UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS DE CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO

BRUNO RIBEIRO

USO DE MODELAGEM DE TÓPICOS PARA PARA SOLUCIONAR O PROBLEMA DE COLD START EM SISTEMAS DE HIGH RECALL INFORMATION RETRIEVAL

CHAPECÓ
2021

BRUNO RIBEIRO

USO DE MODELAGEM DE TÓPICOS PARA PARA SOLUCIONAR O PROBLEMA DE COLD START EM SISTEMAS DE HIGH RECALL INFORMATION RETRIEVAL

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. Denio Duarte

CHAPECÓ
2021

BRUNO RIBEIRO

USO DE MODELAGEM DE TÓPICOS PARA PARA SOLUCIONAR O PROBLEMA DE COLD START EM SISTEMAS DE HIGH RECALL INFORMATION RETRIEVAL

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. Denio Duarte

Este trabalho de conclusão de curso foi defendido e aprovado pela banca avaliadora em: 16/12/2021.

BANCA AVALIADORA

Prof. Dr. Denio Duarte – UFFS

Prof. Dr. Guilherme Dal Bianco – UFFS

Prof. Me. Geomar Schreiner – UFSC

RESUMO

Sistemas de High Recall Information Retrieval (HRIR) são utilizados quando o usuário deseja obter todos os documentos existentes que sejam relevantes à sua busca. Contudo, esses sistemas passam por uma fase de treinamento muito onerosa, na qual exige-se uma classificação manual de alguns documentos. É nessa fase que esses sistemas enfrentam o problema de *cold start*. Esse problema ocorre quando os sistemas ainda não possuem dados suficientes para selecionar e apresentar alguns documentos ao revisor – normalmente uma autoridade no assunto – que irá confirmar ou não a relevância destes. Nesse sentido, a modelagem de tópicos fornece uma solução para gerenciar, organizar e anotar grandes coleções de textos. Algumas pesquisas utilizaram essa abordagem para solucionar o problema de *cold start* em sistemas de recomendação com bons resultados. Esse trabalho propõe utilizar um modelo de tópicos probabilístico Latent Dirichlet Allocation (LDA) para solucionar o problema de *cold start* no sistema de HRIR FASTREAD.

Palavras-chave: Modelagem de tópicos. *Cold start*. Recuperação de informação. *High recall*. Aprendizado de máquina.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 TEMA

O tema deste trabalho é o uso de modelagem de tópicos na obtenção de documentos que sirvam como entrada para sistemas de High Recall Information Retrieval (Recuperação de Informação de Alta Revocação - HRIR) a fim de solucionar o problema de *cold start* que ocorre na fase de treinamento desse tipo de classificador.

1.2 PROBLEMATIZAÇÃO

As áreas jurídica, médica, registro de patentes e a própria pesquisa científica em geral são exemplos de aplicações que exigem que se obtenha todo o material relevante sobre uma determinada consulta. Na área jurídica, com a descoberta eletrônica de documentos em processos civis (*e-discovery*), na área médica, com a revisão sistemática na medicina baseada em evidências, no registro de patentes, ao consultar previamente antes de um novo registro a fim de evitar litígios futuros e na área de pesquisa científica em geral, na revisão de literatura para um determinado assunto.

Yu, Kraft e Menzies (2018) lembram que ao realizar uma revisão de literatura, por exemplo, é comum fazer uma seleção onde um grande número de artigos potencialmente relevantes é coletado por meio de uma consulta inicial (uma pesquisa por palavras-chave no *Google Scholar*, por exemplo). Em seguida esses documentos são revisados e avaliados manualmente quanto à sua relevância.

Para reduzir o esforço dessas revisões manuais, para que os pesquisadores encontrem artigos relevantes mais rapidamente e para que os novos trabalhos sejam amplamente conhecidos, são pesquisados e desenvolvidos métodos de aprendizado de máquina. Esses métodos são utilizados para criar classificadores que afastem os documentos irrelevantes usando o *feedback* dos usuários, entre eles, os sistemas de HRIR.

Segundo Roegiest (2017), sistemas de HRIR são classificadores executados quando deseja-se obter todo o material relevante sobre uma determinada consulta. Os sistemas de HRIR trabalham com grandes coleções de documentos que não possuem classificação alguma quanto ao teor do seu conteúdo, ou seja, não possuem *tags* ou rótulos atribuídos a eles que pudessem ser usados para classificá-los. Segundo Yu, Kraft e Menzies (2018), essa situação demanda uma revisão manual, realizada por uma pessoa, onde os documentos são classificados quanto à sua relevância, para que o modelo tenha uma base de aprendizado.

Essa demorada e tediosa tarefa requer a pesquisa e desenvolvimento de métodos de aprendizado não-supervisionado que classifiquem e selecionem previamente um conjunto de documentos – chamados de sementes – com maior probabilidade de serem relevantes à busca. Esse processo é feito para que seja humanamente possível revisá-los, visto que, dada a grande

quantidade de documentos disponíveis numa coleção, isso poderia se estender por longos períodos de tempo. Em sistemas de HRIR isso é conhecido como problema de *cold start* (YU; KRAFT; MENZIES, 2018).

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Propor uma abordagem de geração de sementes utilizando um modelo de tópicos probabilístico – Latent Dirichlet Allocation (LDA) – para minimizar o problema de *cold start*.

1.3.2 Objetivos Específicos

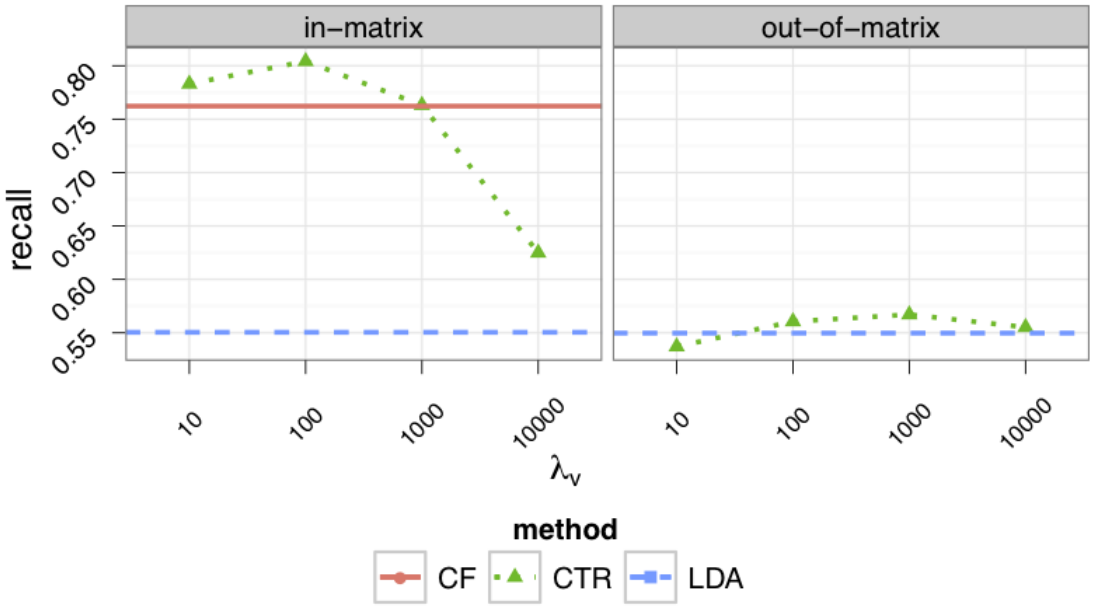
- Preparar as coleções de documentos;
- Identificar o melhor número de tópicos das coleções;
- Identificar quais *TOP-N* tópicos serão utilizados;
- Identificar o melhor número de documentos para serem utilizados como sementes;
- Treinar o modelo de tópicos desenvolvido com as coleções de documentos; e
- Avaliar os resultados.

1.4 JUSTIFICATIVA

A modelagem de tópicos vem sendo utilizada para superar o problema de *cold start* em sistemas de recomendação. Wang e Blei (2011) propuseram um algoritmo para recomendar artigos científicos aos usuários com base no conteúdo e nas classificações de outros usuários. Seu estudo mostrou que essa abordagem faz boas previsões em artigos completamente sem classificação, utilizando Collaborative Topic Regression (CTR) – abordagem que combina Colaborative Filtering (CF) e modelagem de tópicos – e LDA.

Segundo Wang e Blei (2011), CF uma técnica onde os itens são recomendados para um usuário com base em outros usuários com padrões semelhantes de itens selecionados. Já Bobadilla et al. (2012) afirma que o CF baseia-se na maneira pela qual os seres humanos tomam decisões ao longo do tempo: além da própria experiência, as decisões são baseadas também nas experiências e conhecimentos de um grupo de conhecidos. A Figura 1 apresenta os resultados obtidos pelo estudo. O gráfico mostra a revocação obtida com três métodos diferentes: CF, CTR e LDA. No quadro à direita temos os resultados "*out-of-matrix*", ou seja, quando não existem dados dos outros usuários (*cold start*). O eixo x mostra a variação do valor atribuído a um parâmetro chamado de precisão pelos autores do algoritmo.

Figura 1 – Revocação (*recall*) obtida com o algoritmo de Wang e Blei (2011): com dados dos outros usuários (à esquerda) e sem dados (*cold start*, à direita)



Fonte: Wang e Blei (2011)

Também dispostos a enfrentar o problema de *cold start* em sistemas de recomendação, Lin et al. (2013) utilizaram em seu trabalho a relação entre os perfis de aplicativos e seus seguidores do Twitter para estimar a probabilidade de um usuário-alvo gostar de um aplicativo. Para isso, construíram *pseudo-documentos* e *pseudo-palavras* usando dados de usuários, aplicativos, preferências dos usuários e seguidores. Depois, geraram tópicos latentes com LDA a partir dos *pseudo-documentos*, nos quais um grupo latente representa a combinação de interesses dos seguidores.

No FASTREAD, sistema de HRIR desenvolvido por Yu, Kraft e Menzies (2018), a tarefa de superar o problema de *cold start* é feita por uma variação do algorimo BM25, que cria um *ranking* de onde são extraídos alguns *TOP-N* documentos que classifica como os mais relevantes à consulta para apresentar ao revisor, que por sua vez confirmará (ou não) a relevância destes.

Como a modelagem de tópicos apresenta-se promissora e amplamente pesquisada para superar o problema de *cold start* em sistemas de recomendação e tendo em vista que o sistema de HRIR FASTREAD utiliza de um algoritmo de ranqueamento para esse mesmo problema, a ideia desse estudo é descobrir se um classificador baseado em modelagem de tópicos é efetivo para a geração de sementes. A abordagem aqui proposta será comparada com o algoritmo BM25 no FASTREAD.

1.5 ESTRUTURA DO TRABALHO

A sequência deste trabalho é apresentada da seguinte forma: o Capítulo 2 aborda a HRIR, apresentando o conceito de revocação e os sistemas de HRIR. No Capítulo 3 é apresentado o problema de *cold start* nas formas em que ele aparece em diferentes sistemas. O Capítulo 4 discorre sobre o aprendizado de máquina, diferenciando os tipos básicos e entrando no aprendizado não-supervisionado onde é apresentada a modelagem de tópicos e o modelo LDA. O Capítulo 5 apresenta trabalhos relacionados que foram selecionados para auxiliar na compreensão do problema e nortear os objetivos com este trabalho. O Capítulo 6 apresenta a delimitação das etapas de desenvolvimento que serão executadas a seguir e a metodologia utilizada com o propósito de atingir os objetivos definidos. Por fim, o Capítulo 7 apresenta o cronograma com os prazos das atividades planejadas para o desenvolvimento deste trabalho.

2 HIGH RECALL INFORMATION RETRIEVAL

Este capítulo apresenta brevemente o conceito de High Recall Information Retrieval pois é objeto deste trabalho. Porém, antes é apresentada a métrica de revocação (*recall*) para auxiliar na apresentação do HRIR.

2.1 REVOCAÇÃO

Segundo Davis e Goadrich (2006), em um problema de decisão binária, um classificador rotula exemplos como positivos ou negativos. A decisão tomada pelo classificador pode ser representada em uma estrutura conhecida como matriz de confusão. A matriz de confusão tem quatro categorias: (*i*) verdadeiros positivos (*true positive* - TP) são exemplos corretamente rotulados como positivos, (*ii*) os falsos positivos (*false positive* - FP) se referem a exemplos negativos rotulados incorretamente como positivos, (*iii*) negativos verdadeiros (*true negative* - TN) correspondem aos negativos rotulados corretamente como negativos, e (*iv*) os falsos negativos (*false negative* - FN) se referem a exemplos positivos incorretamente rotulados como negativos. A Tabela 1 ilustra uma matriz de confusão, a diagonal principal representa os "acertos", ou seja, aqueles exemplos que o modelo rotulou corretamente. Por sua vez, a diagonal secundária contém as predições incorretas do modelo.

Tabela 1 – Matriz de confusão

	Positivo real	Negativo real
Positivo predito	TP	FP
Negativo predito	FN	TN

Nesse contexto, a revocação (*recall*) mede a fração de exemplos positivos que foram rotulados corretamente, consequentemente, pode ser usada para saber qual a fração de exemplos positivos que o modelo não rotulou corretamente. A revocação é representada pela fórmula:

$$RECALL = \frac{TP}{TP + FN}$$

Assim, quando se deseja avaliar um modelo dando o maior peso para o número de exemplos positivos capturados em relação ao total de exemplos positivos do conjunto de dados, a métrica a ser utilizada é a revocação. Perceba que o número de exemplos positivos corretamente classificação é dividido por esse número mais os exemplos positivos não capturados.

Um exemplo simples para apresentar esta métrica é: suponha um conjunto de dados com 100 exemplos sendo 45 positivos. Suponha que seu modelo capture todos os 45 exemplos positivos mais 30 rotulados erroneamente. Nesse cenário, a revocação seria 1, ou seja, $\frac{45}{45+0}$.

2.2 SISTEMAS DE HIGH RECALL INFORMATION RETRIEVAL

Segundo Roegiest (2017), sistemas de HRIR são classificadores que caracterizam-se por alcançarem altas taxas de revocação e são utilizados quando deseja-se obter todo o material relevante sobre uma determinada consulta.

Segundo Yu e Menzies (2019), no ramo jurídico, por exemplo, advogados são pagos para revisar milhões de documentos quando tentam encontrar evidências para algum caso. As ferramentas de HRIR auxiliares nesse processo são conhecidas como descoberta eletrônica ou *e-discovery*. A medicina baseada em evidências é um outro exemplo, onde os pesquisadores revisam as publicações médicas para reunir evidências para sustentar uma determinada prática ou fenômeno médico. Na engenharia de software, o processo de seleção de artigos relevantes é chamado de seleção primária de estudo e ocorre quando os pesquisadores revisam títulos, resumos, às vezes textos completos de trabalhos de pesquisa candidatos para encontrar aqueles que são relevantes para suas questões de pesquisa.

Todos os exemplos citados têm em comum a necessidade de ter todo o material relevante disponível. Alguns desses problemas podem ser mais brandos, outros, quando envolvem sanções judiciais ou vidas humanas, são mais sensíveis. Esse é o papel dos sistemas de HRIR: evitar que alguma informação relevante seja esquecida ao realizar uma consulta. Por ser uma tarefa complexa, ocorre que a complexidade desses sistemas também é elevada, utilizando a combinação de várias técnicas e ferramentas distintas. Esse trabalho pretende contribuir com a fase inicial da execução de um sistema HRIR, quando ocorre o problema de *cold start*.

3 PROBLEMA DE COLD START

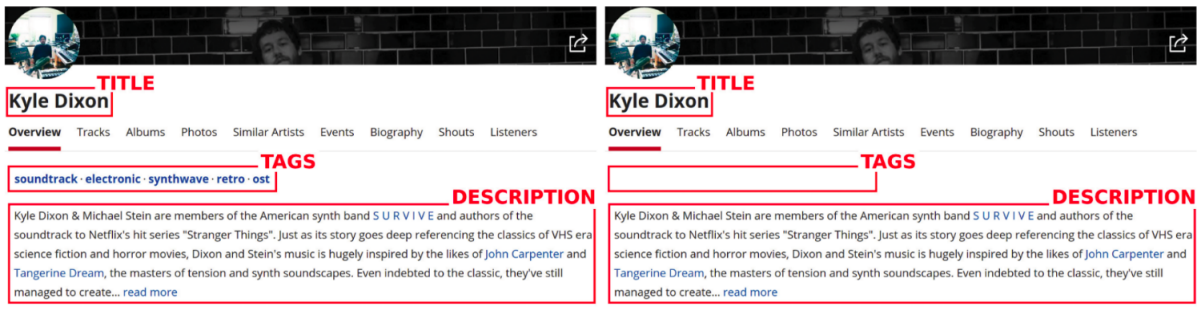
O problema de *cold start*, comum em sistemas de recomendação, ocorre quando não é possível fazer recomendações confiáveis devido a uma falta inicial de classificações. Sistemas de recomendação (Recommendation Systems - RS) permitem que sejam feitas recomendações aos usuários de um sistema em referência aos itens ou elementos nos quais esse sistema se baseia (livros, eletrodomésticos, filmes, material de *e-learning*, etc.) (BOBADILLA et al., 2012).

Segundo Bobadilla et al. (2012), a base de um RS está em seus algoritmos de filtragem colaborativa (Collaborative Filtering - CF) sendo classificada em filtragem demográfica ou filtragem baseada em conteúdo. O RS baseado em conteúdo baseia as recomendações feitas a um usuário nas escolhas que ele fez anteriormente (por exemplo, em um RS de comércio eletrônico, se o usuário já comprou livros de culinária, o RS recomendará um livro novo desse tema e que ele ainda não tenha comprado). Já o RS baseado na filtragem demográfica pressupõe que indivíduos que compartilham certas características pessoais comuns (sexo, idade, região onde vive, etc.) também têm preferências em comum. Atualmente, a CF é a tecnologia mais utilizada e estudada para RS.

Segundo Belém et al. (2019), em sistemas de recomendação, o problema de *cold start* ocorre quando há uma quantidade insuficiente de informações prévias sobre itens ou usuários (por exemplo, quando novos itens ou usuários são cadastrados no sistema), dificultando a inferência de recomendações concretas. No domínio de recomendação de *tags*, por exemplo, *cold start* ocorre na ausência de um conjunto inicial de *tags* associadas ao objeto-alvo.

Na Figura 2 pode-se observar a diferença entre um documento onde há um conjunto inicial de *tags* associadas a ele e outro onde não há, no qual ocorre o problema de *cold start*.

Figura 2 – Comparativo entre cenários sem e com *cold start*



(a) Non cold start (b) Cold start

Fonte: Belém et al. (2019)

Segundo Bobadilla et al. (2012), pode-se distinguir três tipos de problemas de *cold start*: nova comunidade, novo item e novo usuário.

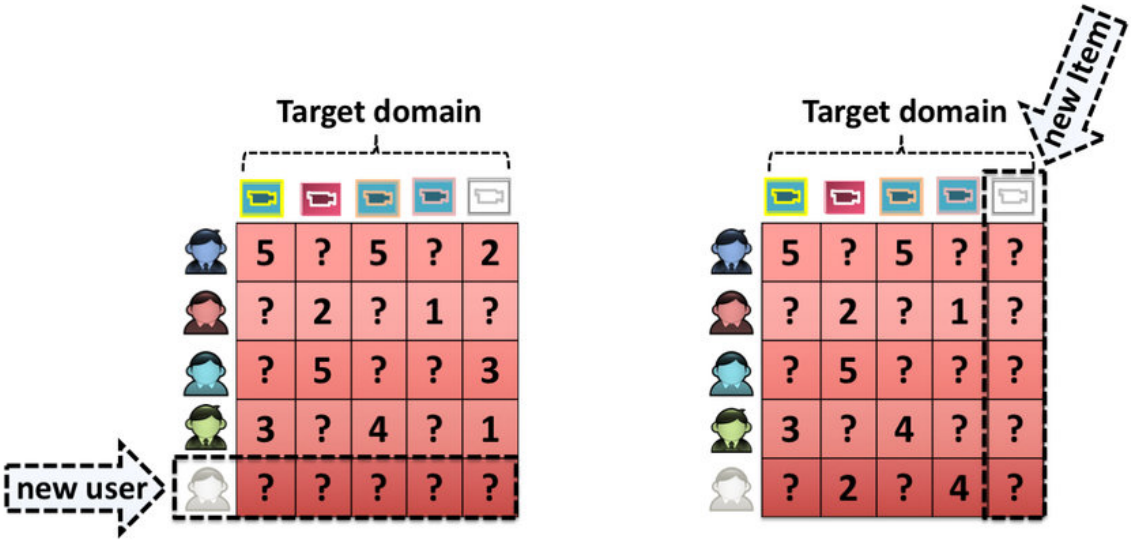
O problema da nova comunidade refere-se à dificuldade em obter, ao iniciar um RS, uma quantidade suficiente de dados (classificações) que permitem recomendações confiáveis. Assim, quando não há usuários suficientes e suas classificações, é difícil manter os novos usuários que

entram, uma vez que o RS possui conteúdo, mas não o suficiente para inferir recomendações precisas. Para os autores, as maneiras mais comuns de resolver o problema são: incentivar os usuários a fazerem as classificações; e não fazer recomendações baseadas em CF até que hajam usuários e classificações suficientes.

O problema do novo item ocorre devido ao fato de os novos itens inseridos no RS geralmente não terem classificações iniciais e, portanto, é pouco provável que sejam recomendados (BOBADILLA et al., 2012). Um item que não é recomendado passa despercebido por grande parte da comunidade de usuários que, desconhecendo-o, não o avalia. Assim, forma-se em um círculo vicioso no qual um conjunto de itens do RS é deixado de fora do processo de classificação e recomendação. Para os autores, o problema do novo item tem menos impacto no RS, no qual os itens podem ser descobertos por outros meios (campanhas de publicidade, *marketing* digital, etc.) e uma solução comum para esse problema é ter um conjunto de usuários responsáveis por classificar cada novo item no sistema.

Finalmente, o problema do novo usuário está entre as grandes dificuldades enfrentadas pelo RS em operação (BOBADILLA et al., 2012) e ocorre quando os usuários se registram, ainda não fizeram classificações e, portanto, não podem receber recomendações personalizadas com base na CF. Quando os usuários inserem suas primeiras classificações, esperam que o RS ofereça recomendações personalizadas, mas o número de classificações realizadas pode ainda não ser o suficiente para fornecer recomendações confiáveis baseadas em CF e, assim, esses usuários podem achar que o RS não oferece recomendações e podem parar de usá-lo. A Figura 3 ilustra o problema de *cold start* do novo usuário e do novo item.

Figura 3 – Ilustração do problema do *cold start* em sistemas de recomendação: problema do novo usuário (à esquerda) e do novo item (à direita)



Fonte: Bakhshandegan Moghaddam e Elahi (2019)

Em suma, RS baseiam suas estratégias na presença de dados adicionais às avaliações

reais (perfis do usuário, *tags* do usuário, publicações do usuário etc.), contudo, nem todos os bancos de dados RS possuem essas informações, ou elas não são consideradas suficientemente confiáveis, completas ou representativas (BOBADILLA et al., 2012).

No contexto de sistemas de HRIR o problema de *cold start* ocorre na fase de treinamento. Segundo Yu e Menzies (2019), alguns sistemas contam com o conhecimento dos revisores sobre o domínio para gerar um exemplo relevante inicial e depois usá-lo para iniciar o processo de aprendizado. Um problema com essa abordagem é que a qualidade do exemplo, que é definida pela experiência dos revisores, pode afetar o desempenho de aprendizado do modelo. Assim, é necessário encontrar novos exemplos caso o primeiro não for suficientemente relevante.

Para superar esse problema, alguns trabalhos utilizam outras técnicas na fase de treinamento do modelo de HRIR. Yu e Menzies (2019), por exemplo, utilizaram um algoritmo de ranqueamento para obter um conjunto menor de documentos para apresentar ao revisor e assim diminuir o esforço necessário nessa fase.

4 APRENDIZADO DE MÁQUINA

Segundo Duarte e Ståhl (2018), busca-se saber se os computadores podem aprender como nós desde que o primeiro computador foi criado. Portanto, o aprendizado de máquina é um subcampo da ciência da computação que visa fazer com que os computadores aprendam.

Duarte e Ståhl (2018) lembram que atualmente os usuários exigem que os computadores executem tarefas complexas e resolvam vários tipos de novos problemas. Enquanto isso, grandes volumes de dados são produzidos a partir de diferentes dispositivos (satélites, *smartphones*, sensores, etc.). Por estas razões, pesquisadores de diversas áreas (estatística, computação, engenharia, etc.) buscam fazer com que os computadores aprendam, propondo novas técnicas para atender às novas demandas dos usuários.

Os dados são a entrada de todo sistema de aprendizado de máquina. Segundo Duarte e Ståhl (2018), os dados contêm exemplos de um determinado domínio e os algoritmos de aprendizado de máquina generalizam esses exemplos nos dados para criar modelos matemáticos. Esses modelos podem ser usados para prever resultados com novos exemplos. Pode-se dividir os algoritmos de aprendizado de máquina em aprendizado supervisionado e não supervisionado.

4.1 SUPERVISIONADO

Segundo Duarte e Ståhl (2018), o aprendizado supervisionado pode ser aplicado quando o conjunto de dados possui um conjunto de rótulos para cada exemplo. Assim, o conjunto de dados é dividido em duas partes: X, que são as características (*features*) dos exemplos e y, que são os rótulos (*labels*). Esses rótulos ajudam o algoritmo de aprendizado a construir o modelo de previsão e a servir de guia para os alunos e pesquisadores.

Tabela 2 – Influência do vento e temperatura sob o ritmo de corrida

Velocidade do vento (Km/h)	Temperatura (°C)	Ritmo (min)
10.5	12.3	3.5
8.9	15.4	3.2
20.2	13.7	5.5
5.1	3.1	4.0

Fonte: Duarte e Ståhl (2018)

Os rótulos podem ser apresentados como valores discretos (classes) ou contínuos (valores numéricos), eles facilitam a avaliação de um modelo, pois existe uma base para comparar com os valores previstos. Segundo Duarte e Ståhl (2018), dependendo do tipo de rótulo, podemos aplicar regressores ou classificadores. Por exemplo, a Tabela 2 descreve dados sobre a influência da velocidade do vento e da temperatura no ritmo de corrida de um atleta. Nela, o rótulo (Ritmo) representa valores contínuos, logo, o conjunto de dados pode ser usado como entrada para

algoritmos de regressão supervisionados. Contudo, se alterarmos o rótulo para valores discretos (por exemplo, rápido, lento, normal, etc.), o problema se tornará de classificação.

Duarte e Ståhl (2018) lembram que todo problema de regressão pode ser transformado em um problema de classificação, dividindo os valores contínuos em classes. Assim, a primeira etapa do desenvolvimento do sistema é analisar o conjunto de dados para identificar qual abordagem deve ser utilizada.

4.2 NÃO SUPERVISIONADO

Segundo Duarte e Ståhl (2018), o aprendizado não supervisionado aplica-se quando o conjunto de dados não possui rótulos, ou seja, quando não há classificação anterior dos exemplos. Logo, o objetivo é inferir as classes ou grupos do conjunto de dados sem a ajuda dos rótulos. Para os autores, o aprendizado não supervisionado é menos objetivo que o aprendizado supervisionado, pois nele não existem rótulos para orientar o usuário na análise. Isso demanda que o domínio do conjunto de dados deve ser conhecido pelo usuário para criar modelos úteis, caso contrário, os resultados podem não ser compreensíveis.

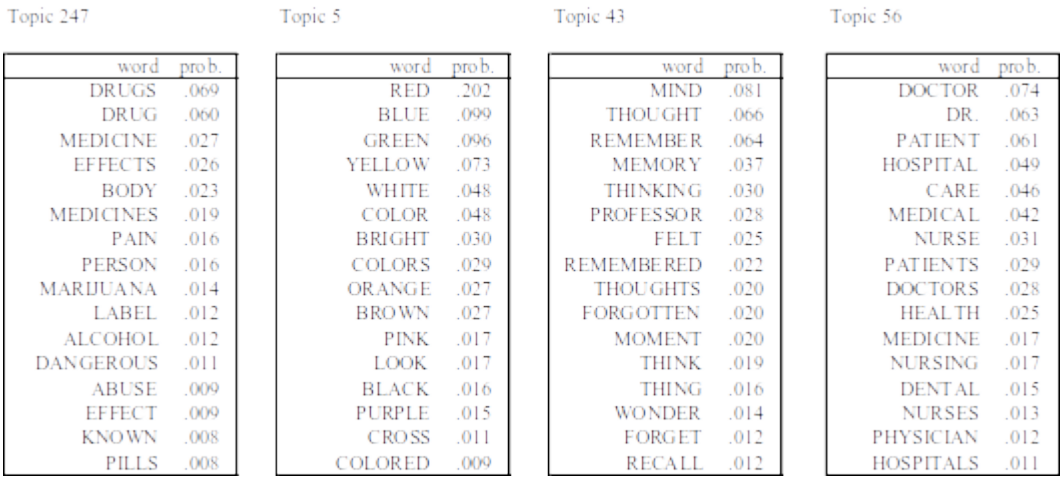
Para Duarte e Ståhl (2018), ainda que o aprendizado não supervisionado seja mais difícil de modelar do que o aprendizado supervisionado, a importância dessas técnicas tem aumentado, pois no mundo existem mais dados não rotulados do que os rotulados. Além disso, muitos problemas de aprendizado estão relacionados a problemas não supervisionados: sistemas de recomendação, classificação do comportamento do cliente em um site, segmentação de mercado, recuperação de informação, entre outros. A modelagem de tópicos é uma técnica de aprendizado de máquina não supervisionado para classificação de documentos.

4.2.1 Modelagem de Tópicos

Segundo Steyvers e Griffiths (2007), a modelagem de tópicos baseia-se na ideia de que documentos são misturas de tópicos, onde cada tópico é uma distribuição de probabilidade. Um modelo de tópicos é um modelo generativo para documentos, ou seja, especifica um procedimento probabilístico simples pelo qual os documentos podem ser gerados. Para criar um novo documento, escolhe-se uma distribuição sobre os tópicos. Então, para cada palavra nesse documento, escolhe-se um tópico aleatoriamente de acordo com esta distribuição e extrai uma palavra desse tópico. Técnicas estatísticas padrão podem ser usadas para inverter esse processo, inferindo o conjunto de tópicos responsáveis pela geração de uma coleção de documentos.

A Figura 4 mostra quatro tópicos de exemplo derivados do corpus TASA, uma coleção de mais de 37 mil passagens de texto de materiais educacionais (por exemplo, linguagem e artes, estudos sociais, saúde, ciências) coletados pela Touchstone Applied Science Associates. Segundo Steyvers e Griffiths (2007), a figura mostra as dezesseis palavras com maior probabilidade em cada tópico. As palavras nesses tópicos estão relacionadas ao uso de drogas, cores,

Figura 4 – Uma ilustração de quatro (de 300) tópicos extraídos do corpus do TASA.



Topic 247

word	prob.
DRUGS	.069
DRUG	.060
MEDICINE	.027
EFFECTS	.026
BODY	.023
MEDICINES	.019
PAIN	.016
PERSON	.016
MARIJUANA	.014
LABEL	.012
ALCOHOL	.012
DANGEROUS	.011
ABUSE	.009
EFFECT	.009
KNOWN	.008
PILLS	.008

Topic 5

word	prob.
RED	.202
BLUE	.099
GREEN	.096
YELLOW	.073
WHITE	.048
COLOR	.048
BRIGHT	.030
COLORS	.029
ORANGE	.027
BROWN	.027
PINK	.017
LOOK	.017
BLACK	.016
PURPLE	.015
CROSS	.011
COLORED	.009

Topic 43

word	prob.
MIND	.081
THOUGHT	.066
REMEMBER	.064
MEMORY	.037
THINKING	.030
PROFESSOR	.028
FELT	.025
REMEMBERED	.022
THOUGHTS	.020
FORGOTTEN	.020
MOMENT	.020
THINK	.019
THING	.016
WONDER	.014
FORGET	.012
RECALL	.012

Topic 56

word	prob.
DOCTOR	.074
DR.	.063
PATIENT	.061
HOSPITAL	.049
CARE	.046
MEDICAL	.042
NURSE	.031
PATIENTS	.029
DOCTORS	.028
HEALTH	.025
MEDICINE	.017
NURSING	.017
DENTAL	.015
NURSES	.013
PHYSICIAN	.012
HOSPITALS	.011

Fonte: Steyvers e Griffiths (2007)

mente e memória, e consultas médicas. Os autores afirmam que documentos com conteúdo diferente podem ser gerados escolhendo diferentes distribuições sobre os tópicos. Por exemplo, dando uma probabilidade igual aos dois primeiros tópicos, pode-se construir um documento sobre uma pessoa que tomou muitos medicamentos e como isso afetou a percepção das cores. Ao dar igual probabilidade aos dois últimos tópicos, pode-se construir um documento sobre uma pessoa que sofreu uma perda de memória, o que exigiu uma consulta médica. Dentre os modelos de tópicos conhecidas, temos o Latent Dirichlet Allocation.

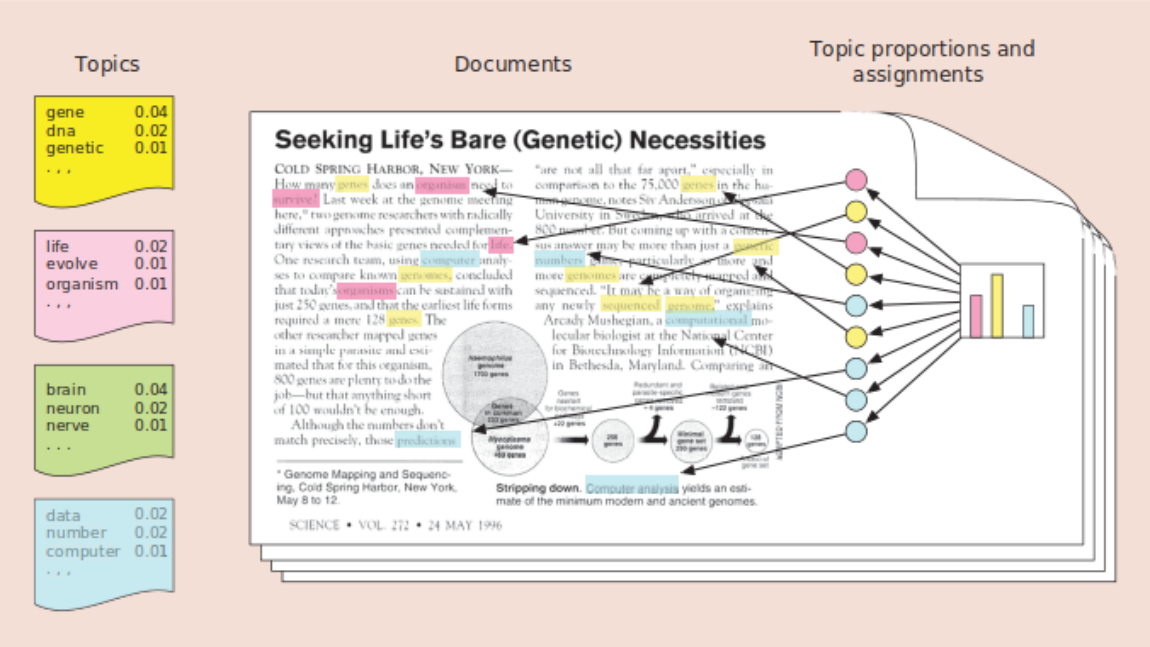
4.2.1.1 Latent Dirichlet Allocation

Segundo Blei (2012), o Latent Dirichlet Allocation (LDA) é o modelo de tópicos mais simples. A intuição por trás da LDA é que os documentos exibem vários tópicos. Por exemplo, o artigo mostrado na Figura 5, intitulado *"Seeking Life's Bare (Genetic) Necessities"*, é sobre o uso de análise de dados para determinar o número de genes que um organismo precisa para sobreviver (em um sentido evolutivo).

Na Figura 5, Blei (2012) destaca diferentes palavras usadas no artigo. As palavras sobre análise de dados são destacadas em azul; as sobre biologia evolutiva em rosa e; as sobre genética são destacadas em amarelo. Ao destacar todas as palavras do artigo – excluindo as palavras com pouco significado, como *"and"*, *"but"* e *"if"* (*stopwords*) – nota-se que o artigo combina genética, análise de dados e biologia evolutiva em diferentes proporções. A informação de que o artigo combina esses tópicos ajuda a classificá-lo em uma coleção de artigos científicos.

Segundo Blei (2012), o LDA é mais facilmente explicado por seu processo generativo, um processo aleatório e imaginário pelo qual o modelo assume que os documentos surgiram. Funciona da seguinte forma: define-se formalmente um tópico para ser uma distribuição em um vocabulário fixo. Por exemplo, o tópico de genética possui palavras sobre genética e o tópico de

Figura 5 – Exemplo da distribuição de palavras em tópicos e dos tópicos no documento.



Fonte: Blei (2012)

biologia evolutiva possui palavras sobre esse assunto. Tecnicamente, assume-se que os tópicos são gerados primeiro, antes da geração de qualquer documento. A seguir, para cada documento, são geradas palavras em um processo com duas etapas:

a) Escolha aleatoriamente uma distribuição sobre os tópicos;

b) Para cada palavra no documento;

- Escolha aleatoriamente um tópico da distribuição sobre os tópicos na etapa 1;
- Escolha aleatoriamente uma palavra da distribuição correspondente sobre o vocabulário.

Blei (2012) lembra que esse modelo estatístico reflete a intuição de que os documentos exibem vários tópicos, onde cada documento exibe os tópicos em diferentes proporções (etapa a); cada palavra em cada documento é extraída de um dos tópicos (etapa b2), onde o tópico selecionado é escolhido na distribuição por documento sobre os tópicos (etapa b1). No exemplo, a distribuição colocaria probabilidade em genética, análise de dados e biologia evolutiva, e cada palavra é extraída de um desses três tópicos. No LDA, todos os documentos da coleção compartilham o mesmo conjunto de tópicos, mas cada documento exibe esses tópicos em proporções diferentes.

Para Blei (2012), o objetivo da modelagem de tópicos é descobrir automaticamente os tópicos de uma coleção de documentos, ou seja, "reverter"o processo generativo. Trata-se de descobrir qual é a estrutura oculta que provavelmente gerou a coleção de documentos observada. O autor ressalta que os algoritmos não têm informações sobre esses assuntos e os artigos não

são rotulados com tópicos ou palavras-chave. As distribuições de tópicos interpretáveis surgem computando a estrutura oculta que provavelmente gerou a coleção observada de documentos.

Segundo Blei (2012), o LDA faz parte do campo da modelagem probabilística. Nele, as variáveis observadas são as palavras dos documentos; as variáveis ocultas são a estrutura do tópicos; e o processo generativo é como descrito anteriormente. O problema computacional de inferir a estrutura de tópicos ocultos dos documentos é calcular a distribuição posterior, a distribuição condicional das variáveis ocultas (tópicos), dados os documentos.

Para Blei (2012), a utilidade dos modelos de tópicos vem do fato de que a estrutura de tópicos oculta inferida se assemelha à estrutura temática da coleção. Essa estrutura oculta interpretável faz anotações sobre cada documento da coleção – uma tarefa que é muito difícil de executar manualmente – e essas anotações podem ser usadas para auxiliar tarefas como recuperação de informações, classificação e exploração de coleções. Assim, a modelagem de tópicos fornece uma solução para gerenciar, organizar e anotar grandes coleções de textos, motivo pelo qual optou-se pela sua utilização neste trabalho.

5 TRABALHOS RELACIONADOS

Foram selecionados quatro trabalhos relacionados a fim de guiar o estudo e o propósito deste trabalho, dois deles apresentam soluções para HRIR e outros dois que abordam o uso de modelagem de tópicos em sistemas de recomendação, que também enfrentam o problema de *cold start*.

5.1 HIGH RECALL INFORMATION RETRIEVAL

O trabalho de Yu, Kraft e Menzies (2018) apresenta uma abordagem chamada FASTREAD que utiliza Active Learning (AL) e máquinas de vetores de suporte (Support Vector Machines - SVM) para tratar o problema de HRIR.

AL É um método de aprendizado de máquina onde o algoritmo pode consultar interativamente o usuário ou alguma outra fonte de informações (RUBENS; KAPLAN; SUGIYAMA, 2011) e SVM é um classificador que encontra um hiperplano, ou linha de separação, entre dados de duas classes (GUNN et al., 1998).

O FASTREAD é baseado em outros três métodos do estado-da-arte: Wallace et al. (2010), Miwa et al. (2014) e Cormack e Grossman (2014). Os domínios escolhidos para o HRIR foram a descoberta de documentos legais e medicina baseada em evidências.

Os autores constatam que os três métodos analisados são construídos a partir de outras técnicas que abordam quatro questões básicas: (1) quando iniciar o treinamento, (2) qual documento consultar em seguida, (3) quando interromper o treinamento e (4) como equilibrar os dados de treinamento. Cada uma dessas quatro questões possui duas ou mais formas de tratar e cada método que serviu de base para a pesquisa propôs uma das abordagens a seguir:

a) Quando iniciar o treinamento;
   - Paciente (P): o algoritmo mantém a amostragem aleatória até que um número suficiente de documentos relevantes seja recuperado, considerando 5 como um número suficiente;
   - Apressado (H): considera um único documento relevante suficiente;

b) Qual documento consultar em seguida;
   - Amostragem de incerteza (U): os exemplos não-rotulados mais próximos do plano de decisão SVM são amostrados para consulta;
   - Amostragem de certeza (C): os exemplos não-rotulados mais distantes do plano de decisão SVM e localizados no lado relevante são amostrados para consulta;

c) Quando interromper o treinamento;
   - Parar o treinamento (S): o algoritmo interrompe o treinamento quando o classificador estabilizar. Segundo o autor o classificador é considerado estável depois que mais de 30 estudos relevantes foram recuperados como amostras de treinamento;

- Continuar treinando (T): o algoritmo nunca interrompe o treinamento, se a estratégia de consulta for U, o algoritmo alterna para C após estabilizar, mas o treinamento nunca para;

d) Como equilibrar os dados de treinamento;

- Sem balanceamento de dados (N): o algoritmo não equilibra os dados de treinamento;
- Subamostragem agressiva (A): o algoritmo utiliza subamostragem agressiva depois que o classificador fica estável;
- Ponderação (W): o algoritmo utiliza a ponderação para o balanceamento de dados, antes e depois que o classificador fica estável;
- E mistura de ponderação e subamostragem agressiva (M): a ponderação é aplicada antes que o classificador esteja estável, enquanto a subamostragem agressiva é aplicada depois disso.

Utilizando todas essas possibilidades em diferentes combinações foram avaliadas 32 diferentes métodos de aprendizado, incluindo Wallace et al. (2010) (PUSA), Miwa et al. (2014) (PCTW) e Cormack e Grossman (2014) (HCTN) que foram os métodos do estado-da-arte pesquisados pelos autores. A combinação que mostrou melhores resultados nos experimentos foi a HUTM pois reduziu o esforço necessário para encontrar documentos relevantes ao máximo e atingiu uma revocação de 95%. Essa abordagem foi chamada de FASTREAD.

Já o trabalho de Yu e Menzies (2019) é uma continuação do trabalho apresentado anteriormente e busca implementar melhorias na primeira versão do FASTREAD. Nesta segunda pesquisa, os autores buscam responder três perguntas que não tinham sido capazes de responder no primeiro trabalho que são (1) como começar ou como controlar a seleção inicial de documentos, (2) quando parar ou como saber quando parar a revisão com segurança e (3) como corrigir os erros de classificação feitos por humanos.

A primeira pergunta implica em como controlar a seleção inicial de documentos. Segundo os autores, as fases seguintes de aprendizado incremental podem variar bastante dependendo da seleção feita nesta etapa. Isso é importante, pois, uma seleção inicial ruim pode aumentar bastante o número de documentos que devem ser rotulados. A solução encontrada é que o primeiro artigo relevante pode ser identificado anteriormente, aplicando um pouco de conhecimento de domínio para orientar a amostragem inicial. Para isso, o algoritmo de ranqueamento BM25 foi utilizado pois mostrou melhores resultados no estudo realizado, reduzindo o esforço necessário de 10% a 20%.

Funciona da seguinte forma:

a) O BM25 começa consultando um conjunto de palavras-chave e classifica os exemplos não rotulados com base em suas pontuações;

b) O BM25 solicita ao revisor que revise 10 exemplos em ordem decrescente de suas pontuações; E

c) Finalmente, se o número de exemplos relevantes encontrados for maior ou igual a 1, inicia a fase de AL, caso contrário, o BM25 tenta um conjunto de palavras-chave diferente.

O BM25 é um algoritmo de recuperação de saco de palavras (*bag-of-words*) que classifica um conjunto de documentos com base nos termos de consulta que aparecem em cada documento, independentemente de sua proximidade dentro do documento (ROBERTSON; ZARAGOZA et al., 2009).

A segunda pergunta implica em como saber quando a revisão pode ser interrompida com segurança. Embora provavelmente exista sempre mais um artigo relevante a ser encontrado, é útil saber quando a maioria dos artigos foi encontrada (por exemplo, 95%). Sem esse conhecimento o algoritmo pode parar muito cedo, deixando muitos documentos relevantes para trás ou parar tarde demais, causando leituras desnecessárias mesmo depois que todos os documentos relevantes forem encontrados. Segundo os autores, a taxa na qual as SVM incrementais encontram documentos segue uma relação matemática simples. Logo, durante o processo de revisão, a revocação dos documentos relevantes pode ser obtida com precisão através do uso de um regressor logístico semi-supervisionado. Assim, é possível determinar quando uma determinada revocação foi atingida e interromper a revisão.

A última pergunta assume que os revisores humanos não são perfeitos e, algumas vezes, rotularão os documentos relevantes como não relevantes e vice-versa. Deste modo, as ferramentas (para revisões de literatura) devem ser capazes de reconhecer e reparar rotulagem incorreta. Como solução, os autores apontam que esses erros podem ser eficientemente identificados e corrigidos, revisando periodicamente alguns documentos rotulados, cujos rótulos não estiverem mais de acordo com o AL.

A extensão proposta para o FASTREAD capaz de solucionar essas três importantes questões os autores deram o nome de FAST^2.

Na resposta à primeira pergunta foi descrita a solução encontrada para o problema de *cold start* do FASTREAD, com o BM25.

Em seguida serão apresentados trabalhos nos quais a modelagem de tópicos, a qual é objeto de estudo deste trabalho, foi utilizada para resolver esse tipo de problema em sistemas de recomendação.

5.2 ENFRENTANDO O PROBLEMA DE *COLD START*

Wang e Blei (2011) desenvolveram um algoritmo de aprendizado de máquina para recomendar artigos científicos aos usuários em uma comunidade científica *online*. O algoritmo usa dois tipos de dados – as bibliotecas dos outros usuários e o conteúdo dos artigos – para formar suas recomendações. Para cada usuário, o algoritmo pode encontrar artigos antigos que são importantes para outros usuários semelhantes e novos artigos cujo conteúdo reflete os interesses específicos do usuário.

A abordagem utilizada pelos autores combina ideias de CF utilizando Latent Factor Model (LFM) e análise de conteúdo com base na modelagem probabilística de tópicos. LFM é uma metodologia do estado-da-arte de CF baseada em modelos (WANG; BLEI, 2011).

Assim como no LFM, o algoritmo desenvolvido usa informações das bibliotecas de outros usuários para recomendar a um usuário específico artigos daqueles que gostaram de artigos semelhantes a ele. LFMs funcionam bem para recomendar artigos conhecidos, mas não conseguem fazer o mesmo para artigos que jamais foram vistos pelo modelo.

Para esse problema (também conhecido como problema de *cold start*) o algoritmo proposto usa modelagem de tópicos, que fornece uma representação dos artigos em termos de temas latentes descobertos na coleção de documentos. Quando usado no sistema de recomendação, esse componente pode recomendar artigos com conteúdo semelhante a outros artigos de que um usuário gosta. A representação de tópicos dos artigos permite que o algoritmo faça recomendações significativas sobre os artigos antes que alguém os avalie.

O método desenvolvido combina essas abordagens em um modelo probabilístico. Funciona da seguinte forma: um artigo que não foi visto por muitos será recomendado com base mais em seu conteúdo; um artigo amplamente visto será recomendado com base nos outros usuários. Segundos os autores esse método tem melhor desempenho do que os métodos de Matrix Factorization (MF) – onde dados são colocados em uma matriz com uma dimensão representando usuários e a outra dimensão representando itens de interesse (KOREN; BELL; VOLINSKY, 2009) – utilizados isoladamente, indicando que o conteúdo desse artigo pode melhorar os sistemas de recomendação. Além disso, o método consegue usar o conteúdo de novos artigos para fazer previsões sobre quem irá gostar deles.

O trabalho de Lin et al. (2013) descreve um método que procura por informações retiradas do Twitter para fornecer recomendações relevantes de aplicativos recentemente lançados. Nessa situação ocorre o problema de *cold start*, quando os aplicativos ainda não possuem classificações dos usuários nas quais um CF poderia obter dados suficientes para fazer recomendações.

O método utiliza as arrobas (@) do Twitter para acessar a conta de um aplicativo e extrair os IDs de seus seguidores. Com isso, cria pseudo-documentos que contêm os IDs dos usuários da rede social interessados em um aplicativo e, em seguida, aplica o LDA para gerar tópicos latentes. No momento do teste, um usuário de destino que busca recomendações é mapeado para esses tópicos. Ao usar o relacionamento transitivo de tópicos latentes para aplicativos, estima a probabilidade de o usuário gostar do aplicativo.

Lin et al. (2013) mostram que, ao incorporar informações do Twitter usando modelagem de tópicos, sua abordagem supera o problema de *cold start* da recomendação de aplicativos e supera significativamente outras técnicas de recomendação do estado-da-arte em até 33%.

O trabalho aqui proposto difere dos citados anteriormente pois propõe utilizar o LDA tal qual Wang e Blei (2011) e Lin et al. (2013) para o problema de *cold start* porém aplicado ao sistema de HRIR de Yu e Menzies (2019) como alternativa ao uso do BM25.

6 METODOLOGIA

A área de pesquisa foi sugerida pelo orientador e discutida junto ao orientando. Mediante a definição dessa etapa, em acordo entre orientador e orientando, iniciaram-se os estudos. Foram previamente selecionados pelo orientador os artigos sobre o LDA (BLEI, 2012) e FASTREAD (YU; KRAFT; MENZIES, 2018) e (YU; MENZIES, 2019), que serviram como bases do estudo proposto neste trabalho.

A partir da compreensão das ideias e da relação entre os artigos de base, iniciou-se a pesquisa bibliográfica buscando por trabalhos que propunham objetivos semelhantes aos propostos neste trabalho. A plataforma de busca utilizada foi a *Google Scholar*[1] e as *strings* de busca utilizadas foram "cold start", "machine learning", "topic modeling"e "high recall". Como resultado, foram retornados aproximadamente 70 artigos. Destes, foram selecionados os trabalhos que apresentam similaridade ao estudo proposto neste trabalho, de acordo com a relevância (quantidade de citações, local e data de publicação).

A seguir, foram filtrados e elencados os artigos de Wang e Blei (2011) e Lin et al. (2013) os quais juntaram-se a Yu, Kraft e Menzies (2018) e Yu e Menzies (2019) – previamente selecionados pelo orientador – como os principais trabalhos relacionados a proposta deste estudo. Ao realizar a leitura de Wang e Blei (2011) e Lin et al. (2013) observou-se a utilização da modelagem de tópicos para solucionar o problema de *cold start* em sistemas de recomendação. Já nos artigos de Yu, Kraft e Menzies (2018) e Yu e Menzies (2019) observou-se a utilização de uma implementação do algoritmo BM25 para solucionar esse mesmo problema, desta vez, em um sistema de HRIR. Assim, considerou-se a ideia de utilizar um modelo de tópicos LDA para substituir o BM25 como solução ao problema de *cold start* do FASTREAD.

Definido o tema de pesquisa, foi efetuado o planejamento. O trabalho então foi segmentado em etapas menores que em conjunto contemplam os objetivos propostos neste trabalho. As etapas foram divididas da seguinte forma:

a) **Etapa I**: pesquisa bibliográfica. Identificação e análise dos trabalhos relacionados;

b) **Etapa II**: preparação das coleções de documentos. Coleta e pré-processamento dos textos;

c) **Etapa III**: aplicação do modelo LDA. Codificação do modelo e treinamento utilizando os dados das coleções;

d) **Etapa IV**: identificação do número de tópicos que representa melhor as coleções;

e) **Etapa V**: identificação dos *TOP-N* tópicos utilizados para classificar as sementes;

f) **Etapa VI**: identificação da quantidade de documentos utilizados como sementes;

g) **Etapa VII**: aplicação do modelo LDA ao FASTREAD. Inserção do modelo desenvolvido no código-fonte substituindo a solução original para o *cold start*;

[1] https://scholar.google.com/

h) **Etapa VIII**: definição dos casos de teste. Quais coleções são utilizadas e quais as *queries* de busca são aplicadas às coleções a fim de criar testes comparativos entre a solução original e o modelo LDA;

i) **Etapa IX**: avaliação dos resultados. Comparativo entre a revocação obtida com o BM25 e o modelo LDA desenvolvido utilizando os testes definidos na etapa anterior;

A **Etapa I** compreende a identificação e análise dos artigos correlatos. Essa etapa visa a análise do problema de *cold start* e o estudo das tecnologias envolvidas, tais como LDA e HRIR. Também consiste na identificação das diferentes fases do sistema FASTREAD e seus algoritmos a fim de substituir a solução utilizada por Yu, Kraft e Menzies (2018) para solucionar o problema de *cold start*.

A **Etapa II** consiste na reunião e preparação das coleções de documentos. Nesta etapa ocorre o pré-processamento dos textos através da remoção de *stopwords* (palavras frequentes como "the", "is", etc. que não possuem semântica específica) e técnicas de *stemming* (as palavras são reduzidas a uma raiz ao remover a inflexão através da eliminação de caracteres desnecessários, geralmente um sufixo) e lematização (outra abordagem para remover a inflexão, determinando a parte da fala e utilizando um banco de dados detalhado do idioma).

Optou-se pela utilização da biblioteca *gensim*[2] tanto para efetuar o pré-processamento dos textos quanto para implementação do modelo LDA. A escolha dessa biblioteca se deu principalmente ao fato de ser uma biblioteca da linguagem Python[3], a mesma utilizada na implementação do FASTREAD, isso viabiliza a codificação dos algoritmos no mesmo código-fonte.

A **Etapa III** é destinada a implementação do modelo LDA, utilizando a biblioteca *gensim*. Essa etapa inclui o treinamento do modelo utilizando as coleções reunidas e processadas na etapa anterior.

Na **Etapa IV** ocorre a identificação do número de tópicos que melhor representa cada uma das coleções de documentos. Essa etapa é importante pois um número de tópicos pequeno ou grande demais pode fazer com que a estrutura de tópicos se afaste da estrutura temática da coleção. Para isso, é importante conhecer as coleções de documentos que serão utilizadas e aproximar o número de tópicos do número de temas da coleção.

A **Etapa V** visa a escolha dos *TOP-N* tópicos utilizados para classificar as sementes. Aqui são testados diferentes valores de N (1, 2 e 3, por exemplo). O valor de N é definido conforme a relevância dos documentos obtidos pelos tópicos.

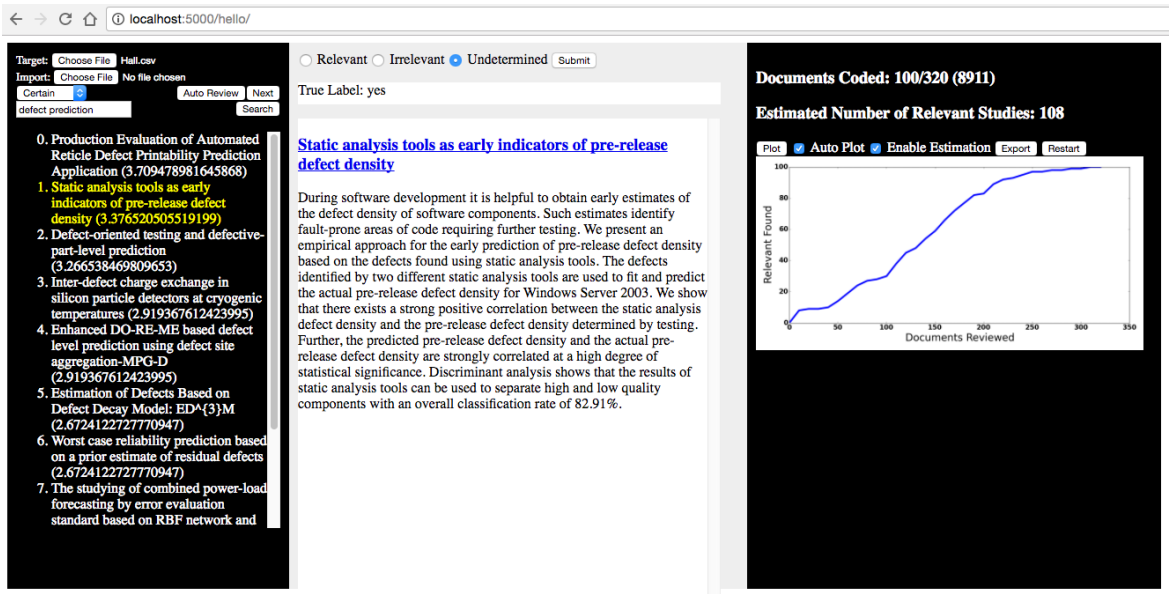
A quantidade de documentos utilizados como semente é definida na **Etapa VI**. A quantidade de documentos definida nessa etapa controla quantos deles são apresentados ao revisor na tela do FASTREAD para que este confirme ou não a sua relevância.

A Figura 6 ilustra a tela do FASTREAD apresentada ao revisor. Do lado esquerdo encontra-se a *query* de busca (*"defect prediction"*) e um conjunto de documentos. Ao clicar

[2] https://radimrehurek.com/gensim/
[3] https://www.python.org/

sobre um desses documentos é exibido ao revisor o título e o resumo (*abstract*) do documento (ao centro), onde ele pode classificá-lo quanto à sua relevância sobre a busca. Do lado direito encontram-se dados sobre a quantidade de documentos revisados e documentos relevantes encontrados na coleção. Na primeira utilização dessa tela e até que o primeiro documento relevante seja classificado, os documentos exibidos são aqueles classificados pelo algoritmo que atua na fase de *cold start*.

Figura 6 – Tela apresentada ao revisor no sistema FASTREAD



Fonte: Yu, Kraft e Menzies (2018)

A **Etapa VII** trata da substituição no código-fonte do FASTREAD do algoritmo de ranqueamento BM25 pelo modelo LDA desenvolvido. Nessa etapa, o repositório onde encontra-se o código-fonte é clonado e são feitas as modificações.

A **Etapa VIII** aborda os casos de teste. É necessário definir quais coleções são utilizadas e quais as *queries* de busca são aplicadas às coleções. Com testes bem definidos é possível comparar o FASTREAD utilizando BM25 e LDA como solução ao problema de *cold start*.

E finalmente, na **Etapa IX** são avaliados os resultados, utilizando os casos de teste criados na **Etapa VIII** e tendo como métrica a revocação obtida com o FASTREAD.

7 CRONOGRAMA

O cronograma de realização das etapas deste trabalho está organizado segundo a tabela a seguir. Cada célula representa uma quinzena do mês da sua referida coluna:

Atividades	Ago		Set		Out		Nov		Dez		Jan		Fev		Mar		Abr		Mai		Jun		Jul	
Execução da etapa I	X	X	X	X	X	X	X	X	X	X														
Execução da etapa II											X	X												
Execução da etapa III													X	X	X	X	X							
Execução da etapa IV													X	X	X	X	X							
Execução da etapa V													X	X	X	X	X							
Execução da etapa VI													X	X	X	X	X							
Execução da etapa VII																		X	X	X	X			
Execução da etapa VIII																		X	X	X	X			
Execução da etapa IX																					X	X		
Redação da monografia													X	X	X	X	X	X	X	X	X	X	X	X

Tabela 3 – Cronograma de atividades

REFERÊNCIAS

BAKHSHANDEGAN MOGHADDAM, Farshad; ELAHI, Mehdi. Cold Start Solutions For Recommendation Systems, mai. 2019. DOI: 10.13140/RG.2.2.27407.02725.

BELÉM, Fabiano M et al. Exploiting syntactic and neighbourhood attributes to address cold start in tag recommendation. **Information Processing & Management**, Elsevier, v. 56, n. 3, p. 771–790, 2019.

BLEI, David M. Probabilistic topic models. **Communications of the ACM**, Association for Computing Machinery (ACM), v. 55, n. 4, p. 77, abr. 2012. DOI: 10.1145/2133806.2133826. Disponível em: <https://doi.org/10.1145/2133806.2133826>.

BOBADILLA, JesúS et al. A collaborative filtering approach to mitigate the new user cold start problem. **Knowledge-Based Systems**, Elsevier, v. 26, p. 225–238, 2012.

CORMACK, Gordon V; GROSSMAN, Maura R. Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In: ACM. PROCEEDINGS of the 37th international ACM SIGIR conference on Research & development in information retrieval. [S.l.: s.n.], 2014. p. 153–162.

DAVIS, Jesse; GOADRICH, Mark. The relationship between Precision-Recall and ROC curves. In: PROCEEDINGS of the 23rd international conference on Machine learning - ICML '06. [S.l.]: ACM Press, 2006. DOI: 10.1145/1143844.1143874. Disponível em: <https://doi.org/10.1145/1143844.1143874>.

DUARTE, Denio; STÅHL, Niclas. Machine Learning: A Concise Overview. In: STUDIES in Big Data. [S.l.]: Springer International Publishing, set. 2018. p. 27–58. DOI: 10.1007/978-3-319-97556-6_3. Disponível em: <https://doi.org/10.1007/978-3-319-97556-6_3>.

GUNN, Steve R et al. Support vector machines for classification and regression. **ISIS technical report**, University of Southampton, v. 14, n. 1, p. 5–16, 1998.

KOREN, Yehuda; BELL, Robert; VOLINSKY, Chris. Matrix factorization techniques for recommender systems. **Computer**, IEEE, n. 8, p. 30–37, 2009.

LIN, Jovian et al. Addressing cold-start in app recommendation. In: PROCEEDINGS of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '13. [S.l.]: ACM Press, 2013. DOI: 10.1145/2484028.2484035. Disponível em: <https://doi.org/10.1145/2484028.2484035>.

MIWA, Makoto et al. Reducing systematic review workload through certainty-based screening. **Journal of biomedical informatics**, Elsevier, v. 51, p. 242–253, 2014.

ROBERTSON, Stephen; ZARAGOZA, Hugo et al. The probabilistic relevance framework: BM25 and beyond. **Foundations and Trends® in Information Retrieval**, Now Publishers, Inc., v. 3, n. 4, p. 333–389, 2009.

ROEGIEST, Adam. **On Design and Evaluation of High-Recall Retrieval Systems for Electronic Discovery**. [S.l.]: UWSpace, 2017. Disponível em: <http://hdl.handle.net/10012/11464>.

RUBENS, Neil; KAPLAN, Dain; SUGIYAMA, Masashi. Active Learning in Recommender Systems. In: KANTOR, P.B. et al. (Ed.). **Recommender Systems Handbook**. [S.l.]: Springer, 2011. p. 735–767. DOI: 10.1007/978-0-387-85820-3_23.

STEYVERS, Mark; GRIFFITHS, Tom. Probabilistic topic models. **Handbook of latent semantic analysis**, v. 427, n. 7, p. 424–440, 2007.

WALLACE, Byron C et al. Semi-automated screening of biomedical citations for systematic reviews. **BMC bioinformatics**, BioMed Central, v. 11, n. 1, p. 55, 2010.

WANG, Chong; BLEI, David M. Collaborative topic modeling for recommending scientific articles. In: PROCEEDINGS of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11. [S.l.]: ACM Press, 2011. DOI: 10.1145/2020408.2020480. Disponível em: <https://doi.org/10.1145/2020408.2020480>.

YU, Zhe; KRAFT, Nicholas A.; MENZIES, Tim. Finding better active learners for faster literature reviews. **Empirical Software Engineering**, Springer Nature, v. 23, n. 6, p. 3161–3186, mar. 2018. DOI: 10.1007/s10664-017-9587-0. Disponível em: <https://doi.org/10.1007/s10664-017-9587-0>.

YU, Zhe; MENZIES, Tim. FAST2: An intelligent assistant for finding relevant papers. **Expert Systems with Applications**, Elsevier BV, v. 120, p. 57–71, abr. 2019. DOI: 10.1016/j.eswa.2018.11.021. Disponível em: <https://doi.org/10.1016/j.eswa.2018.11.021>.