Uma Proposta para Redução do Conjunto de Treinamento Utilizando Aprendizagem Ativa

Maicon Brandão, Marcelo Acordi, Guilherme Dal Bianco

[1]Universidade Federal da Fronteira Sul
Campus Chapecó
Chapecó – SC – Brazil

{maincon.brandao,marcelopancotte}@gmail.com, guilherme.dalbianco@uffs.edu.br

Abstract. *Supervised methods are commonly used in numerous tasks, such as classification. However, supervised methods depend on creating a labeled training set to represent the dataset patterns. Identifying informative and representative instances can reduce the labeling cost. In this context, active learning aims to select more informative instances to be labeled to reduce the training set size. This paper aims to propose weights for an active learning algorithm to reduce the number of labeled instances. In other words, our goal is to reduce the impact of class imbalance by using weights for the active learning method. Preliminary experiments demonstrated that it is possible to reduce the labeled set's size without impacting the method's effectiveness.*

Resumo. *Métodos supervisionados são comumente utilizados em inúmeras tarefas como na classificação de informações. Porém, a aprendizagem do método supervisionado depende da criação de um conjunto de treinamento rotulado capaz de representar os padrões presentes na base de dados. Identificar exemplos informativos e representativos pode representar uma redução de custos. Neste contexto, a aprendizagem ativa tem como objetivo selecionar instâncias mais informativas para serem rotuladas a fim de se reduzir o conjunto de treinamento. Este artigo tem como objetivo propor pesos para um algoritmo de aprendizagem ativa para reduzir a quantidade de instâncias selecionadas. Em outras palavras, almeja-se reduzir o impacto do desbalanceamento de classes a partir da utilização de pesos para o método de aprendizagem ativa. Os experimentos preliminares demonstraram que é possível reduzir o tamanho do conjunto rotulado sem impactar na eficácia do método.*

1. Introdução

Com o aumento do uso de sistemas computacionais, o número de documentos (documentos textuais) armazenados aumenta cada vez mais. Esta mudança fez com que cada vez mais se produzissem novos estudos sobre estes documentos, com o objetivo de se extrair um maior conhecimento. No entanto, para que os documentos armazenados tenham valor agregado, pode ser necessário que os mesmos sejam classificados de acordo com suas características [Ayodele 2010]. Logo, visando facilitar a tarefa de classificação, técnicas de Aprendizagem de Máquina (AM) são utilizadas.

Métodos de aprendizado supervisionado baseiam-se na ideia de um *professor*, que apresenta ao seu *aluno* (o AM) um conjunto de exemplos o qual representa o conhecimento do ambiente, na forma [entrada, saída esperada] [Haykin 1999]. O *aluno* gera

a partir destes exemplos uma representação deste conhecimento (modelo), sendo capaz de produzir saídas corretas para novas entradas [Lorena and de Carvalho 2007]. Desta forma, o aprendizado do *aluno* é totalmente dependente dos exemplos apresentados (conjunto de treinamento), o que pode desencadear uma série de problemas.

Para que um algoritmo de aprendizado supervisionado alcance um bom desempenho, é necessário que o conjunto de treinamento seja altamente representativo. Em busca disso, tende-se a rotular (categorizar) um grande número de documentos. Esta tarefa pode ser realizada por um usuário não especialista [Bilenko and Mooney 2003]. Contudo, o número de exemplos rotulados, não garante a representatividade do conjunto de treinamento sobre a base de dados. Isto se dá ao fato de que muitos exemplos pouco informativos, ou seja documentos que quando rotulados não agregam novas informações ao modelo de aprendizado, podem ser selecionados [Dal Bianco 2014]. Além disso, geralmente o processo de rotulagem é difícil, demorado e com alto custo manual [Settles 2009].

A aprendizagem ativa tem como objetivo selecionar instâncias representativas para reduzir o esforço de rotulação. Também evita a inclusão de instâncias redundantes ao conjunto de treinamento [de Magalhães Silva 2012]. Ou seja, os exemplos selecionados são apenas os considerados de maior relevância para o aprendizado do método, ou melhor, aqueles que possuem maior informatividade sobre os demais. Desta forma, o conjunto de treinamento produzido é menor contudo equivalente em conhecimento. Pode-se dizer então que estes métodos possibilitam minimizar o esforço de rotulagem e garantem a formação de conjuntos de treinamento altamente informativos o que irá resultar em um melhor desempenho dos métodos de aprendizagem [Settles 2009]. No método de aprendizagem ativa, chamado de SSAR (*Selective Sampling using Association Rules*) por exemplo, são utilizadas regras de associação para quantizar a informatividade das instâncias para se evitar a rotulados de dados redundantes [Silva et al. 2011] (mais detalhes serão abordados na Seção 2).

Apesar dos métodos de aprendizado ativo serem efetivos na seleção de instâncias para o conjunto de treinamento, apresentam dificuldades em selecionar um número significativo de documentos pertencentes a classes com menor representatividade em repositórios de dados desequilibrados [Bianco et al. 2023]. Um repositório de dados é dito como desequilibrado quando uma ou mais classes agregam a maior parte das instâncias a elas, enquanto outra classe possui um número muito pequeno de instâncias associadas. Isto ocorre devido ao conjunto de documentos pertencentes a uma das classes ser dominante. Por exemplo, em um cenário de identificação de deduplicação em bases de dados [Dal Bianco et al. 2018], o número de duplicatas representa menos de 5% do conjunto total de instâncias [1]. Este fator, pode prejudicar o desempenho do classificador, nos casos em que os documentos pertencentes a classe menos expressiva, são os documentos a serem classificados como relevantes.

Neste artigo, é proposto uma alteração no método SSAR com objetivo de reduzir o conjunto de treinamento selecionado mantendo a eficácia do método. A abordagem tem como foco penalizar instâncias que apresentam informações redundantes com o conjunto de treinamento para evitar que as mesmas sejam enviadas para a rotulação. Os experimentos preliminares executados em duas bases de dados reais demonstram que a utilização de

[1] Neste artigo o termo instância e documento representam uma tupla na base de dados relacional.

pesos pode reduzir o conjunto de treinamento sem impactar na eficácia do método.

O restante do artigo é organizado da seguinte forma. Na Seção 2 são apresentados os principais conceitos envolvendo o trabalho, em conjunto com alguns trabalhos relacionados. Na próxima seção, é descrita a proposta deste artigo. Na Seção 4, são apresentados os experimentos realizados. Por fim, na última seção a conclusão é descrita.

2. Referencial Teórico e Trabalhos Relacionados

Nesta seção, são apresentados os principais conceitos e métodos importantes para a compreensão deste trabalho. Além disso, serão apresentados alguns trabalhos relacionados.

A Aprendizado de Máquina (AM) é um subcampo da ciência da computação que busca permitir que os computadores aprendam. Alguns dos principais de métodos de AM, de acordo com [Sarker 2021] são listados a seguir:

- Supervisionado: é utilizado em cenários onde o conjunto de dados possui um rotulo (ou classe). O método irá aprender, a partir dos padrões presentes no conjunto de treinamento, uma função para mapear a entrada em uma saída. Entre os métodos pode-se destacar o *SVM*, *Random Forest*, entre outros. Tais métodos são bastante aplicados no contexto da classificação de textos, análise de sentimento, entre outros;
- Não-Supervisionado: tem-se um conjunto de instâncias de treinamento não rotulado. O algoritmo irá descobrir padrões e similaridades entre as instâncias. Desta forma, é produzido um modelo para predição de novas instâncias. Neste tipo de aprendizado pode ser difícil avaliar a performance do modelo pelo fato de não ter dados rotulados. A clusterização e as *Regras de associação*, por exemplo, são amplamente utilizada para agrupar informações.

A Aprendizagem Ativa (AA) tem a finalidade de selecionar instâncias consideradas informativas para serem adicionadas ao conjunto de treinamento. A aprendizagem ativa tem como objetivo possibilitar que um método de aprendizado supervisionado execute satisfatoriamente com uma quantidade menor de instâncias em seu treinamento. O objetivo da AA é selecionar instâncias para formar um conjunto de treinamento informativo, sendo selecionadas de acordo com uma estratégia e rotuladas por um usuário. A AA é utilizada em tarefas de aprendizado supervisionado com escassez de instâncias rotuladas, em que é custoso em tempo ou caro obter rótulos [Settles 2010].

Tradicionalmente na AA, os dados de treino são compostos por uma pequena quantidade de exemplos rotulados pelo revisor (ou seja, professor). A partir destes dados, um modelo é treinado e utilizado na escolha de instâncias não rotuladas. As instâncias não rotuladas selecionadas tem seu rótulo aferido pelo oráculo (*e.g.,* humano revisor) e posteriormente são adicionadas ao conjunto de dados rotulados. O ciclo se repete até que um critério de parada específico seja alcançado. A seguir serão descritos alguns métodos de AA.

O método de Consulta por Comitê (*Query by Committee* é amplamente usado devido sua simplicidade [Kee et al. 2018, Zhao et al. 2006]. A ideia consiste na geração de um grupo de hipóteses (modelos de classificação) para identificar em quais instância ocorrem divergências nas predições. As instância que obtiverem uma maior incerteza são selecionadas para rotulação. Ou seja, a seleção de instâncias é baseada no desacordo

Figura 1. Exemplo de um conjunto de dados não rotulados.

Documentos não rotulados				
Doc1	A	B	C	D
Doc2	A	B	R	D
Doc3	W	S	C	D
Doc4	A	T	X	P

Figura 2. Adição da primeira instância a T.

Conjunto de Treinamento					
Documento					Rótulo
Train_Doc1	A	B	C	D	0

entre um conjunto de classificadores, que pode ser formado por Árvore de Decisão, Naïve Bayes, Artificial Neural Networks, entre outros.

A amostragem por incerteza, inicialmente proposta por [Lewis and Gale 1994], tem como objetivo selecionar as instâncias do qual se têm mais incerteza em relação ao seu rótulo. Na abordagem, um modelo probabilístico de classificação binário pode ser usado para selecionar as instâncias com probabilidade próxima de 50

Uma primeira deficiência dos métodos discutidos anteriormente de AA é que um ruído (*outlier)* pode ser considerado como uma instância informativa, ampliando o custo de rotulação. Outro ponto fraco dos métodos tradicionais de AA, é a necessidade de um conjunto de treinamento inicial para o problema da partida fria *cold start*.

O método SSAR proposto por [Silva et al. 2011] explora o uso de regras de associações para identificar instâncias não-redundantes (informativas) para serem rotuladas pelo usuário. Por redundantes entende-se instâncias que não acrescentam informações ao modelo de aprendizagem de máquina. O SSAR propõe selecionar os documentos mais informativos para serem rotulados, maximizando a diversidade enquanto o custo manual é reduzido. O SSAR tem como vantagem não demandar de um conjunto de instâncias iniciais, ou seja, o método é capaz de selecionar instâncias informativas sem a presença de um conjunto inicial de treinamento.

Inicialmente, o método SSAR seleciona o documento $U_i \in$ U com maior redundância em relação ao atual conjunto de treinamento *T*. Deste modo, a ideia é fornecer para que o modelo aprenda de forma genérica a informatividade do conjunto de dados. A Figura 1 ilustra um exemplo da aplicação do SSAR. Na figura a primeira coluna à esquerda representa o id do documento, enquanto as demais colunas simbolizam os atributos pertencentes a cada um dos documentos. O documento considerado o mais redundante em *U* é o *Doc1*, logo o mesmo é escolhido como primeiro elemento de T. Já a Figura 2 representa T após a adição do primeiro elemento (Doc1). A coluna *Documento* na figura representa cada documento atribuído a T, enquanto a coluna *rótulo* equivale ao rótulo de cada um dos documentos pertencentes a T. É importante salientar que após um documento ser adicionado a T, o mesmo não é removido de U.

A partir do critério de seleção para a primeira instância, garante-se que quando realizada a função de amostragem, parte significativa de *U* já estará representada por T, mesmo que com apenas um documento. Este fato permite de forma implícita que o método possa convergir ao fim exigindo menor esforço do usuário, além de garantir

Figura 3. Projeção produzida, a partir do exemplo, para verificação do número de regras geradas entre um documento pertencente a U e os documentos pertencentes a T.

Projeção 01					
	Train_Doc1				Nº RA
Doc1	A	B	C	D	15
Doc2	A	B	-	D	7
Doc3	-	-	C	D	3
Doc4	A	-	-	-	1

maior eficácia da função de amostragem abordada.

O relacionamento de recursos presente entre diferentes documentos pertencentes a U, pode ser representado através do uso de regras de associação. Sendo que as regras formadas podem ser representadas por $X \rightarrow Y$, onde o antecessor X representa um conjunto de qualquer mistura de recursos disponibilizados, e o consequente Y representa a sua classificação. O número de regras geradas varia de acordo com a relação que existe entre os documentos. Quanto maior o número de regras geradas entre um documento e o atual conjunto de treino, maior será a quantidade de recursos compartilhados entre estes. Logo, o SSAR procura por documentos que tenham grande representatividade sobre U e não sejam redundantes em T, ou seja gerem o menor número de regras de associação em relação a T.

Em seu funcionamento, o algoritmo gera uma projeção de T, na qual cada documento de entrada U_i é comparado a cada documento T_i gerando suas respectivas regras. Pode-se visualizar esta projeção no exemplo da Figura 3, a coluna a esquerda representa o relacionamento entre os recursos de cada documento em relação a instância já presente no conjunto de treinamento, e a coluna a direita aponta o número de regras geradas por este relacionamento. Dado que o documento escolhido para ser acrescentado a T será aquele que apresentar menor redundância entre seus parâmetros. No exemplo, o documento selecionado é o *Doc4*, pois foi o documento que resultou na menor soma de regras.

Figura 4. Segunda projeção gerada, a partir do exemplo, para verificação do número de regras geradas entre um documento pertencente a U e os documentos pertencentes a T.

Projeção 02					
	Train_Doc1				Nº RA
Doc1	A	B	C	D	15
	Train_Doc2				+
Doc1	A	-	-	-	1
	Train_Doc1				Nº RA
Doc2	A	B	-	D	7
	Train_Doc2				+
Doc2	A	-	-	-	1
	Train_Doc1				Nº RA
Doc3	-	-	C	D	3
	Train_Doc2				+
Doc3	-	-	-	-	0
	Train_Doc1				Nº RA
Doc4	A	-	-	-	1
	Train_Doc2				+
Doc4	A	T	X	P	15

Figura 5. Conjunto T gerado pela entrada U (Figura 1)

Conjunto de Treinamento					
Documento					Rótulo
Train_Doc1	A	B	C	D	0
Train_Doc2	A	T	X	P	1
Train_Doc3	W	S	C	D	0

Para cada novo elemento acrescentado a T, uma nova projeção é criada, comparando cada documento de entrada a cada instância pertencente a T. Este comportamento pode ser visto na Figura 4, que dá sequência ao exemplo, onde desta vez o documento escolhido é o *Doc3*, resultando no conjunto T mostrado na Figura 5. Este ciclo irá ocorrer até que o documento escolhido para ser adicionado a T, já esteja presente em T. Quando isto ocorre, a instância não é adicionada novamente ao conjunto de treinamento e o ciclo é rompido. Neste momento T está pronto para ser apresentado a um algoritmo de classificação.

Este método de amostragem é projetado para reduzir a redundância entre os documentos no conjunto de treinamento, resultando em um conjunto de menor tamanho, contudo similar em informatividade em relação ao conjunto de entrada. Dessa forma, o método SSAR foi utilizado como base para o desenvolvimento da proposta apresentada a seguir.

3. Abordagem Proposta

Esta seção tem como objetivo descrever as alterações aqui propostas para o método SSAR. As alterações desenvolvidas, são focadas na parte da quantização das regras. A proposta é definir uma regra de penalização para os valores dos documentos (não rotulados) que são equivalentes (mesmo valor) que os atributos de documentos já rotulados da classe majoritária, a fim de induzir a seleção de um número maior de instâncias da classe minoritária. Em outras palavras, em cenários de classificação como na filtragem de email considerados como *spam*, a classe alvo (é um email com *spam*) é substancialmente inferior em relação ao número de amostras que a classe majoritária, (não ser um *spam*) dificultando o processo de seleção de amostras significativas.

Em mais detalhes, no Algoritmo 1 é demonstrado a função desenvolvida para contabilizar a quantidade de regras para cada rótulo na projeção de uma instância. Levando em consideração que o método seja usado com duas classes (positiva e negativa). A seleção da instância que será rotulada, é sempre a que gera menos regras na projeção.

A função desenvolvida no Algoritmo 1 tem como entrada uma variável nomeada como $regras$, que armazena todas as regras geradas da projeção de uma instância com seus detalhes como suporte mínimo, rótulo, etc. No início do algoritmo (Linhas 1, 2 e 3) são atribuídos valores de inicialização às variáveis necessárias, para contar as regras positivas e negativas. No Laço (Linhas 4 à 11) ocorre a iteração sobre as regras, já a condição (Linha 5) verifica se a regra tem rótulo positivo, caso sim, incrementa o contador de regras positivas (Linha 6). Caso não, incrementa o contador de regras negativas (Linha 8). A contabilização ocorre quando acaba a quantidade de regras da variável de entrada, iteradas por u.

Algorithm 1: Quantificar número de rótulos das regras geradas da projeção de *i*.

```
Input: regras[]
qdtPositivo = 0;
qtdNegativo = 0;
u = 0;
while regras[u] do
    if regras[u][label] == 1 then
        qdtPositivo += 1;
    else
        qtdNegativo += 1;
    u++;
```

$$qtdRegras = qtdRNegativas * PESO + qtdRPositivas \quad (1)$$

A estratégia de penalização, se baseia em aumentar o número total de regras da projeção de uma instância. Sendo implementada a partir da multiplicação do número de regras do rótulo negativo de uma projeção por um *Peso*, demonstrado na Equação 1. Dessa forma, as instâncias que tiverem mais regras de rótulo negativo tendem a ter um total de regras maior por causa da penalização, forçando assim que assim não sejam selecionadas.

De forma simplista, a estratégia aqui proposta visa selecionar instâncias da classe de menor representatividade na base de dados a fim de se produzir um treinamento reduzido.

4. Experimentação

Nesta seção, a abordagem proposta será analisada experimentalmente para avaliar a eficácia e o custo de rotulação. Nesta seção, inicialmente será apresentado a configuração da base de dados e as métricas de avaliação. Por fim, será descrito a experimentação juntamente com a discussão dos resultados.

4.1. Base de Dados

Foram utilizadas duas bases de dados reais: IMDBxNetflix e DBLPxCiteseer. A base IMDBxNetflix foi criada consultando o serviço de interface público de aplicação (API) do Netflix e IMDB, que representam acervos sobre filmes [Dal Bianco et al. 2013] para se identificar instâncias duplicatas (que representam o mesmo objeto no mundo real). Foram integrados os pares através do cálculo da similaridade de atributos em comum como título, diretor e ano de lançamento. Já a base DBLPxCiteseer foi criada através da integração dos conjuntos de dados do Citeseer e DBLP. DBLPxCiteseer foi produzida usando os atributos título, autor, e ano de publicação. As bases IMDBxNetflix e DBLPxCiteseer, são compostas por 3.009 e 3.037 pares correspondentes respectivamente, a primeira com 2.011 rótulos negativos e 998 positivos, a segunda com 1.803 rótulos negativos e 1.234 rótulos positivos. As bases de dados foram rotuladas manualmente por cinco estudantes de ciência da computação [Dal Bianco et al. 2013].

4.2. Métricas de avaliação

Neste artigo, foram utilizadas métricas padrões de avaliação como precisão, revocação e F1 [Cruz 2019]. A métrica revocação computa a relação de documentos relevantes (instâncias verdadeiras-positivas) encontrados sobre o total de documentos relevantes. Já a precisão, mensura o número de acerto em relação aos documentos relevantes recuperados. Por fim, a métrica F1, produz a média ponderada entre a precisão e a revocação. Os experimentos foram repetidos 10 vezes e a média foi reportada.

4.3. Análise dos Experimentos

A seguir são apresentados os experimentos realizados com intuito de avaliar o comportamento da abordagem proposta. Nas bases de dados, duas classes estão presentes (positiva e negativa), sendo sempre a classe negativa dominante. Em geral, nos experimentos é avaliado o comportamento usando os algoritmos de classificação *SVM* e *RF*. Como treino é utilizado o conjunto de instâncias selecionadas pelo método ativo e como teste a base completa. Os experimentos foram conduzidos para avaliar o efeito das abordagens propostas sobre a seleção ativa do método SSAR.

Pesos	**SVM**			**RF**			**Treino**
	Prec	**Rev**	**F1**	**Prec**	**Rev**	**F1**	**Total**
1	0,93	0,92	0,92	0,97	0,87	0,92	200
2	0,93	0,74	0,82	0,96	0,88	0,92	222
3	0,95	0,87	0,91	0,96	0,88	0,92	143
4	0,97	0,85	0,91	0,97	0,85	0,91	106
5	0,93	0,90	0,91	0,96	0,88	0,92	51
6	0,93	0,90	0,91	0,96	0,89	0,92	46
7	0,93	0,88	0,91	0,96	0,86	0,91	42
8	0,93	0,88	0,91	0,97	0,85	0,91	42
9	0,93	0,88	0,91	0,97	0,86	0,91	40
10	0,95	0,72	0,82	0,99	0,79	0,88	31

Tabela 1. Avaliação da penalização na base de dados IMDBxNetflix.

Pesos	**SVM**			**RF**			**Treino**
	Prec	**Rev**	**F1**	**Prec**	**Rev**	**F1**	**Total**
1	0,93	0,94	0,94	0,94	0,90	0,92	177
2	0,91	0,95	0,93	0,96	0,90	0,93	102
3	0,90	0,96	0,93	0,95	0,91	0,93	90
4	0,89	0,96	0,93	0,95	0,89	0,92	81
5	0,84	0,98	0,91	0,93	0,89	0,91	76
6	0,84	0,98	0,91	0,93	0,89	0,91	76
7	0,84	0,98	0,91	0,93	0,89	0,91	76
8	0,84	0,98	0,91	0,93	0,89	0,91	76
9	0,84	0,98	0,91	0,93	0,89	0,91	76
10	0,84	0,98	0,91	0,93	0,89	0,91	76

Tabela 2. Avaliação da penalização na base de dados DBLPxCiteseer.

Nas Tabelas 1 e 2 são reportadas a variação do Peso (1-10) aplicando os algoritmos de classificação SVM e Random Forest (RF) e o número de instâncias selecionados pela abordagem propostas para compor o treinamento. Desconsiderando a penalização neutra, ou seja, a penalização por 1, que não altera o total de regras geradas. Como pode ser observado, com o uso da penalização foi possível reduzir o total do conjunto de instâncias selecionadas para a criação do conjunto de treinamento. Tal comportamento pode pode ser explicado, pelo fato de que as instâncias com um maior número de regras de rótulo negativo, foram penalizadas com mais frequência, removendo ou postergando sua seleção para o conjunto final. Avaliando a redução do treino e *F1*, o resultado mais promissor para *IMDBxNetflix* pode ser considerado a penalização por 6, já para *DBLPxCiteseer* pode ser considerado a penalização por 3. É visto que mesmo com a redução no total de instâncias selecionadas, a métrica $F1$ continua acima de 90% em pelo menos um dos classificadores, considerando até a penalização por 9 das avaliações em ambas as bases.

5. Conclusão

Este artigo teve como objetivo propor uma melhoria em um método de aprendizagem ativa para reduzir o tamanho do conjunto de treinamento manualmente rotulado. Foi utilizado o *SSAR*, método ativo proposto por [Silva 2012], para o desenvolvimento da abordagem. O SSAR tem como objetivo selecionar instâncias informativas que complementam os padrões presentes no conjunto de treinamento. A seleção é feita utilizando como critério a quantização das regras geradas usando regras de associação. Dessa forma, foi proposto a utilização de pesos para alterar artificialmente a importância de instâncias a fim de selecionar mais documentos da classe minoritária. A experimentação demonstrou que foi possível reduzir significativamente a proporção de instâncias selecionadas. Por fim, como trabalho futuro, pretende-se executar o método proposto em bases de dados com grande número de instâncias para avaliar o comportamento em tal cenário.

Referências

[Ayodele 2010] Ayodele, T. O. (2010). Types of machine learning algorithms. In *New advances in machine learning*. IntechOpen.

[Bianco et al. 2023] Bianco, G. D., Duarte, D., and Gonçalves, M. A. (2023). Reducing the user labeling effort in effective high recall tasks by fine-tuning active learning. *Journal of Intelligent Information Systems*, pages 1–20.

[Bilenko and Mooney 2003] Bilenko, M. and Mooney, R. J. (2003). Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39–48. ACM.

[Cruz 2019] Cruz, L. A. (2019). Modelo para recuperação de informação em repositórios institucionais utilizando a técnica de sumarização a partir da seleção de atributos do cassiopeia.

[Dal Bianco 2014] Dal Bianco, G. (2014). Redução do esforço do usuário na configuração da deduplicação de grandes bases de dados.

[Dal Bianco et al. 2013] Dal Bianco, G., Galante, R., Heuser, C. A., and Gonçalves, M. A. (2013). Tuning large scale deduplication with reduced effort. pages 1–12.

[Dal Bianco et al. 2018] Dal Bianco, G., Gonçalves, M. A., and Duarte, D. (2018). Bloss: Effective meta-blocking with almost no effort. *Information Systems*, 75:75–89.

[de Magalhães Silva 2012] de Magalhães Silva, R. (2012). Aprendizado ativo para ordenação de resultados.

[Haykin 1999] Haykin, S. (1999). Neural networks, a comprehensive foundation, prentice-hall inc. *Upper Saddle River, New Jersey*, 7458:161–175.

[Kee et al. 2018] Kee, S., Del Castillo, E., and Runger, G. (2018). Query-by-committee improvement with diversity and density in batch active learning. *Information Sciences*, 454:401–418.

[Lewis and Gale 1994] Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. *In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval.*

[Lorena and de Carvalho 2007] Lorena, A. C. and de Carvalho, A. C. (2007). Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, 14(2):43–67.

[Sarker 2021] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3):160.

[Settles 2009] Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

[Settles 2010] Settles, B. (2010). Active learning literature survey. *Computer Sciences Technical Report*.

[Silva et al. 2011] Silva, R., Gonçalves, M. A., and Veloso, A. (2011). Rule-based active sampling for learning to rank. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 240–255. Springer.

[Silva 2012] Silva, R. D. M. (2012). Aprendizado ativo para ordenação de resultados. *Instituto de Ciências Exatas*.

[Zhao et al. 2006] Zhao, Y., Xu, C., and Cao, Y. (2006). Research on query-by-committee method of active learning and application. In *Advanced Data Mining and Applications: Second International Conference, ADMA 2006, Xi'an, China, August 14-16, 2006 Proceedings 2*, pages 985–991. Springer.