolha dos limiares foi feita após análise das curvas ROC (Figs. 6 e 7), utilizando o critério do limiar mais próximo ao ponto máximo (100) do eixo de sensibilidade e mínimo (0) no eixo de falso positivos. Para o conjunto de dados A, o limiar definido foi de 0,25178987 e para o conjunto de dados B foi de 0,48363631.

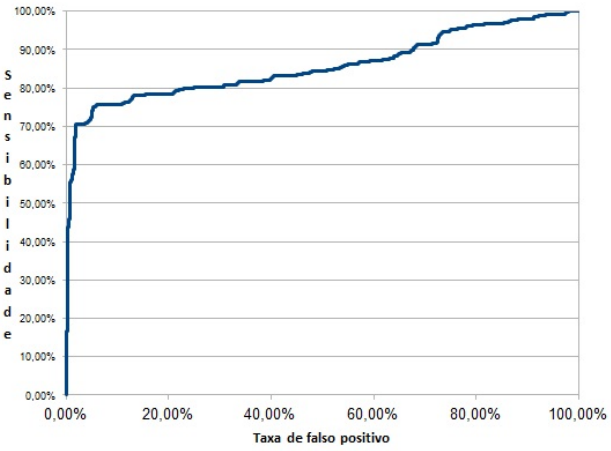


Figura 6: Curva ROC utilizada para o estabelecimento do limiar para MLP para o conjunto de dados A

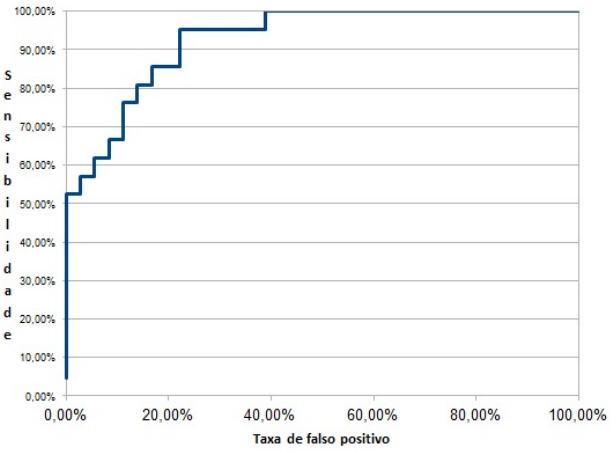


Figura 7: Curva ROC utilizada para o estabelecimento do limiar para MLP para o conjunto de dados B

Foram construídas as matrizes de confusão dos três classificadores, posteriormente utilizadas para o cálculo das demais métricas de avaliação. As matrizes são apresentadas na Fig. 8 com os resultados para o conjunto de dados A e na Fig. 9 com os resultados para o conjunto de dados B.

As demais métricas de avaliação são apresentadas nas Tabelas 3 e 4.

Devido ao uso de diferentes conjuntos de dados, não

Figura 8: Matrizes de confusão dos classificadores com o conjunto de dados A

MLP

		Predito	
		+	-
Atual	+	515	215
	-	61	3091

NaiveBayes

		Predito	
		+	-
Atual	+	491	239
	-	358	2795

DecisionStump

		Predito	
		+	-
Atual	+	360	370
	-	2280	2873

Figura 9: Matrizes de confusão dos classificadores com o conjunto de dados B

MLP

		Predito	
		+	-
Atual	+	20	1
	-	8	28

NaiveBayes

		Predito	
		+	-
Atual	+	12	9
	-	9	27

DecisionStump

		Predito	
		+	-
Atual	+	0	21
	-	2	34

Tabela 3: Resultados para o conjunto de dados A

Classificadores	Precisão	Sensibilidade	medida-F
MLP	89,41%	70,55%	78,87%
NaiveBayes	57,83%	67,26%	62,19%
DecisionStump	13,64%	49,32%	21,36%

Tabela 4: Resultados para o conjunto de dados B

Classificadores	Precisão	Sensibilidade	medida-F
MLP	71,43%	95,24%	81,63%
NaiveBayes	57,14%	57,14%	57,14%
DecisionStump	0%	0%	0%

é possível comparar diretamente os resultados da Fig. 3 com os resultados produzidos pelo presente trabalho.

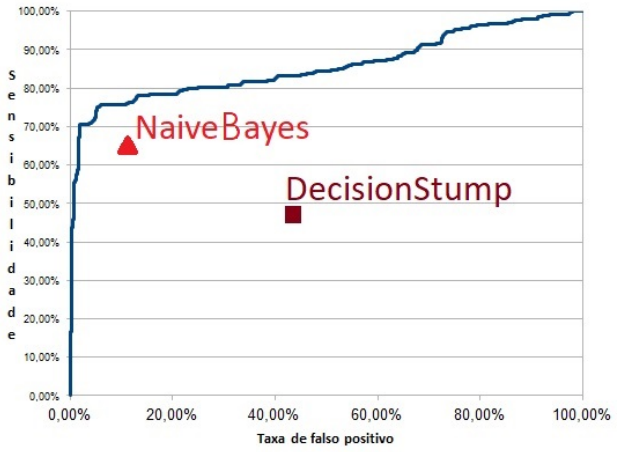
O uso do classificador baseado em Rede Bayesiana (*NaiveBayes*) apresentou resultados razoáveis para os dois conjuntos de teste, tendo um melhor resultado no conjunto de dados mais balanceado (conjunto de dados B).

O classificador com uso de árvore de decisão (*DecisionStump*) teve o pior resultado para os dois conjuntos de dados, bem como a pior localização na curva ROC.

Para as três medidas apresentadas nas Tabelas 3 e 4, a MLP apresentou os melhores resultados. Destaca-se os valores de medida F próximos a 80%.

As curvas ROC do MLP com a indicação do desempenho dos demais classificadores são apresentadas nas Figs. 10 e 11.

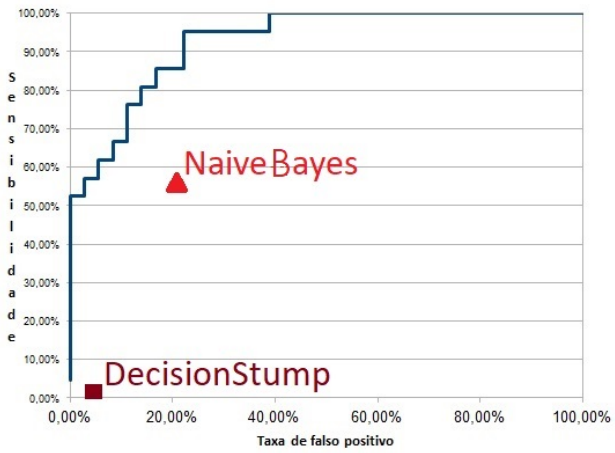
Figura 10: Curva ROC dos classificadores com o conjunto de dados A



Destaca-se nas Figs. 10 e 11 que a curva ROC do MLP está acima dos pontos de desempenho dos outros dois algoritmos testados. Vale lembrar que quanto maior a área sob a curva melhor será o desempenho do algoritmo. Isto indica que, mesmo com diferentes limiares, a solução baseada em MLP possuirá um bom desempenho, permitindo que um usuário selecione limiares mais ou menos restritivos se quiser maximizar, respectivamente, a precisão ou a revocação.

A MLP apresentou resultados melhores para os dois conjuntos de dados quando comparada com os outros classificadores testados. Destaca-se uma melhor aproximação na curva ROC que os demais classificadores,

Figura 11: Curva ROC dos classificadores com o conjunto de dados B



tanto do ponto máximo de sensibilidade quanto do mínimo em taxa de falso positivo. Para o cenário testado, o classificador DecisionStump teve um desempenho bastante ruim, não conseguindo classificar nenhum trabalho como "incluído" no conjunto de dados B. O fato dos conjuntos de dados serem desbalanceados, esparsos e não possuírem muitas instâncias (em especial o conjunto B) aumenta a possibilidade dos classificadores não conseguirem produzir um modelo com regras generalizadas, maximizando as chances da classificação de elementos como pertencentes à classe majoritária (neste caso, à classe "excluído").

5 Conclusões e Trabalhos Futuros

O presente trabalho propôs e apresentou dois estudos de caso sobre o uso de mineração de texto na seleção de artigos em revisões da literatura.

Foi proposto o uso do algoritmo MLP que, para os dois estudos de casos propostos, apresentou resultados bastante superiores aos algoritmos testados, baseados, respectivamente, em redes bayesianas e árvores de decisão.

Destaca-se que MLP foi selecionado para o presente trabalho por já ter apresentado bons resultados em outras aplicações de classificação de textos em trabalhos correlatos.

Como trabalhos futuros pretende-se: explorar outras estratégias de seleção automática de atributos (como algoritmos específicos para esta finalidade ou estratégias para atribuição de pontuações/importâncias das palavras que podem ser utilizadas como filtros); testar outras representações de dados, por exemplo com bigramas, trigramas ou *embeddings*; avaliar o desempenho de outros algoritmos de classificação.

Outras abordagens também se fazem interessantes como trabalhos futuros. Uma delas é o uso de um processo com interação do usuário sobre a relevância dos trabalhos e, a partir dessa interação, melhorar a classificação (*relevance feedback*). Alternativamente, é possível realizar o agrupamento (*clustering*) dos dados, agrupando trabalhos semelhantes e avaliar a pertinência dos grupos em relação à revisão sistemática. Por fim, é possível desenvolver algoritmos que, com base em artigos selecionados, sugiram novas palavras-chave para melhorar a *string* de busca.

Referências

Al-Zubidy, A., Carver, J. C., Hellmann, S. and Martin, M. (2014). Review of systematic literature review tools, *University Of Alabama Technical Report* .

Brito, J. and Digiampietri, L. (2013). Uma revisão acerca da recomendação personalizada de conteúdo, *Revista de Sistemas de Informação da FSMA* **12**.

Browne, M. W. (2000). Cross-validation methods, *Journal of mathematical psychology* **44**(1): 108–132. https://doi.org/10.1006/jmps.1999.1279.

Budhi, G. S., Chiong, R., Pranata, I. and Hu, Z. (2017). Predicting rating polarity through automatic classification of review texts, *2017 IEEE Conference on Big Data and Analytics (ICBDA)*, IEEE, pp. 19–24. https://doi.org/10.1109/ICBDAA.2017.8284101.

CAPES (2005). Número de pós-graduandos cresce no brasil. Disponível em http://www.capes.gov.br/36-noticias/1168-blank-73641651.

CAPES (2017). Pós-graduação brasileira teve avanço qualitativo na última década. Disponível em https://bit.ly/2XFukFd.

Fawcett, T. (2006). An introduction to roc analysis, *Pattern Recognition Letters* **27**(8): 861 – 874. https://doi.org/10.1016/j.patrec.2005.10.010.

Feng, L., Chiam, Y. K. and Lo, S. K. (2017). Text-mining techniques and tools for systematic literature reviews: A systematic literature review, *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*, IEEE, pp. 41–50. https://doi.org/10.1109/APSEC.2017.10.

Fürnkranz, J. (2017). Decision stump, *in* C. Sammut and G. I. Webb (eds), *Encyclopedia of Machine Learning and Data Mining*, Springer, p. 330. https://doi.org/10.1007/978-1-4899-7687-1_285.

Galvão, C. M., Sawada, N. O. and Trevizan, M. A. (2004). Revisão sistemática: recurso que proporciona a incorporação das evidências na prática da enfermagem, *Revista Latino-americana de enfermagem* **12**(3): 549–556. http://dx.doi.org/10.1590/S0104-11692004000300014.

Hernandes, E., Zamboni, A., Fabbri, S. and Thommazo, A. D. (2012). Using GQM and TAM to evaluate StArt-a tool that supports Systematic Review, *CLEI Electronic Journal* **15**(1): 3. Disponível em http://www.scielo.edu.uy/scielo.php?script=sci_arttext&pid=S0717-50002012000100003&nrm=iso.

Holmes, G., Donkin, A. and Witten, I. H. (1994). Weka: A machine learning workbench. http://dx.doi.org/10.1109/ANZIIS.1994.396988.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J. and Linkman, S. (2009). Systematic literature reviews in software engineering–a systematic literature review, *Information and software technology* **51**(1): 7–15. https://doi.org/10.1016/j.infsof.2008.09.009.

Molléri, J. S. and Benitti, F. B. V. (2015). Sesra: a web-based automated tool to support the systematic literature review process, *Proceedings of the 19th international conference on evaluation and assessment in software engineering*, ACM, p. 24. https://doi.org/10.1145/2745802.2745825.

Mortenson, M. J. and Vidgen, R. (2016). A computational literature review of the technology acceptance model, *International Journal of Information Management* **36**(6): 1248–1259. https://doi.org/10.1016/j.ijinfomgt.2016.07.007.

Murphy, K. P. et al. (2006). Naive bayes classifiers, *University of British Columbia* **18**: 60.

Salter, J. and Antonopoulos, N. (2006). Cinemascreen recommender agent: combining collaborative and content-based filtering, *IEEE Intelligent Systems* **21**(1): 35–41. https://doi.org/10.1109/MIS.2006.4.

Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J. et al. (2001). Item-based collaborative filtering recommendation algorithms., *Www* **1**: 285–295. https://doi.org/10.1145/371920.372071.

Soucy, P. and Mineau, G. W. (2001). A simple knn algorithm for text categorization, *Proceedings 2001 IEEE International Conference on Data Mining*, IEEE, pp. 647–648. https://doi.org/10.1109/ICDM.2001.989592.

Vaidya, J. and Clifton, C. (2004). Privacy preserving naive bayes classifier for vertically partitioned data, *Proceedings of the 2004 SIAM International Conference on Data Mining*, SIAM, pp. 522–526. https://doi.org/10.1137/1.9781611972740.59.

Willett, P. (2006). The Porter stemming algorithm: then and now, *Program: electronic library and information systems* **40**(3): 219–223. https://doi.org/10.1108/00330330610681295.