MATHEUS VINÍCIUS TODESCATO

A NEW STRATEGY TO SEED SELECTION FOR THE HIGH RECALL TASK

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Guilherme Dal Bianco

Este trabalho de conclusão de curso foi defendido e aprovado pela banca avaliadora em: 12/5/2021.

BANCA AVALIADORA

Prof. Dr. Guilherme Dal Bianco – UFFS

Prof. Dr. Fernando Bevilacqua – UFFS

Prof. Dr. Denio Duarte – UFFS

A New Strategy to Seed Selection for the High Recall Task

Matheus Vinícius Todescato, Jean Carlo Hilger, Guilherme Dal Bianco

***Resumo*—High Recall Information Retrieval (HIRE) aims at identifying all (or nearly all) relevant documents given a query. HIRE, for example, is used in the systematic literature review task, where the goal is to identify all relevant scientific articles. The documents selected by HIRE as relevant define the user effort to identify the target information. On this way, one of HIRE goals is only to return relevant documents avoiding overburning the user with non-relevant information. Traditionally, supervised machine learning algorithms are used as HIRE' core to produce a ranking of relevant documents (e.g. SVM). However, such algorithms depend on an initial training set (seed) to start the process of learning. In this work, we propose a new approach to produce the initial seed for HIRE focus on reducing the user effort. Our approach combines an active learning approach with a raking strategy to select only the informative examples. The experimentation shows that our approach is able to reduce until 18% the labeling effort with competitive recall.**

***Index Terms*—Information Retrieval, High-Recall Information Retrieval, Active Learning, Cold start.**

I. Introdução

É Notório o fato de que cada vez mais dados são gerados. Na mesma medida que a capacidade de produzir dados eleva-se, cresce a urgência em encontrar informações pertinentes em meio a estes grandes volumes. Soluções para problemas deste gênero integram o objeto central de estudo da *IR (Information Retrieval)*, cujo objetivo é definido como a busca (em grandes coleções de dados) por determinado material que atenda à uma necessidade do usuário [1].

Sob certas circunstâncias, porém, a mera busca por informações relevantes não é suficiente para solucionar um problema. Há situações em que deseja-se obter todos os documentos informativos existentes em uma base de dados. A *HIRE (High-Recall Information Retrieval)* [2] é uma subárea da IR que tem como propósito atender à esta exigência, fornecendo ferramentas apropriadas para tal. Técnicas de HIRE são aplicadas em diversos cenários. No contexto acadêmico, vale mencionar métodos que auxiliam na revisão suportada por tecnologia (TAR - *Technology Assisted Review*), cujo pressuposto é retornar a um usuário documentos (usualmente artigos científicos), de modo a atender as necessidades preestabelecidas, geralmente, por meio de uma consulta textual.

Comumente, métodos HIRE retornam ao usuário um ranqueamento dos documentos (ordenados por relevância), fazendo uso de técnicas de aprendizado supervisionado para este fim [3]. No processo de identificação de documentos relevantes, o usuário realiza a rotulação de diversas instâncias presentes na base de dados, a fim de possibilitar o treinamento do algoritmo. É imprescindível requisitar o menor número de rotulações, mitigando o esforço por parte do usuário. Para alcançar tal objetivo, vale a utilização de aprendizado ativo, cuja eficácia mostra-se promissora [4]. A aprendizagem ativa tem como objetivo selecionar apenas documentos que possuem maior valor informativo, evitando a apresentação de documentos redundantes [5]. Assim, o esforço se dá nos documentos que têm maior probabilidade de serem relevantes ou que podem aprimorar o método. O método S-CAL (*Scalable Continuous Active Learning*) [6] usufrui desta técnica.

Dentre os diversos desafios que surgem na modelagem de sistemas HIRE, vale destacar o problema da geração do conjunto de treinamento inicial (semente). Este conjunto inicial, utilizado pelo algoritmo de aprendizado no início do processo, é primordial para a identificação de padrões que configuram um documento relevante. Esta etapa possui grande impacto no esforço despendido pelo usuário, posto que pode acelerar ou atrasar o aprendizado do algoritmo. Ou seja, se nenhum documento (ou semente) relevante é adicionado ao treinamento, o algoritmo de ranqueamento dificilmente conseguirá produzir um bom ordenamento. O problema descrito é conhecido como *cold-start* [7], estando presente não apenas no campo do HIRE mas também em algoritmos de aprendizado supervisionado em geral.

Neste trabalho, o método de S-CAL [6] foi estendido com uma abordagem para a seleção da semente inicial visando a redução do esforço manual. A proposta, chamada de S-CAL++, emprega o algoritmo BM25 para a geração de um ranqueamento (em ordem decrescente de relevância) de documentos. Posteriormente, é aplicado sobre o ranque o algoritmo de aprendizagem ativo, conhecido como SSAR [8], com o intuito de remover informações redundantes e desnecessárias, possibilitando assim um menor esforço de rotulação. Os experimentos foram realizados com a base de dados do CLEF 2017 [9], contendo artigos científicos da área de medicina. Tal experimentação apontou que o método proposto reduziu o esforço inicial no processo de seleção de semente em até 18%, quando comparado ao método base.

Na Seção II são apresentados os fundamentos teóricos, utilizados no processo de confecção desta contribuição. A Seção III trata de trabalhos correlatos a este, que visam solucionar problemas que surgem em técnicas HIRE. Na Seção IV, é apresentada e discutida a proposta de seleção de semente, aplicada ao método S-CAL. Em seguida, são apresentados os experimentos na seção V. Por fim, a Seção VI conclui o

Matheus Vinícius Todescato, Universidade Federal da Fronteira Sul (UFFS), Campus Chapecó, Brasil, matheus.todescato@estudante.uffs.edu.br.

Jean Carlo Hilger, Universidade Federal da Fronteira Sul (UFFS), Campus Chapecó, Brasil, jean.hilger@estudante.uffs.edu.br.

Guilherme Dal Bianco, Universidade Federal da Fronteira Sul (UFFS), Campus Chapecó, Brasil, guilherme.dalbianco@uffs.edu.br.

trabalho.

II. Referencial Teórico

Nesta seção, são apresentados os principais conceitos e métodos permeando a implementação sugerida. As informações aqui apresentadas são importantes para o entendimento deste trabalho.

A. Extração de Características

Documentos são tradicionalmente apresentados em formato de texto livre. Tal representação deve ser modificada, geralmente para o formato numérico, a fim de possibilitar seu processamento por algoritmos de aprendizagem de máquina. Uma das técnicas básicas existentes com este propósito é o BoW (sigla para *Bag of Words*), que consiste em contabilizar a ocorrência de palavras em um dado documento [1]. Desta forma, o documento passa a ser representado por um vetor contendo o número de ocorrências de cada palavra que o compõe.

Alguns documentos podem apresentar tamanho elevado em comparação a outros em um mesmo corpus e consequentemente o número de ocorrências de suas palavras será maior. Neste cenário, a utilização do BoW fica comprometida, uma vez que sua representação torna-se tendenciosa para os documentos mais extensos. Técnicas mais sofisticadas para extração de características não possuem tal inconveniente. É o caso do TF-IDF (*Term Frequency - Inverse Document Frequency*) que consiste em um algoritmo de extração de características que, semelhante ao BoW, utiliza a frequência das palavras para construir os vetores de característica. Porém, o TF-IDF considera também a frequência dos termos perante à base como um todo, fazendo com que o resultado seja independente do tamanho dos documentos. O valor TF-IDF de um termo é dado pelo produto do TF (*Term Frequency*) e do IDF (*Inverse Document Frequency*) deste mesmo termo [10]. Deste modo, tem-se:

$$tf_{td} = \frac{f_{t,d}}{|d|}, \tag{1}$$

onde $f_{t,d}$ denota a frequência do termo t no documento d e $|d|$ representa o tamanho do documento, ou seja, o total de termos presentes nele,

$$idf_t = log\frac{|D|}{df_t}, \tag{2}$$

onde D é a coleção completa de documentos, $|D|$ corresponde ao total de documentos na coleção e df_t configura o número de documentos que possuem o termo t.

Finalmente,

$$tf\text{-}idf = tf \times idf \tag{3}$$

Outro método tradicional para geração do ranque é a partir do BM25, cuja principal inovação foi trazer a modelos probabilísticos os valores utilizados no tf-idf [1]. Embora haja diversas variantes para o algoritmo, pode-se defini-lo, de maneira mais genérica, conforme a fórmula [1]:

$$BM25 = \sum_{t \in q} idf_t \times \frac{tf_{td}(k_1 + 1)}{k_1((1-b) + b \times (L_d/L_a)) + tf_t d}, \tag{4}$$

onde k_1 e b são parâmetros arbitrários, tf_{td} e idf_t referem-se aos termos discutidos anteriormente, q é a consulta do usuário, L_d é o tamanho (em número de termos) do documento sendo processado e L_a é a média dos comprimentos dos documentos da coleção (também, em número de termos). Após calculado, o valor do BM25 pode ser utilizado para a geração de um ranque ou até mesmo como uma característica, a ser utilizada por outro algoritmo.

B. Aprendizado Supervisionado

Algoritmos de aprendizado supervisionado tem como propósito fornecer uma função que mapeia dados para determinadas classes [1]. Para isso, é necessário que um conjunto de treinamento (conjunto de dados compostos por rótulos) suficientemente informativo seja fornecido para o aprendizado de padrões existentes [11].

Mais formalmente, seja um conjunto de dados X, definido por n características. A cada exemplar $x_i \in X$, com $x_i = (c_1, c_2, ..., c_n)$ onde c_j representa uma característica j, é associado um rótulo $y_i \in Y$. Desta forma, diz-se que um algoritmo de aprendizado supervisionado E tem como propósito encontrar valores $E(x_i) = \hat{y}_i$ tais que $\hat{y}_i - y_i \approx 0$.

Algoritmos supervisionados podem ser divididos em classificadores e regressores [11]. O primeiro tem como resultado um valor discreto, representando uma classe à qual o exemplo de entrada pertence. O segundo resultará em um valor contínuo, que representa uma possível resposta (saída) de acordo com a entrada. Entre os algoritmos baseados em técnicas supervisionadas pode-se indicar [11]: *Naïve Bayes*, *Support Vector Machines* - SVM e Redes Neurais Artificiais (ANN, do inglês *Artificial Neural Networks*).

Naïve Bayes é um algoritmo probabilístico, cuja fundamentação remete ao teorema de Bayes, por meio do qual é possível obter probabilidades condicionadas. Seu aprendizado é efetuado buscando quais termos no documento fornecem mais evidências de pertencer à determinada classe [1]. Para a aplicação do algoritmo, assume-se uma forte independência entre as características analisadas.

O algoritmo SVM visa dividir o conjunto de dados em segmentos. Esta divisão é feita encontrando um hiperplano de tal modo que, a distância entre o hiperplano e os pontos mais próximos a ele seja máxima [1]. Embora o algoritmo seja primordialmente para classificação binária (onde há apenas duas classes), é possível utilizá-lo com mais classes.

ANNs são algoritmos que possuem inspiração, sem muito rigor, no funcionamento do cérebro humano [11]. São compostos por diversos elementos chamados neurônios que relacionam-se através de cálculos matemáticos, cujos valores são obtidos na fase de treinamento. A grande vantagem de redes neurais - em especial as profundas - em relação aos demais algoritmos de *machine learning*, é seu potencial de aprendizagem na presença de bases de dados com uma elevada quantidade de características [11].

C. Aprendizado não Supervisionado

Algoritmos não supervisionados não requerem dados rotulados para o treinamento do método. Nesta categoria, instâncias x_i pertencentes a um conjunto de dados X não estão atrelados à um rótulo y, e por consequência, a classificação (ou regressão) dá-se por meio de inferências acerca de padrões encontrados nos dados [11]. Alguns algoritmos de clusterização são exemplos do emprego de aprendizado não supervisionado.

A clusterização tem como finalidade dividir o conjunto de dados em grupos (*clusters*) de modo que os exemplos dentro de um mesmo grupo sejam semelhantes entre si [11]. O algoritmo *K-means*, por exemplo, promove a geração de grupos, computando iterativamente as distâncias (Euclidiana, Manhattan, dentre outras) das instâncias para o seu centroide [1]. O centroide de um *cluster* é definido como a média das distâncias - tomadas sobre os valores das *features* - de todos os documentos contidos nele.

D. Aprendizado Ativo

Em muitas aplicações, a utilização de técnicas de aprendizado de máquina - em essência, na modalidade supervisionada - requer um alto grau de qualidade dos dados de treinamento sobre os quais o algoritmo irá operar [5]. Sobretudo, a existência de dados rotulados é imprescindível. Não obstante, há cenários, como no HIRE, em que a obtenção de rótulos representa um custo elevado, inviabilizando a utilização de técnicas supervisionadas. Assim sendo, o emprego de técnicas alternativas mostra-se de fundamental importância, dentre as quais há o aprendizado ativo.

Técnicas de aprendizado ativo partem do pressuposto de que o algoritmo irá escolher quais documentos serão rotulados, evitando inserção de instâncias não informativas junto ao treinamento [5]. De maneira sucinta, o algoritmo selecionará amostras de um conjunto de dados cuja rotulação é desconhecida, e irá solicitar para que um usuário defina o rótulo. Em seguida, tais instâncias passam a integrar o conjunto de treino (com rótulos) que será consumido por um algoritmo supervisionado.

Silva, Gonçalves e Veloso [8] propuseram um método para ranqueamento baseado em regras de associação, chamado SSAR (*Selective Sampling using Association Rules*). Regras de associação permitem identificar as relações existentes entre os valores constituintes do conjunto de dados. No SSAR, as regras de associação são utilizadas para se identificar documentos com informações redundantes, evitando a sua rotulação. Vale ressaltar que o aprendizado baseado em regras de associação visa inferir um conjunto de regras acerca da base de dados, e não tomar decisões com base em um conjunto de regras pré-estabelecidas.

O algoritmo explora o fato de haver redundâncias nas informações dos documentos de uma coleção. Sejam U e D o conjunto de documentos não rotulados e o conjunto de treino, respectivamente. Para cada documento $u \in U$ são extraídas regras de associação, e caso possua menos regras do que qualquer outro documento já em D, é requisitada a rotulação de u que passa a integrar D [8]. Ao final do método, documentos que pertencem a D serão os que possuem maior diversidade de informações, e portanto, os que devem ser rotulados.

Tabela I
EXEMPLO DE UM CONJUNTO DE DOCUMENTOS NÃO ROTULADOS (U).

Documento	Características
1	$a\ y\ c$
2	$x\ y\ z$
3	$x\ b\ z$

A título de exemplo, considera-se um conjunto de documentos como descrito na Tabela I. O algoritmo constrói uma projeção do conjunto não rotulado para selecionar instâncias a integrarem o conjunto de treino. Isto é alcançado selecionando os documentos que sejam menos redundantes em relação à D. Inicialmente, seleciona-se o documento 2, em seguida o documento 1 e por fim o documento 3. Cada vez que um documento é adicionado à D, as características redundantes são omitidas, a projeção é recalculada (no intuito de não incluir informações semelhantes às que já foram verificadas pelo usuário) e o número de regras de associação é computado novamente. Documentos com o menor número de regras serão enviados para um usuário rotular. O resultado final da projeção é exibido na Tabela II.

Tabela II
EXEMPLO DA PROJEÇÃO DO CONJUNTO NÃO ROTULADO.

Documento	Características	# de Regras
2	$x\ y\ z$	5
1	$a - c$	3
3	$- b -$	2

III. TRABALHOS RELACIONADOS

O método apresentado em [3], denominado AutoTAR (do inglês *Autonomous TAR*) tem por objetivo proporcionar uma ferramenta autônoma para o problema do TAR, de forma que não sejam necessários ajustes em parâmetros para tópicos ou base de dados específicos. Embora a autonomicidade do método o torne uma base eficaz para a implementação de outros métodos [6], [4], [12], a elevada demanda de documentos rotulados por parte do usuário é um fator prejudicial.

O algoritmo denominado S-CAL [6] é proposto como uma melhoria ao AutoTAR, aspirando uma maior escalabilidade em grandes conjuntos de dados, ao passo que esforço de rotulação reduzido e elevado *recall* são mantidos. Tal feito é possível pois assume-se que uma sub-amostra de documentos representa grande parte dos constituintes da amostra original.

Desse modo, à cada iteração, é requisitada a rotulação de uma sub-amostra dos documentos, considerados os mais promissores a representarem informações relevantes.

A estrutura de funcionamento do S-CAL é descrita na Fig. 1). Primeiramente, é construído um documento relevante sintético para a semente inicial (Passo 1 e Passo 2). O

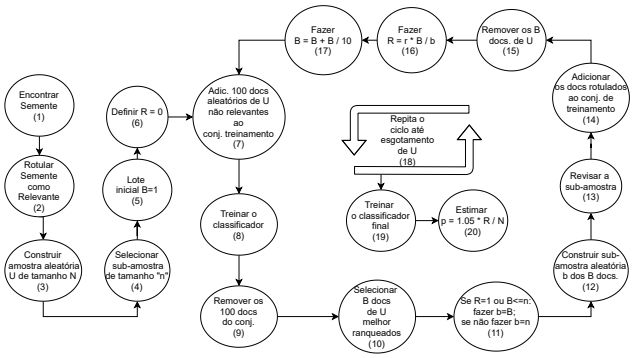

Figura 1. Estrutura de funcionamento S-CAL [6].

tamanho do lote de documentos começa em 1 (Passo 5). Como mostrado no Passo 3, é construída uma amostra aleatória "U" da população de documentos (ao invés de utilizar a base inteira), e dessa amostra é selecionada uma sub-amostra "n" (Passo 4). Ao conjunto de treinamento também se adiciona 100 documentos aleatórios, porém vindo da amostra "U". É então treinado o classificador (Passo 7 e Passo 8) e aplicado nele o conjunto dos documentos não rotulados a fim de ranqueá-los. Então são removidos os 100 documentos adicionados aleatoriamente (Passo 9). O lote de revisão B aumenta em 10% a cada ciclo (Passo 17) e quando B fica maior do que "n" ou "R" (iniciado em 0 no Passo 6 e calculado novamente no Passo 16) é selecionado apenas uma sub-amostra aleatória de tamanho "n" dos documentos de B. Dessa forma, não é mais necessário rotular todo o lote (Passos 10, 11, 12 e 13). Essa sub-amostra continua a ser adicionada ao conjunto de treinamento a cada ciclo (Passo 14). Os documentos "B" de "U" também são removidos mesmo não sendo completamente rotulados (Passo 15). Ao final do ciclo, onde "U" estará esgotado (Passo 18), é treinado o classificador pela última vez (Passo 19) e estimando um valor "p" (Passo 20) que corresponde ao limite ou ponto final, delimitando uma parte do ranque do classificador para mostrar ao usuário.

Em [13] é proposto um método baseado em aprendizado ativo, chamado de *Fast2*, para a busca sistemática na literatura no contexto do HIRE. O diferencial deste trabalho é que são propostas novas formas de abordar os desafios relacionados à geração de treinamento inicial, o erro humano no processo de revisão e o ponto de parada para o método. O Fast2 traz uma abordagem baseada em aprendizado ativo com utilização de métodos para construção da semente inicial. Para isso, foi utilizado o método de ranqueamento baseado no BM25 (descrito na Seção II), onde, após gerado o ranking são selecionados lotes de 10 documentos mais bem classificados para rotulação, revisando até que se encontre algum documento relevante. Esse documento relevante é então utilizado como semente para iniciar o processo de treinamento do modelo.

O Fast2, semelhante ao AutoTar, utiliza uma abordagem incremental na qual a cada ciclo um lote de documentos é avaliado. Tal lote é treinado utilizando o algoritmo SVM, empregando uma estratégia de predição de erros humanos e um estimador de *recall* que para o método quando o valor esperado foi atingido.

O trabalho apresentado em [14] tem como foco a geração de semente inicial para a construção do treinamento. A abordagem tem como base dois módulos principais: (A) um gerador de pseudo-documento que leva em consideração a informação da semente para pré-treinar uma rede neural; (B) um módulo de autotreinamento com documentos reais não rotulados utilizando a rede treinada pelos pseudo-documentos. Nos experimentos realizados os autores identificaram que a abordagem tem uma performance significativamente melhor do que os métodos bases (TF-IDF, LDA, etc [10]). No entanto, um problema identificado é a falta de integração entre as informações diferentes das sementes.

O método aqui proposto explora a geração de treinamento inicial (semente) utilizando a geração de ranqueamento acoplada a aprendizagem ativa com regras de associação como diferencial para minimizar o impacto da escolha da semente inicial.

IV. Proposta S-CAL++

Conforme já descrito, o método S-CAL utiliza como documento inicial relevante (ou semente), a consulta realizada pelo usuário. No entanto, não há garantia de que uma semente sintética contenha termos similares aos documentos relevantes presentes na base de dados. Neste trabalho, é proposta uma nova abordagem para seleção da semente explorando a criação de ranque e a aprendizagem ativa.

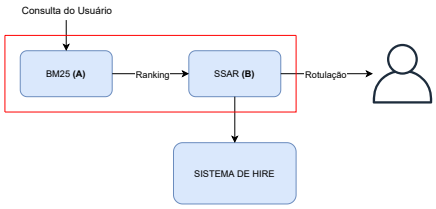

Figura 2. Proposta de gerador de semente.

A Fig. 2 ilustra uma visão geral da proposta (destacando em vermelho a contribuição). A técnica para geração de ranqueamento BM25 (A), é incorporada ao início do método para identificar os documentos mais promissores de acordo com a consulta do usuário. O método BM25 atribui a cada documento uma pontuação, utilizando para tal a frequência dos termos da consulta presentes no documento. Assim, o método SSAR (B) é aplicado para remover os documentos redundantes do ranque, ou seja, evitar que o usuário receba documentos similares e com informações não relevantes. Desta forma, com uma semente contendo características similares aos demais documentos relevantes, a convergência do método S-CAL pode ser alcançada antes, diminuindo o esforço de rotulação.

O SSAR é essencial para que não se desperdice esforço do usuário ao analisar o ranque gerado pelo BM25 com documentos não relevantes. Dado que a consulta do usuário pode ser pouco informativa, em alguns cenários um substancial número de documentos podem ser processados até que se encontre o primeiro relevante. No entanto, o SSAR é um método custoso em termos computacionais, devido a necessidade de recalcular a cada documento rotulado a informatividade daqueles que

ainda não foram rotulados (conforme descrito na seção II). Devido a isto, o método proposto fornece ao SSAR apenas lotes de N documentos do ranque, diminuindo consideravelmente o custo de processamento do mesmo. Caso dentro de um lote não seja encontrado um documento relevante, é então selecionado os próximos N, usando a ordem do ranque, até encontrar um documento relevante.

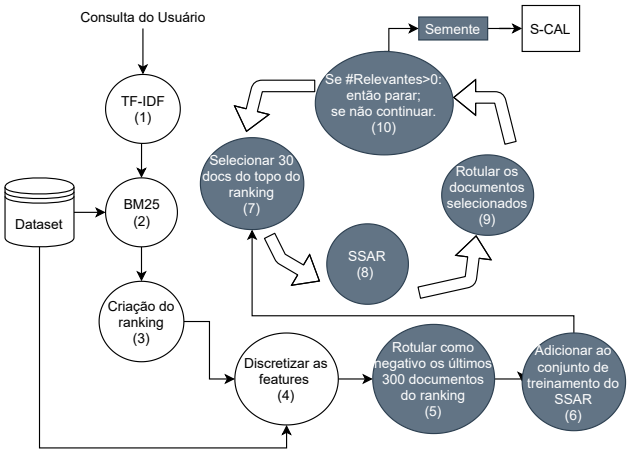

Figura 3. Estrutura de funcionamento do gerador de semente.

O funcionamento detalhado do S-CAL++ é ilustrado da Fig. 3, onde os passos destacados em cinza representam a contribuição deste trabalho. Inicialmente, é utilizado o método TF-IDF nos termos da consulta do usuário para transformalo em um vetor de características (Passo 1). Em seguida, o método BM25 é aplicado em toda a base de dados (Passo 2) para se criar um ranqueamento a partir da similaridade dos documentos com a consulta do usuário (Passo 3). Ou seja, o ordenamento do ranque será baseado na similaridade dos termos da consulta do usuário em relação aos documentos presentes na base de dados.

Para que seja possível utilizar o SSAR, é necessário discretizar as *features* (Passo 4). Após, são então selecionados os últimos N documentos do ranking e rotulados como negativos (Passo 5). Esses N documentos são adicionados ao conjunto de treinamento para que o SSAR identifique os padrões dos possíveis documentos não relevantes, evitando a sua rotulação (Passo 6). Em seguida, são selecionados os N documentos pertencentes ao topo do ranque, e enviados para o processamento do SSAR (Passo 7). O SSAR identifica os documentos mais informativos (Passo 8). Os documentos selecionados são então rotulados pelo usuário (Passo 9). Se pelo menos um dos documentos rotulados pelo usuário for relevante, então o processo de seleção de semente é finalizado. Caso contrário, será executado novamente o Passo 7 escolhendo os próximos N documentos do ranking para ser processado pelo SSAR. O ciclo será executado até que se encontre um documento relevante. É importante salientar que o SSAR tem como função evitar que todos os N documentos sejam submetidos para o usuário, reduzindo assim o esforço para se encontrar a semente inicial. Para isso, o SSAR descarta os documentos similares, que estão no topo do ranque, que já foram rotulados como não relevantes pelo usuário.

V. EXPERIMENTOS

Nesta seção, serão descritos os experimentos realizados com objetivo de avaliar e comparar a abordagem proposta. Primeiramente, será apresentada as características da base de dados. Em seguida, será apresentado como foi implementada a extração de características, a configuração dos experimentos, e as métricas aplicadas. Por fim, são apresentados e discutidos os resultados encontrados nos experimentos.

A. Base de Dados

O conjunto de dados é composto por um total de 125.464 documentos, contendo 20 tópicos. A prevalência (taxa de documentos relevantes sobre o total da base de dados dos tópicos varia de 0,002% até 0,49%, ou seja, há entre 2 e 619 documentos relevantes para uma determinada consulta, representando uma quantidade extremamente baixa. Isso impõe uma dificuldade ainda maior na tarefa de HIRE.

Cada tópico é associado a uma consulta de usuário, representada por uma frase no idioma inglês - mesmo idioma dos documentos da base. Tais consultas têm o intuito de simular uma busca realizada por um usuário durante a pesquisa por documentos relacionados a determinado assunto [15]. A Tabela III retrata um exemplo de uma consulta para o tópico $tr19$.

Tabela III
EXEMPLO DE CONSULTA PARA O TÓPICO $tr19$

Tópico	**Consulta**
$tr19$	Urine tests for Down syndrome screening.

B. Extração de características

A base de dados utilizada encontra-se no formato texto, consequentemente faz-se necessário o uso de técnicas de extração de características que permitam usufruir ao máximo dos algoritmos de aprendizado e ranqueamento. Vale considerar que devido ao fato do algoritmo S-CAL compreender o núcleo do algoritmo S-CAL++, os métodos de extração de características de ambos são muito similares, como descrito a seguir.

Primeiramente, utilizou-se da técnica *Bag of Words* (BoW) para uma conversão de texto para números. No entanto, para evitar a perda de informações com termos pouco relevantes, recorreu-se ao uso do TF-IDF, conforme descrito na Seção II-A. Também, empregou-se o algoritmo SVD (*Singular Value Decomposition*) cujo objetivo é reduzir a dimensão dos atributos, para possibilitar discretização dos mesmos.

C. Configuração dos experimentos

Os experimentos foram realizados com objetivo de avaliar o comportamento do método proposto S-CAL++ em relação ao método S-CAL. A intuição é avaliar se a nova abordagem para a geração do treinamento inicial causa impactos positivos no processo.

Os algoritmos foram executados 5 vezes, considerando sempre todos os tópicos, com exceção dos tópicos $tr3$ e $tr4$, cujo número de documentos relevantes demonstrou-se ser substancialmente baixo e ambos os métodos não convergiram. Assim sendo, os resultados foram tomados como sendo a média de todas estas execuções.

D. Métricas

No intuito de validar o desempenho do método proposto, buscou-se por métricas que resultem em informações conclusivas no contexto da tarefa de HIRE. A principal delas, o *recall*, consiste na relação entre os documentos relevantes encontrados e o total de documentos relevantes presentes na base [1], cuja representação matemática é exposta na Equação 5.

$$Recall = \frac{\text{\# relevantes retornados}}{\text{\# total de relevantes}} \tag{5}$$

Além deste, aplicou-se o esforço de rotulação, que trata do valor absoluto de documentos manualmente rotulados pelo usuário, durante todo o processo. Analisando tais métricas, é possível inferir a qualidade do ranque final retornado para o usuário - no sentido de quantos documentos retornados são de fato relevantes. De modo geral, almeja-se atingir um *recall* elevado concomitantemente a um esforço de rotulação mínimo.

Para comparar estatisticamente os valores de *recall* foi utilizado testes de significância estatística (teste-t) com um intervalo de confiança de 95%.

E. Análise do custo de rotulação para se encontrar o primeiro documento relevante

Este experimento teve como objetivo comparar o custo de rotulação para encontrar o primeiro documento relevante (ou seja, a primeira semente positiva). Idealmente, quanto antes a semente positiva for selecionada, menor será o custo de rotulação com documentos não-relevantes. Em outras palavras, é importante que a abordagem HIRE seja capaz de selecionar a semente positiva o quanto antes possível para se iniciar o processo de aprendizagem do método ativo. O experimento foi realizado comparando três abordagens: (*i*) S-CAL++ (BM25 + SSAR); (*ii*) BM25 puro (sem aprendizagem ativa), inspirado no método FAST2; e (*iii*) a geração de semente sintética usando a consulta do usuário, baseado no S-CAL.

A Figura 4 ilustra o esforço de rotulação para os métodos testados. É notável que são utilizados, na grande maioria dos tópicos, menos de 15 documentos para encontrar a semente positiva (documento relevante) tanto no S-CAL++ quanto no BM25 puro, enquanto a utilização de semente sintética se mostra menos eficiente com um custo elevado de rotulação. Porém, no tópico 17, que representa um caso onde a prevalência é extremamente baixa, os métodos requisitaram um volume maior de documentos rotulados. O método de semente sintética não tem resultado pois não foi capaz de encontrar um documento relevante durante o processo. Neste cenário, o método S-CAL++ foi capaz de encontrar um documento relevante e reduziu em 149 o número de documentos rotulados em comparação ao BM25 puro. Em média, o S-CAL++ reduziu, em comparação com o BM25, o número de documentos rotulados em 55% e para o tópico 17, em específico, foi obtido uma redução de cerca de 67%.

É importante notar que o BM25 puro se mostrou um relevante método de ranqueamento e que para certos casos consegue satisfazer as necessidades. Porém, quando a prevalência é extremamente baixa, o método de aprendizado ativo se mostra promissor para mitigar o custo de rotulação. Já o método de semente sintética tem uma eficácia reduzida em cenários na qual a prevalência é muito baixa.

F. Análise do recall no processo de treinamento

Um ponto importante que deve ser salientado é a evolução da *recall* ao longo do processo de treinamento. Ou seja, quanto antes o usuário obtiver os documentos relevantes, menor será o tempo gasto para realizar a consulta. Além disso, se o usuário só acessar documentos não relevantes no início do processo, a convergência do método será lenta, demandando um elevado esforço de rotulação.

Neste experimento, o objetivo é avaliar se o método proposto de seleção de semente auxilia na recuperação de documentos relevantes durante o processo de treinamento do S-CAL. Dessa forma, foi comparando o método S-CAL (com a semente sintética) e o S-CAL++ (com a semente gerada pela abordagem proposta). A tabela IV detalha o *recall* de cada tópico para ambos os algoritmos. Como pode ser observado, o S-CAL++ é equivalente ou superior na grande maioria dos tópicos. De acordo com a mesma tabela, é perceptível que a semente gerada auxilia a recuperar mais documentos relevantes em relação às sementes sintéticas, resultando em um ganho de até 9%. A única exceção foi o tópico $tr12$ que apresentou uma perda de 2%. Urge uma exploração mais aprofundada para compreender a razão de tal perda.

A Fig. 5 consolida os resultados, mostrando a diferença na evolução da curva de *recall* média e do número de documentos rotulados ao longo do processo de treinamento do S-CAL e do S-CAL++ para os tópicos. Como pode ser observado, o ganho de *recall* acontece antes, assim como o joelho da curva, onde o método já não encontra mais nenhum documento relevante. Assim, o SCAL++, neste requisito, apresenta resultados superiores na grande maioria dos tópicos após a finalização do processo de aprendizagem.

Por fim, este experimento demonstrou que a geração de semente do S-CAL++ foi capaz de auxiliar no processo de treinamento, recuperando em média mais documentos relevantes que o método comparado.

G. Custo de rotulação vs. recall

O objetivo deste experimento é avaliar o custo de rotulação final (após o processo de treinamento e da geração do ranque final) e o *recall* ao acoplar o método S-CAL++, em relação ao S-CAL.

Na Fig. 6 são exibidos o esforço de rotulação (Eixo Y, esquerda), o *recall* (Eixo Y, direita) e o tópico com o respectivo número de documentos relevantes no Eixo X. Como pode ser observado, os métodos são equivalentes em relação ao *recall*,

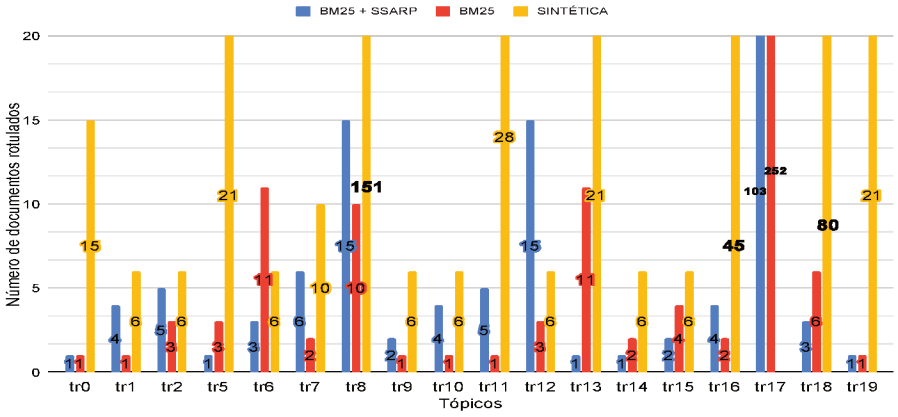

Figura 4. Comparação das abordagens S-CAL++, BM25 e sintética para identificação do primeiro documento relevante.

Tabela IV
COMPARAÇÃO DO *Recall* DOS MÉTODOS COMPARADOS PARA CADA TÓPICO.

	S-CAL	S-CAL ++
tr0	0,93	0,93
tr1	0,70	0,75
tr2	0,65	0,66
tr5	0,92	0,93
tr6	0,70	0,76
tr7	0,75	0,77
tr8	0,63	0,86
tr9	0,94	0,96
tr10	0,83	0,87
tr11	0,96	0,96
tr12	0,84	0,82
tr13	0,93	0,96
tr14	0,98	0,98
tr15	0,76	0,77
tr16	0,58	0,67
tr18	0,38	0,42
tr19	0,46	0,50

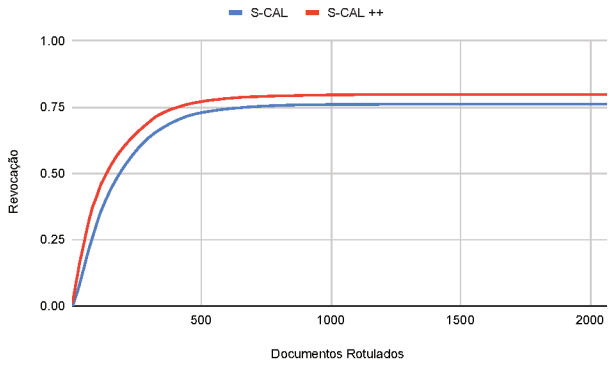

Figura 5. Curvas de crescimento médio do *recall* no processo de treinamento

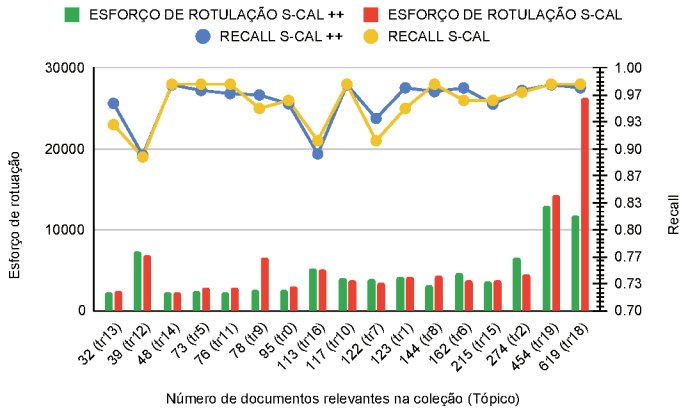

Figura 6. *Recall* vs. custo de rotulação.

no entanto, o custo de rotulação varia consideravelmente. O S-CAL é melhor em 7 tópicos, enquanto o S-CAL++ em 9, tendo apenas 1 tópico com empate. Um ponto a destacar é a substancial redução no custo de rotulação do tópico $tr18$ e o $tr9$, onde o método proposto tem uma eficiência maior.

Uma visão geral do custos dos processos comparados é oferecida na Tabela V, onde é apresentada a média para todos os tópicos em relação ao *recall* e o custo de rotulação. Ambos os métodos têm a mesma média de *recall* de 96%. Isto é devido ambos os métodos só encerrarem a execução quando o *recall* mínimo de 95% ser satisfeito. O S-CAL++ tem uma redução de cerca de 1000 documentos no esforço de rotulação, o que representa uma contração de 18%. Isso pode ser explicado observando o ranque final gerado, que é maior no S-CAL.

Por fim, estes experimentos demonstraram que a seleção de uma semente mais informativa auxilia substancialmente na redução do esforço do usuário. Além disso, foi possível mensurar na experimentação que o S-CAL++ é capaz de aprimorar o processo de treinamento, recuperando mais documentos relevantes no início do processo, se comparado ao S-CAL. Dessa forma, o treinamento gerado no S-CAL++ permite que o número total de documentos recuperados (retornados ao usuário como relevantes) seja menor, em média, que o S-CAL.

VI. CONCLUSÃO

Este artigo teve como objetivo apresentar um método para aprimorar o processo de identificação da semente inicial para o problema da HIRE, aplicado sobre o algoritmo S-CAL. O método proposto, chamado de S-CAL++, combina a geração

Tabela V
MÉDIA DE ESFORÇO DE ROTULAÇÃO E *Recall*

	Esforço manual	*Recall*
S-CAL	5846	96%
S-CAL ++	**4777**	96%

de ranqueamento com a aprendizagem ativa. O ranqueamento é utilizado para encontrar documentos promissores, já a aprendizagem ativa tem como função reduzir o número de documentos redundantes rotulados pelo usuário. A experimentação realizada demonstrou que o S-CAL++ foi capaz de reduzir em até 18% no esforço empenhado pelo usuário. Além disso, foi possível constatar que a geração de uma semente informativa auxilia na identificação de documentos relevantes com menor esforço do usuário se comparado ao método base. Todavia, a investigação de técnicas mais efetivas para geração da semente faz-se necessária, de modo a refinar os resultados obtidos e possibilitar a busca pela semente em bases de dados na qual os documentos relevantes são raros.

Nos próximos trabalhos pretende-se explorar técnicas de aprendizado profundo, que mostram-se poderosas para o processamento de dados não estruturados. Assim, compete examinar o impacto de tais técnicas para o método. Além disso, novos experimentos serão desenvolvidos em outros domínios (documentos envolvendo notícias, patentes, entre outros) para se avaliar o comportamento do S-CAL++.

AGRADECIMENTOS

Esse projeto foi financiado com recursos do Edital 459/GR/UFFS/2019.

REFERÊNCIAS

[1] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2018.
[2] A. Roegiest, "On design and evaluation of high-recall retrieval systems for electronic discovery," 2017.
[3] G. V. Cormack and M. R. Grossman, "Autonomy and reliability of continuous active learning for technology-assisted review," *arXiv preprint arXiv:1504.06868*, 2015.
[4] Z. Yu, N. A. Kraft, and T. Menzies, "Finding better active learners for faster literature reviews," *Empirical Software Engineering*, 2018. [Online]. Available: https://doi.org/10.1007/s10664-017-9587-0
[5] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.
[6] G. V. Cormack and M. R. Grossman, "Scalability of continuous active learning for reliable high-recall text classification," pp. 1039–1048, 2016.
[7] M. Fisichella, R. Kawase, and U. Gadiraju, "Automatic classification of documents in cold-start scenarios," 2009.
[8] R. Silva, M. A. Gonçalves, and A. Veloso, "Rule-based active sampling for learning to rank," *Machine Learning and Knowledge Discovery in Databases Lecture Notes in Computer Science*, p. 240–255, 2011.
[9] E. Kanoulas, D. Li, L. Azzopardi, and R. Spijker, "Clef 2017 technologically assisted reviews in empirical medicine overview," in *CEUR Workshop Proceedings*, vol. 1866, 2017, pp. 1–29.
[10] S. Qaiser and R. Ali, "Text mining: Use of tf-idf to examine the relevance of words to documents," *International Journal of Computer Applications*, vol. 181, no. 1, p. 25–29, 2018.
[11] Y. Goodfellow and A. Courville, *Machine Learning Basics*. The MIT Press, 2016, p. 95–160.
[12] G. V. Cormack and M. R. Grossman, "Engineering quality and reliability in technology-assisted review," in *ACM SIGIR*, 2016, pp. 75–84.
[13] Z. Yu and T. Menzies, "Fast2: An intelligent assistant for finding relevant papers," *Expert Systems with Applications*, vol. 120, pp. 57 – 71, 2019.
[14] Y. Meng, J. Shen, C. Zhang, and J. Han, "Weakly-supervised neural text classification," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 983–992.
[15] E. Kanoulas, D. Li, L. Azzopardi, and R. Spijker, "Clef 2017 technologically assisted reviews in empirical medicine overview," in *CEUR Workshop Proceedings*, vol. 1866, 2017, pp. 1–29.

Matheus Vinícius Todescato é discente do curso de Ciência da Computação pela Universidade Federal da Fronteira Sul, atualmente bolsista e membro do grupo de pesquisa em aprendizagem de máquina, realizando trabalhos com foco em recuperação da informação e correção de *bugs* em código.

Jean Carlo Hilger é discente do curso de Ciência da Computação pela Universidade Federal da Fronteira Sul. É integrante do grupo de pesquisa em aprendizado de máquina.

Guilherme Dal Bianco é doutor pela Universidade Federal do Rio Grande do Sul (2012) e, atualmente, é professor adjunto da Universidade Federal da Fronteira Sul (UFFS). Seus interesses em pesquisa são: extração de informações, tratamento de consultas, e integração de dados.