CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

MODELO NEURAL FRACAMENTE SUPERVISIONADO DE BUSCA DE ESPECIALISTAS EM REPOSITÓRIO DE DADOS CIENTÍFICOS

SÉRGIO JOSÉ DE SOUSA

Orientador: Prof. Dr. Thiago Magela Rodrigues Dias
Centro Federal de Educação Tecnológica de Minas Gerais

Coorientador: Prof. Dr. Adilson Luiz Pinto
Universidade Federal de Santa Catarina

BELO HORIZONTE
AGOSTO DE 2021

SÉRGIO JOSÉ DE SOUSA

MODELO NEURAL FRACAMENTE SUPERVISIONADO DE BUSCA DE ESPECIALISTAS EM REPOSITÓRIO DE DADOS CIENTÍFICOS

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Mestre em Modelagem Matemática e Computacional.

Área de concentração: Modelagem Matemática e Computacional

Linha de pesquisa: Sistemas Inteligentes

Orientador: Prof. Dr. Thiago Magela Rodrigues Dias
Centro Federal de Educação Tecnológica de Minas Gerais

Coorientador: Prof. Dr. Adilson Luiz Pinto
Universidade Federal de Santa Catarina

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL
BELO HORIZONTE
AGOSTO DE 2021

Sousa, Sérgio José de
S725m Modelo neural fracamente supervisionado de busca de especialistas em repositório de dados científicos / Sérgio José de Sousa. – 2021.
55 f.

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.
Orientador: Thiago Magela Rodrigues Dias.
Coorientador: Adilson Luiz Pinto.
Dissertação (mestrado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Classificação – Modelos matemáticos – Teses. 2. Especialistas – Teses. 3. Aprendizagem – Processamento de dados – Teses. 4. Recuperação de dados (computação) – Teses. 5. Curriculum vitae – Teses. I. Dias, Thiago Magela Rodrigues. II. Pinto, Adilson Luiz.. III. Centro Federal de Educação Tecnológica de Minas Gerais. IV. Título.

CDD 519.5

Elaboração da ficha catalográfica pela bibliotecária Jane Marangon Duarte, CRB 6º 1592 / Cefet/MG

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO CURSO DE MESTRADO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

"MODELO NEURAL FRACAMENTE SUPERVISIONADO DE BUSCA DE ESPECIALISTAS EM REPOSITÓRIO DE DADOS CIENTÍFICOS"

Dissertação de Mestrado apresentada por **Sérgio José de Sousa**, em 31 de agosto de 2021, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

Prof. Dr. Thiago Magela Rodrigues Dias (Orientador)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Adilson Luiz Pinto (Coorientador)
Universidade Federal de Santa Catarina

Prof. Dr. Washington Luís Ribeiro de Carvalho Segundo
Instituto Brasileiro de Informação em Ciência e Tecnologia

Prof. Dr. Thiago de Souza Rodrigues
Centro Federal de Educação Tecnológica de Minas Gerais

Visto e permitida a impressão,

Profª. Drª. Elizabeth Fialho Wanner
Presidenta do Colegiado do Programa de Pós-Graduação em Modelagem Matemática e Computacional

Dedico este trabalho em memória de meu pai, à minha esposa Lauranne, e aos familiares e amigos que estiveram ao meu lado em toda minha jornada.

“The best way to predict the future is to invent it.” (Alan Kay, 1940)

Resumo

Com o crescente volume de dados produzidos nos dias atuais, percebe-se cada vez mais usuários utilizando de diversos tipos de sistemas, como, por exemplo, sistemas de armazenamento de dados profissionais e acadêmicos. Dada a grande quantidade de dados armazenados, é notável a dificuldade de se encontrar candidatos com perfis apropriados a uma determinada atividade. Neste contexto, para tentar solucionar esse problema surge a recuperação ou busca de especialistas, um ramo da recuperação de informações, que consiste em, dada uma consulta, documentos são recuperados e são relacionados como unidades indiretas de informações das especialidades dos candidatos, com isso, alguma técnica é usada para agregar esses documentos gerando um escore. Possuindo um número menor de pesquisas relacionadas, a busca de especialistas na área acadêmica com modelos neurais se mostra um desafio ainda maior devido à complexidade desses modelos e à necessidade de grandes volumes de dados com julgamentos de relevância ou rótulos para seu treinamento. Diante disso, este trabalho propõe uma técnica de expansão e geração de dados fracamente supervisionados onde os julgamentos de relevância são criados com técnicas heurísticas, tornando possível utilizar modelos que exigem grandes volumes de dados. Além disso, é proposto uma técnica utilizando *autoencoder* profundo para selecionar documentos negativos ou julgamentos de irrelevância e por fim um modelo de ranqueamento baseado em redes recorrentes denominado *Dual Embedding LSTM* que foi capaz de superar todos os *baselines* comparados.

Palavras-chave: Modelo de ranqueamento. Fraca supervisão. Aprendizado profundo. Recuperação de Especialistas. Busca de Especialistas. Plataforma Lattes.

Abstract

With the growing volume of data produced today, it is clear that more and more users are using different types of systems, such as, for example, professional and academic data storage systems. Given the large amount of stored data, the difficulty of finding candidates with appropriate profiles for a particular activity is noteworthy. In this context, to try to solve this problem comes the expertise retrieval, a branch of information retrieval, which consists of, given a query, documents are recovered and used as indirect units of information for the candidates and some aggregation techniques are used in these documents to generate a score to the candidate. There are several models and techniques to work with this problem, some have been tested extensively but the search for specialists in the academic field with neural models has a smaller amount of research, this fact is due to the complexity of these models and the need for large volumes of data with judgments of relevance or labeled for your training. Therefore, this work proposes a technique of expansion and generation of weak supervised data where the relevance judgments are created with heuristic techniques, making it possible to use models that require large volumes of data. In addition, is proposed a technique of deep auto-encoder to select negative documents and finally a ranking model based on recurrent neural networks that was able to overcome all the baselines compared.

Keywords: Ranking model. Weak supervision. Deep learning. Expertise retrieval. Lattes Platform.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

BM25	Best Match 25 - Melhor Correspondência 25
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
GPU	Graphics Processing Unit
LExR	Lattes Expertise Retrieval
LSTM	Long Short-Term Memory
LM	Language Model - Modelo de Linguagem
MLE	Maximum Likelihood Estimation
MSE	Mean Squared Error
PCA	Principal Component Analysis
RI	Recumeração de Informação
TF-IDF	Term Frequency - Inverse Document Frequency
TREC	Text Retrieval Conference

Sumário

1 Introdução

O desafio de encontrar as informações que se desejam cresce juntamente com o volume dos dados. Logo, ferramentas de apoio como sistemas de recomendação e recuperação de informações são imprescindíveis ao realizar buscas, seja busca de sites, recomendações em sites de compras, busca de especialistas, entre outras possibilidades.

A busca de especialistas existe desde antes da invenção do computador e denota a necessidade de encontrar alguém com algum conhecimento específico. No ramo da psicologia (CHI; GLASER; FARR, 2014) é dito que o desempenho superior dos especialistas é adquirido através da experiência, nas repetições e ao estruturar das atividades duradouras. Para Lin et al. (2017) especialistas são pessoas com conhecimento ou que dominam habilidades detalhadas em áreas específicas. Essa tarefa é desafiadora pois é necessário avaliar o que a pessoa já fez, trabalhou e/ou produziu a fim de encontrar quais são suas especialidades e quem se destaca entre várias opções.

Na computação, motivada pelos recentes avanços na recuperação de informação e nas necessidades organizacionais a *Text Retrieval Conference (TREC)* entre os anos de 2005 e 2008 incluiu em suas linhas de desafios a busca de especialistas na trilha *Enterprise* do evento (CRASWELL; VRIES; SOBOROFF, 2005). Desde então diversos trabalhos foram propostos e várias subdivisões foram destacadas como é o caso de especialistas acadêmicos. Em geral, dada uma consulta em linguagem natural, o sistema de buscas deve propor uma ordem para os candidatos sendo seus documentos utilizados para representá-los.

O subdomínio busca de especialistas no meio acadêmico é uma das áreas que vem recebendo cada vez mais atenção apesar de possuir menos pesquisas relacionadas (BALOG et al., 2012), que pode estar relacionada à complexidade do problema ou pela dificuldade de se obter dados relevantes ao tema. É possível encontrar algumas bases como em Balog et al. (2007), Berendsen et al. (2013), Tang et al. (2008) em que pode-se observar conjuntos de dados de especialistas com foco em somente um tópico, não sendo generalista, centralizados em apenas uma organização ou em apenas uma área. Nesse contexto é possível destacar algumas plataformas de compartilhamento e armazenamento de projetos e trabalhos acadêmicos que se destacam como por exemplo Academia[1], DBLP[2], Google Scholar[3], Plataforma Lattes[4], Microsoft Academic Search[5], ResearchGate[6].

[1] Academia: https://www.academia.edu/
[2] DBLP: https://dblp.uni-trier.de/
[3] Google Scholar: https://scholar.google.com/
[4] Plataforma Lattes: http://lattes.cnpq.br/
[5] Microsoft Academic Search: https://academic.microsoft.com/
[6] ResearchGate: https://www.researchgate.net/

Especificamente, a Plataforma Lattes desenvolvida e mantida pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) surge como uma importante fonte de conhecimento científico. Em Dias e Moita (2015) os autores demonstraram o quão relevante os dados contidos na plataforma são para a compreensão dos pesquisadores e da ciência brasileira, incluindo além de informações pessoais, profissionais e acadêmicas, produção científica e tecnológica. Portanto, a Plataforma Lattes é uma fonte de informações de alta qualidade dos indivíduos (LANE, 2010) sendo uma boa fonte de dados para alimentar modelos e técnicas no geral como por exemplo, para identificação de especialistas.

1.1 Motivação

Para qualquer sistema de Recuperação de Informação (RI) a relevância dos itens retornados dado uma consulta é de extrema importância. Essas medidas podem variar de acordo com a robustez, sensibilidade e eficiência que se espera que o sistema demonstre (MITRA; CRASWELL et al., 2018). Esses atributos nos ajudam a comparar e selecionar modelos tradicionais e neurais que recentemente ganharam destaques.

A primeira rede neural artificial proposta em Rosenblatt (1957) em um formato bem simples denominado perceptron que hoje é basicamente uma unidade de neurônio de um rede atual. Um perceptron é basicamente um classificador binário linear que tenta encontrar uma melhor separação dos dados. Um dos maiores problemas desse modelo é a incapacidade de separar dados não lineares como uma distribuição *xor*, para ser resolvido devemos aumentar a profundidade e largura da rede, adicionando neurônios em uma ou mais camadas. Isso gera outros problema, o sobretreino ou *overfitting*, quando o modelo decora os dados de treino não conseguindo generalizar a validação, outro grande problema é a necessidade de uma arquitetura de hardware avançada que era cara na época. Com a redução de preços de hardwares como *GPUs*, o avanço em técnicas que reduzem o *overfitting* como *dropout* e a maturidade e especialidades das arquiteturas proporcionou um grande avanço e um destaque em redes neurais profundas.

Redes neurais trouxeram grandes melhorias nas áreas de visão computacional, processamento de linguagem natural e reconhecimento de fala (LECUN; BENGIO; HINTON, 2015), diferente de técnicas tradicionais essas redes se beneficiam de grandes quantidades de dados, possuindo uma capacidade de aprender contextos e relações difíceis de identificar. Mais recentemente tivemos algumas tentativas de propor e adaptar essas técnicas para a recuperação de informação (RI).

Nesses modelos neurais aplicados a RI destaca-se um grande problema, a necessidade de grandes quantidades de dados rotulados. Esses rótulos consistem em julgamentos de relevância, ou seja, um conjunto de triplas contendo uma consulta, um documento e qual a pontuação ou escore dessa relação. A rotulação de dados é dispendiosa e pode levar

muito tempo, o que inviabiliza a aplicação desses modelos em muitos casos.

Motivado pelos novos modelos neurais e pela necessidade de dados com julgamentos de relevância, o presente trabalho apresenta uma alternativa de modelos com uma fraca supervisão em que os rótulos são gerados por heurísticas. Logo, nesta Dissertação é proposto uma técnica para gerar pseudo-julgamentos de relevância para os documentos, tendo em vista que para uma dada consulta é necessário também amostras de documentos relevantes e amostras de documentos não relevantes. Para gerarmos amostras de documentos positivos para as consultas propomos uma estratégia para geração de julgamentos de relevâncias através de técnicas clássicas de recuperação de informação baseadas em modelo de linguagem com suavização bayesiana usando distribuição de Dirichlet (ZHAI; LAFFERTY, 2017); para amostras de documentos negativos é proposto um *deep autoencoder* (HINTON; SALAKHUTDINOV, 2006), calculando os candidatos mais distantes de cada consulta e extraindo seus documentos.

Para realizar uma reclassificação dos documentos propomos uma arquitetura neural dual com camadas recorrentes para recalcular a ordem e os escores dos documentos que por fim compõe o escore de cada candidato.

1.2 Justificativa

Os principais argumentos desta Dissertação é a ascensão dos modelos neurais e a necessidade de grandes quantidades de dados com rótulos. Tendo em vista o ramo da busca de especialistas, propomos uma técnica para, através de uma heurística aplicando modelos tradicionais para gerar julgamentos de relevância, um *deep autoencoder* aplicado aos candidatos a especialistas para identificar documentos mais distantes para cada consulta e por fim um modelo *Dual Embedding LSTM* para re-ordenar os resultados.

Para tanto, apresentamos as perguntas de pesquisa que estão relacionadas e serão respondidas ao longo desta Dissertação:

Q1. A fraca supervisão pode obter resultados melhores que a técnica tradicional usada para gerar os julgamentos de relevâncias aplicando em problemas de busca de especialistas?

Q2. A geração de documentos negativos para as consultas através de um *deep autoencoder* pode superar o padrão de selecionar documentos aleatórios?

Q3. O modelo *Dual Embedding LSTM* pode superar os modelos da literatura?

1.3 Contribuições

Dentro do contexto de busca de especialistas é apresentado um arcabouço para geração de pseudo-julgamentos de relevância e o experimento é realizado utilizando a base de dados *LExR* apresentada por Mangaravite e Santos (2016) e descrita mais adiante. Assim sendo, dentre as principais contribuições deste Dissertação se destacam:

- Apresentação de uma técnica para gerar os rótulos em documentos com relação positiva às consultas.
- Apresentação de um modelo para encontrar os documentos mais distantes das consultas, ou seja, os documentos com relação negativa dada uma consulta.
- Proposta de um modelo *Dual Embedding LSTM* com o objetivo de realizar o re-ranqueamento de documentos retornados por uma consulta.

1.4 Organização do trabalho

O restante desta Dissertação está organizado da seguinte forma:

- No Capítulo 2, é apresentada a fundamentação que serve de base e assuntos que são permeados por esta Dissertação. Este capítulo inclui os conceitos de recuperação de informação, tanto modelos tradicionais quanto modelos neurais.
- No Capítulo 3, são apresentados, discutidos e comparados alguns trabalhos relacionados aos assuntos abordados nesta Dissertação que também servem de inspiração e como análises comparativas.
- No Capítulo 4, é descrito a base de dados utilizada para validar as técnicas e modelos propostos, bem como as transformações realizadas nos dados.
- No Capítulo 5, é apresentado todo o arcabouço proposto, incluindo as técnicas e modelos para selecionar documentos positivos e negativos dada uma consulta e por fim, um modelo *Dual Embedding LSTM* para realizar o re-ranqueamento dos documentos.
- No Capítulo 6, é apresentada a estrutura do experimento, os resultados obtidos e análises dos resultados.
- Por fim, no Capítulo 7, são apresentadas as considerações finais, conclusões, publicações e próximas etapas para possíveis trabalhos futuros.

2 Fundamentação Teórica

Criar modelos de RI é uma tarefa desafiadora cujo propósito é criar uma função que, atribui um escore dada uma consulta e um documento, essa função é aplicada por todos ou parte dos documentos que por fim são ordenados de maneira decrescente. Esse processo pode ser dividido em dois componentes: (1) uma técnica para representação dos dados de entrada do modelo, e (2) a definição da função que calcula o grau de similaridade gerando um escore entre consulta e documento.

Segundo Balog et al. (2012) quando lidamos com problema de busca de especialistas acrescentamos mais dois componentes na modelagem, (1) o modelo de ranque que agrega todos documentos de um candidato e (2) um esquema que associa documento e candidato, em muitos casos apresentados na *TREC* a relação candidato e documento é incerta, são e-mails trocados, postagens em sites, etc. o que gera essa necessidade de calcular a incerteza.

Então, um modelo de recuperação de informação simples pode ser definido pela quádrupla:

$$D, Q, I, S(q_i, d_j)$$

, onde D é o corpus dos documentos, Q representa o conjunto das consultas, I é a técnica usada para transformar os dados, tanto documentos quanto consultas e por fim S a função que calcula a similaridade gerando um escore entre documento e consulta. Já para a busca de especialistas acrescentamos mais três elementos:

$$D, Q, C, I, S(q_i, d_j), A(c_k, d_j), F_c$$

, onde C representa o conjunto de candidatos, $A(c_k, d_j)$ a função de associação documento-candidato e por fim F_c a função de ranqueamento de candidatos.

Uma importante unidade que faz parte dos documentos e consultas são os termos de indexação que podem ser definidos por uma pessoa ou extraídos dos corpus de documentos e consultas. Esses termos constroem uma visão lógica dos documentos, descrevendo para o modelo e para a máquina o seu significado. Essa representação pode variar dependendo da necessidade dos modelos.

A seguir são descritos alguns dos principais Modelos Clássicos de RI (Seção 2.1) em seguida alguns Modelos de Ranqueamento de Especialistas (Seção 2.2).

2.1 Modelos Clássicos de RI

Existem vários modelos para tentar resolver problemas de RI, nas próximas seções serão apresentados alguns dos mais importantes e relevantes para esta Dissertação, cada um possuindo uma maneira de representar os documentos e consultas.

Uma das primeiras e mais básicas formas, foi adotar uma representação booleana, consiste em verificar a existência ou ausência dos termos da consulta nos documentos. Sua vantagem é justamente sua simplicidade e suas desvantagens incluem a ausência de um ranqueamento que pode levar a um retorno com poucos ou muitos documentos.

Com representações mais elaboradas a Subseção 2.1.1 trata de modelos com uma representação vetorial, em que os termos fazem parte de um grande vetor que compõe os documentos e a Subseção 2.1.2 apresenta modelos que utilizam representações probabilísticas entre termo, documentos e consultas.

2.1.1 Modelos Algébricos e Vetoriais

Como demonstrado em Salton, Wong e Yang (1975) a representação booleana é muito restritiva e não permite um casamento parcial. Modelos vetoriais utilizam de pesos não binários para indexação dos termos de documentos e consultas permitindo cálculos de distância e similaridade entre documentos/documentos e documentos/consultas. Dessa maneira, basta ordenar os documentos de acordo com o grau de similaridade entre consulta e documento, incluindo casamentos parciais aumentando a chance de se ter um ranqueamento melhor.

Outro aspecto importante é assumir a independência de termos, dessa maneira é possível agrupar e gerar um grande vetor de termos para cada documento e consulta. Em Jones (1972) foi realizado testes estatísticos para avaliar a especificidade dos termos em três coleções e verificado que a frequência é necessária para um bom desempenho num geral para os modelos e os resultados indicam melhorias consideráveis no desempenho aplicando essa técnica simples.

Assim os documentos e consultas podem ser definidos como nas representações abaixo:

$$\overrightarrow{d_j} = (w_1,j, w_2,j, ...w_T,j),$$
$$\overrightarrow{q} = (w_1,q, w_2,q, ...w_T,q),$$

onde $\overrightarrow{d_i}$ e $\overrightarrow{q}$ são a representação vetorial de um documento e da consulta respectivamente, $w_i,_j$ é o peso associado ao termo i no documento d_j assim como $w_i,_q$ é o peso associado ao termo i na consulta, e T é o número total de termos da coleção.

Com essa representação podemos aplicar técnicas de ponderação ou de cálculo de similaridade como pode ser visto a seguir.

2.1.1.1 Ponderação *TF-IDF*

TF-IDF é uma ponderação composta pela frequência do termo (*Term Frequency, TF*) e frequência inversa de documento (*Inverse Document Frequency, IDF*). Será descrito a seguir cada um desses componentes separadamente.

Luhn (1957) propôs uma hipótese em que o peso de um termo de um vocabulário é proporcional a sua frequência de maneira que um termo é mais relevante para um documento se sua frequência for maior, demonstrando uma possível importância ou um tópico-chave.

Uma das versões mais comuns de cálculos de *TF* é log na base 2 da frequência do termo, como pode ser visto na Equação (1), dessa maneira o peso da frequência cresce de maneira mais suave.

$$w_{i,j} = tf_{i,j} = \begin{cases} 1 + log_2 f_{i,j} & \text{se} f_{i,j} > 0 \\ 0 & \text{caso contrário} \end{cases} \tag{1}$$

Aplicando apenas a frequência para geração dos pesos, gera-se um problema no qual termos muito comuns podem ganhar muita relevância influenciando no processo de ranqueamento. A ponderação pela frequência inversa do documento (*IDF*) que foi desenvolvida em Jones (1972), veio com um ferramental para tentar mitigar esse problema, dando mais valor a termos com maior especificidade, ou seja quanto mais específico um termo maior será seu valor, um exemplo simples, a especificidade do termo “carro” e “motocicleta” tende a ser muito maior que a palavra “veículo”.

Uma maneira de realizar o cálculo de *IDF* é aplicando o log_2 ao valor da quantidade de documentos da coleção (N) sobre a quantidade de documentos que possuem o termo i (n_i) como na Equação (2) que também possui uma característica suavizada.

$$IDF_i = log_2 \left(\frac{N}{n_i} \right) \tag{2}$$

Assim ao juntarmos estas duas equações temos *TF-IDF* (Equação (3)) que gerou muitos pontos positivos incluindo a característica priorizar e valorizar termos frequentes cuja especificidade também seja alta. Vale ressaltar que termos frequentes com especificidade baixa tendem a ser *stopwords* ou palavras vazias, como “e”, “ou”, “a”, “o” e termos com especificidade alta e frequência baixa costumam ser palavras em estrangeiros ou erros de grafia.

$$TF\_IDF_i = (1 + log_2 f_{i,j}) \times log_2 \left(\frac{N}{n_i} \right) \tag{3}$$

2.1.1.2 Modelo Vetorial com Cálculo de Similaridade

Para calcular a correlação entre o documento d_j e a consulta do usuário q através dos vetores encontrados com T dimensões devemos aplicar uma função de similaridade, como, por exemplo, através do cosseno do ângulo cuja equação pode ser vista na Equação (4).

$$cosine(d_j,q) = \frac{\overrightarrow{d_j} \bullet \overrightarrow{q}}{\|\overrightarrow{d_j}\| \times \|\overrightarrow{q}\|} = \frac{\sum_{i=1}^{T} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{T} (w_{i,j})^2} \times \sqrt{\sum_{i=1}^{T} (w_{i,q})^2}} \tag{4}$$

O elemento $\overrightarrow{d_j} \bullet \overrightarrow{q}$ é o produto interno entre os vetores de documentos e consultas, $\|\overrightarrow{d_j}\| \times \|\overrightarrow{q}\|$ é a multiplicação das normas. Aqui podemos observar que $\|\overrightarrow{d_j}\|$ normaliza pelo tamanho do documento. Se $w_{i,j}$ e $w_{i,q}$ são $>= 0$, $cosine(d_j,q)$ varia entre 0 e 1, caso contrário esse valor pode variar de -1 a 1.

Os pesos dos vetores podem ser definidos por exemplo aplicando *TF-IDF*, assim teremos valores tratados de acordo com sua frequência, especificidade e normalizado pelo cálculo de similaridade do cosseno, assim quanto menor o valor for o valor do cosseno mais próximo o documento é da consulta.

2.1.1.3 Redes Neurais

Inspirado em neurônios dos cérebros biológicos, as redes neurais artificiais possuem várias unidades e várias camadas, cada neurônio pode ser considerado uma unidade básica de processamento que, dado um valor de entrada com um peso associado, somado um viés, por fim passado por uma função de ativação e assim propagado para os próximas camadas. Calcula-se o erro com alguma função como erro quadrático e esse erro é retropropagado (*backpropagation*) distribuindo esse erro entre camadas e neurônios com uma função de otimização.

Uma das primeiras propostas de modelo neural para RI pode ser vista em Wilkinson e Hingston (1991), que possui três camadas que incluem a camada de entrada com os nós de consulta, uma camada secundária com termos do vocabulário e por fim a camada com nós dos documentos. Com essa estrutura os termos das consultas podem ativar neurônios do vocabulário que por sua vez ativam os documentos, por fim os documentos ativados são ordenados criando o resultado do ranqueamento.

Esse modelo possui diversos problemas, mas se destacam a estrutura estática sendo necessário retreinar o modelo caso um novo documento seja inserido, mudando completamente a estrutura anterior, o mesmo vale para adicionar um novo termo ao vocabulário.

Com uma arquitetura mais robusta e muito parecida com o visto na Subsubseção 2.1.1.2, Salakhutdinov e Hinton (2007) propõem um *deep autoencoder* para aprender uma representação reduzida dos documentos e consultas.

No geral, *autoencoders* possuem uma arquitetura em formato de ampulheta, a primeira camada possuindo vários neurônios e a cada camada mais profunda tem um número menor até chegar na metade do modelo, essa primeira metade é conhecida como *encoder*, a segunda metade, chamada de *decoder*, aumenta de maneira simétrica até as dimensões originais. Basicamente o dado de entrada passa por uma redução, depois uma ampliação até o tamanho original.

Autoencoders possuem diversas aplicações, dentre elas destacam *autoencoder* para redução de ruídos, no qual o dado de entrada possui ruído e o dado de saída é comparado ao dado sem ruído, assim o modelo é treinado para corrigir ruídos; e a redução de dimensionalidade, treinando o *autoencoder* para reconstruir o dado de entrada e no fim separando e utilizando o *encoder* para gerar um dado compacto em um espaço semântico latente, como o autor do artigo Salakhutdinov e Hinton (2007) definiu, um "*hash* semântico".

Usando o vetor gerado aplicando a ponderação *TF-IDF*, reduzindo para esse espaço de *hash* semântico de todo conjunto de documentos e aplicando o mesmo *encoder* na consulta, assim comparando a similaridade do cosseno entre documentos e consulta, for fim encontrando os mais próximos.

2.1.2 Modelos Probabilísticos

O princípio proposto por Robertson (1977) para ranqueamento probabilístico é base para diversos modelos como Modelos de Linguagem e *BM25*, consistem em, dada uma consulta q de um usuário e um documento d_j do corpus, tenta-se estipular qual a probabilidade desse documento ser relevante ou útil. O modelo supõe que existe um conjunto de documentos R que contém todos os documentos que o usuário julga ser a resposta dada uma consulta. Documentos dentro desse conjunto R são tidos como relevantes e os que não fazem parte desse conjunto como não relevantes. Esse modelo também supõe que a probabilidade de relevância depende exclusivamente das representações dos documentos e da consulta.

Após uma série de operações algébricas, operando quando não se tem informação de relevância dos documentos, chega-se na formulação de similaridade que pode ser vista na Equação (5) em que N é o total de todos documentos, n_i são todos documentos que contêm o termo k_i e o componente $k_i \in q \wedge k_i \in d_j$ seleciona todos os termos que estão presentes na consulta e que também estão presentes no documento d_j

$$sim(d_j,q) \propto \sum_{k_i \in q \wedge k_i \in d_j} log_2 \left(\frac{N - n_i + 0.5}{n_i + 0.5} \right) \tag{5}$$

Nota-se a grande semelhança dessa formulação para com IDF (Equação (3)), não possuindo a frequência dos termos nem um fator de normalização. Esses problemas são

resolvidos aplicando extensões como por exemplo BM25 (Subsubseção 2.1.2.1) ou Modelo de Linguagem (Subsubseção 2.1.2.2).

2.1.2.1 BM25

BM25 ou *Best Model 25* é um de uma série de modelos propostos e avaliados empiricamente que estendem do modelo probabilístico clássico, integrando a formulação do *BM11* e *BM15* cuja equação pode ser conferida na Equação (6) extraída de atualização proposta por Robertson, Zaragoza e Taylor (2004).

$$sim_{BH25}(d_j,q) \propto \sum_{k_i \in q \wedge k_i \in d_j} \frac{(K+1)f_{i,j}}{K\left[(1-b)+b\frac{len(d_j)}{avgLen}\right]+f_{i,j}} \times log_2\left(\frac{N-n_i+0.5}{n_i+0.5}\right) \quad (6)$$

A função itera por termos da consulta que também está no documento d_j. O segundo componente dessa equação idêntica a Equação (5) estando relacionado ao *IDF*. O primeiro componente engloba tanto a frequência do termo quanto a normalização, $f_{i,j}$ é a frequência do termo i no documento j, $len(d_j)$ é o tamanho do documento sobre $avgLen$ que é a média dos tamanhos dos documentos. K e b são constantes configuráveis dependendo da coleção utilizada, b é o fator de importância dado a normalização pelo tamanho do documento, um padrão adotado $K = 1$ e $b = 0.75$. Então basta ordenar os documentos pela maior similaridade.

Uma das maiores vantagens do BM25 é a possibilidade do ranque ser computado de maneira automática sem nenhuma informação de relevância ser informada.

2.1.2.2 Modelo de Linguagem

Modelos de linguagem ou *LM*, são técnicas aplicadas para tentar prever a chance de aparecer um determinado termo em uma sequência de termos. Essa premissa é usada em muitas aplicações como, tradução automática, reconhecimento de voz, processamento de linguagem natural e na recuperação de informações em que a ideia é construir um modelo de linguagem para cada documento e verificar qual a chance desse documento gerar a consulta, ou seja, diferente do que foi visto até agora no qual calculamos a probabilidade de ou à semelhança do documento com a consulta aqui usamos o texto do documento para tentar prever a probabilidade de observar a consulta. Com isso basta ordenar as probabilidades dos documentos que tem-se o ranqueamento da consulta.

Por padrão, *LM* lida com a dependência entre os termos, calculando com um processo markoviano, uma estrutura com n-grama na qual os termos são dependentes aos n anteriores. Computacionalmente esse processo é muito caro e fraco em estimativa devido

aos dados esparsos. Se assumirmos a independência entre os termos o *LM* utilizará de unigramas tornando um rápido modelo e com bons resultados (LIU; CROFT, 2005).

$$P(q|\theta_{d_j}) \propto \sum_{i \in q} tf_{i,q} log_2 P(i|\theta_{d_j}) \tag{7}$$

Na Equação (7) temos uma das técnicas de *LM* em que tenta-se localizar a probabilidade de encontrar a consulta dado um modelo de probabilidade de um documento d_j ($P(q|\theta_{d_j})$), $tf_{i,q}$ é a frequência do termo i na consulta q. $P(i|\theta_{d_j})$ é a probabilidade de se encontrar o termo i dado o modelo de probabilidade do documento d_j, uma forma de se calcular é através da Máxima Verossimilhança ou *Maximum Likelihood Estimation (MLE)* como pode ser visto na Equação (8), em que $tf_{i,j}$ é a frequência do termo i no documento j sobre o tamanho do documento d_j ($len(d_j)$).

$$P_{MLE}(i|\theta_{d_j}) = \frac{tf_{i,j}}{len(d_j)} \tag{8}$$

Ao aplicar esse conjunto de equações alguns problemas podem ser observados, como os escores muito otimistas para termos observados e o escore zerado para os não observados. A aplicação de técnicas de suavização ajudam a reduzir esses problemas. A suavização bayesiana com distribuição de Dirichlet é uma alternativa que obteve os melhores resultados como função de estimativa (ZHAI; LAFFERTY, 2017) e pode ser verificada na Equação (9).

$$P^s(i|\theta_{d_j}) = \frac{tf_{i,j} + \lambda \frac{F_i}{\sum_i F_i}}{\sum_i tf_{i,j} + \lambda} \tag{9}$$

A função tenta estimar a probabilidade suavizada de encontrar o termo i dado o modelo de probabilidades do documento j ($P^s(i|\theta_{d_j})$) em que $tf_{i,j}$ é a frequência do termo no documento, F_i a frequência do termo i em todos os documentos da coleção que é dividido pela somatória das frequências de todos os termos de todos documentos. Soma-se esses valores e divide-se pelo somatório das frequências de todos os termos do documento j e esses termos são ponderados pela constante λ que varia de 0 a 1, quanto mais próximo de 1 mais suavizado se torna o modelo de linguagem.

2.2 Ranqueamento de Especialistas

A pesquisa em buscas de especialista é extensa e abrange vários tópicos importantes, em resumo temos um modelo de ranqueamento dos candidatos a especialista (Subseção 2.2.1) e um modelo que associe um documento a um candidato (Subseção 2.2.2) pois essa relação pode ser incerta ou pode ser necessário ponderar e aplicar alguma normalização.

Logo, o corpus de documentos de problemas de ranqueamento de especialistas é composto por documentos vinculados a possíveis autores e coautores. A seguir, esses tópicos serão abordados com mais detalhes.

2.2.1 Modelos de Ranqueamento de Candidatos

2.2.1.1 Modelos de Votação

O problema de busca de especialista pode ser tratado como um problema de votação, no qual dada uma consulta os documentos são ranqueados e uma ponderação é realizada ao agregar os ranques dos documentos para cada candidato a especialista. Uma técnica bem simples é somar a quantidade de documentos agrupados por autores, cada documento contando como um voto, ou somar os escores de cada documento também agrupando por autores.

Macdonald e Ounis (2006) realizaram um experimento aplicando ao todo onze técnicas de ponderação, medindo as eficácias dessas abordagens. Os autores destacaram evidências de um resultado positivo melhorando significantemente o desempenho da recuperação. Uma das técnicas que obteve um melhor resultado foi *expCombMNZ* (Equação (10))

$$score_c = ||D_{q,c}|| * \sum_{j \in D_{q,c}} exp(score_{j,c}) \tag{10}$$

O escore do candidato c ($score_c$) é dado pela quantidade de documentos retornados pela consulta q para o candidato c ($||D_{q,c}||$) vezes o somatório do exponencial dos escores do documentos retornados para o candidato c.

2.2.1.2 Modelos Probabilísticos

Modelos probabilísticos tentam encontrar a probabilidade de um candidato c ser um especialista dada uma consulta q. Esses modelos são divididos em dois grupos com características distintas, são eles, modelos com abordagens generativas e discriminativas.

Em modelos probabilísticos discriminativos o modelo aprenderá a gerar o ranque através de um gabarito para as consultas enquanto nos modelos generativos a probabilidade é gerada pela distribuição conjunta dos eventos sem a necessidade de um gabarito.

- **Modelos Probabilísticos Generativos:** estimando através do teorema de Bayes, Balog e Rijke (2006) propuseram a aplicação da Equação (11) em que os candidatos com maiores valores na probabilidade dada uma consulta são os mais bem posicionados

no ranque.

$$P(c|q) = \frac{P(q|c)P(c)}{P(q)} \propto P(q|c) \tag{11}$$

Na Equação (11) $P(q)$ está relacionado apenas a consulta, logo constante para todos documentos podendo ser retirado. Já $P(c)$ possui uma distribuição uniforme, logo o cálculo de $P(c|q)$ se resume em resolver $P(q|c)$. Diversos modelos para solucionar podem ser encontrados em Balog e Rijke (2006), Macdonald e Ounis (2006).

- **Modelos Probabilísticos Discriminativos:** São treinados modelos de otimização com base em consultas previamente rotuladas. Essa abordagem engloba modelos *learning to rank*, Opitz e Maclin (1999) propuseram um *ensemble*, ou seja um conjunto de modelos ponderados que tentam aprender a ranquear os candidatos.

2.2.2 Modelos de Associação Candidato-Documento

O ranque de especialistas difere entre as abordagens em ambientes empresariais e acadêmicos ao analisarmos as associações entre candidato e documento. Em geral, problemas acadêmicos possuem metadados que informam os autores associados, facilitando a formulação de soluções, já ambientes empresariais muitas vezes essas informações devem ser estimadas as associações entre documentos e candidatos.

Juntamente com a função de associação que pondera a relação candidato-documento existem diversas funções de normalização dos pesos, logo essa relação é dada pela função:

$$f(c, d) = \omega(\rho(c, d)) \tag{12}$$

Onde ρ é a função de associação e ω a função de normalização. Proposto por Balog e Rijke (2006) a função de de associação booleana é dada por

$$\rho(c,d) = \begin{cases} 1, & \text{se há relação entre } c \text{ e } d \\ 0, & \text{caso não} \end{cases} \tag{13}$$

Uma função simples para normalização é dividir o valor da associação pela quantidade de candidatos que estão relacionados àquele documento, essa abordagem é chamada de normalização centrada em documento (BALOG; AZZOPARDI; RIJKE, 2006).

2.3 Avaliação dos Modelos

Existem diversas métricas de avaliação de modelos de RI, uma das mais utilizadas é a precisão nos top-p documentos. Considerando uma relevância maior que zero como igualmente relevantes e calculando a proporção para toda coleção com gabarito.

Outra métrica importante é o Ganho Cumulativo (*CG - Cumulative Gain*). Basicamente soma-se a relevância dos documentos do resultado da pesquisa (Equação (14)) em que p é o tamanho da avaliação e rel_i o valor de relevância.

$$CG_p = \sum_{i=1}^{p} rel_i \tag{14}$$

Extendendo e aprimorando o CG, o Ganho Cumulativo Descontado (*DCG - Discounted Cumulative Gain*) adiciona um elemento que verifica a ordem dos documentos e desconta proporcionalmente à posição (Equação (15)).

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{15}$$

Com a finalidade de tornar uma métrica geral e em um intervalo entre 0 e 1, no $nDCG$ (*Normalized Discounted Cumulative Gain*) quanto maior esse valor mais próximo da resposta ideias ($nDCG = 1$). A Equação (16) apresenta essa extensão onde $IDCG_p$ é DCG_p ideal gerado pelo gabarito da consulta (JÄRVELIN; KEKÄLÄINEN, 2002).

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{16}$$

Esse embasamento teórico é importante ao se avançar nesta Dissertação pois serve de referência para modelos avançados propostos adiante.

3 Trabalhos Relacionados

Por este trabalho ter relações que diversas áreas de conhecimento e contribuições de uma podem ajudar o avanço em outras, esta seção apresentará trabalhos relacionados agrupados por Buscas de Especialistas (Seção 3.1) de maneira geral, Buscas com modelos neurais (Seção 3.2).

3.1 Busca de Especialistas

Em alguns tipos de busca de especialistas um fator que pode ser considerado em modelos de associação é o peso temporal que o documento por ter, como por exemplo em busca de especialistas em que os documentos são produções acadêmicas, essa ponderação pode dar um peso maior para os documentos publicados recentemente.

O problema de busca de especialistas envolve buscar em documentos que são unidades de informação sobre a especialidade do candidato, em um cenário de busca de blogs, os autores são considerados como candidatos e as postagens como seus documentos. Com isso Keikha, Gerani e Crestani (2011a) investiga a classificação de blogs de acordo com sua relevância recorrente para o tópico da consulta, como uma das suas principais características é a sua relação temporal este trabalho propõe uma estrutura probabilística para medir a estabilidade da relevância dos blogs ao longo do tempo. Keikha, Gerani e Crestani (2011a) conseguiram mostrar que os autores de blogs mais relevantes são mais estáveis ao longo do tempo e os experimentos mostraram uma melhoria estatística significante em relação aos métodos comparados. Complementar a esse trabalho Zahedi et al. (2017) propõe uma adição de expansão de consulta temporal utilizando o método TEMPER (KEIKHA; GERANI; CRESTANI, 2011b) superando o modelo anterior.

Um outro problema comum em busca de especialista é identificar a autoria dos documentos, em ambientes acadêmicos temos uma rede complexa de coautorias em que apesar de parecer um problema simples, no entanto, representa um problema maior e de difícil solução. Consequentemente é necessário analisar os metadados e criar mecanismos para identificar os autores dos documentos (SMALHEISER; TORVIK, 2009).

Uma possível estratégia de identificação de autoria é comparar nomes, que mesmo quando os dados comparados possuem nomes completos, existem diversos problemas como homônimos, escritas com grafia diferente ou mesmo mudanças de nomes. Digiampietri e Ferreira (2018) trabalharam com três fontes de dados acadêmicos em que o foco é unificar dados relacionados a Plataforma Lattes, dados institucionais dos docentes da Universidade de São Paulo (USP) e perfis do Google Acadêmico. Já em MOREIRA (2018) os autores

precisavam identificar a rede de coautoria entre os dados da Plataforma Lattes e para isso propuseram uma técnica utilizando de dicionários, no qual, caso haja algum problema nessa identificação é comparada referências de outros documentos para auxiliar na seleção os identificadores dos perfis apropriados.

Inspirados nessas abordagens de identificação de autoria ponderando créditos à essas relações de documento e coautores Li et al. (2021) formaliza uma técnica para atribuir pontuações de relacionamento coautor e documento baseado em similaridade de caminhos e uma série de estratégias de seleção de subconjuntos para identificar a experiência dos autores.

Mangaravite e Santos (2016) utilizando da base *LExR* que consiste em um conjunto de dados de artigos de doutores registrados na Plataforma Lattes até 2015, os autores propuseram uma técnica baseada em teoria da informação em que a associação documento-autor é dada por uma probabilidade, ou seja, um modelo não booleano comumente utilizado. Com isso, em vez de tentar inferir a autoria de um documento, é proposto que esse modelo de associação não booleano reflita na probabilidade de que o documento seja relevante sobre a experiência de um candidato. Também é proposto dois esquemas alternativos de normalização que mede o quão discriminativo uma associação particular documento-autor é em vista as demais associações envolvidas em cada documento do autor. A abordagem superou os *baselines* propostos.

Em Balog et al. (2012) é descrito uma longa revisão de literatura que destaca os avanços dos modelos e algoritmos para busca e ranqueamento de especialistas, resumindo e estabelecendo as relações dessas abordagens. Mais recentemente (HUSAIN et al., 2019) foi conduzida uma revisão que selecionou 96 artigos consistindo em 57 artigos em periódicos, 34 artigos em anais de congresso e três capítulos de livros, analisando o domínio de busca de especialistas, fontes de conhecimento, métodos e base de dados. Verificaram uma tendência crescente na quantidade de pesquisas de RI para buscas de especialistas da academia. Complementar a esses dois trabalhos, Gonçalves e Dorneles (2019) além de incluir artigos posteriores a 2017 à sua revisão propõe uma taxonomia que serve de guia para pesquisadores classificarem trabalhos existentes e compará-los.

Nessa taxonomia, para cada trabalho é identificado:

- **Fonte de Dados:** Incluindo informações sobre o formato, tipo de acesso à esses dados e a maneira de visualizar;
- **Extração dos Dados:** Complexidade de extração, processamento realizado nos dados e se é uma fonte de dados completa ou focada;
- **Representação do Especialista:** Método usado, como grafos, frequência de termos, entre outros. Suporte temporal e suporte semântico;
- **Aplicação:** Objetivo da tarefa executada.

Por fim, Gonçalves e Dorneles (2019) apresenta uma série de problemas em aberto para fomentar pesquisas e os classificando em:

- Associação de especialistas: lidar por exemplo da relação pessoa e documento, documentos com mais de um autor e o quão importante o documento pode ser para um especialista;
- Combinação de múltiplas evidências, incluindo análises por exemplo de modelos de aprendizado automático ou que, de alguma forma aproveitem feedbacks;
- Lidar com buscas de especialistas cujos documentos estão em diversas linguagens;
- Veracidade dos dados de especialistas;
- Interação com usuário;
- Explicações dos resultados;
- Descrições dos especialistas;
- Análises contextuais;
- Colaboração entre domínios;
- Representação única dos especialistas que seja possível ser trocada entre sistemas;

3.2 Busca com Modelos Neurais

No trabalho de Guo et al. (2016) é proposto um modelo baseado em redes neurais profundas chamado DRMM (*Deep Relevance Matching Model*) com o objetivo de realizar correspondências utilizando como dados de entrada histogramas gerados ao comparar a consulta com os documentos.

O trabalho segue detalhando a diferença entre a correspondência semântica e a relevância da correspondência, destacando que em tarefas de RI a relevância da correspondência é de extrema importância.O modelo proposto utiliza de *embeddings* já treinados como *word2vec*, então é calculada a distância do cosseno entre cada termo da consulta e cada termo do documento, formando um histograma com o tamanho variando 0.5 entre -1 e 1. Então é verificada 3 possibilidades de histogramas, um utilizando o valor direto, outro normalizado e por fim outro baseado no log da contagem.

Foram utilizadas as base de dados *Robust04*[1] e *ClueWeb-09-Cat-B*[2] ambos da TREC. O modelo proposto utilizando log no histograma obteve os melhores resultados comparado aos demais *baselines*.

Focado nos problemas de diferentes vocabulários ou documentos que podem conter palavras de consulta sem ser relevantes, Mitra et al. (2016) busca atingir esses pontos e para isso propõe a transformação dos termos em um *embedding word2vec* e um modelo *Dual* para realizar o ranqueamento dos documentos.

[1] TREC 2004 Robust Track: https://trec.nist.gov/data/t13_robust.html
[2] The ClueWeb09 Dataset: https://lemurproject.org/clueweb09

Essa nova abordagem proposta, em vez de considerar as repetições dos termos como evidência de proximidade, é considerada a relação entre os termos da consulta e todos os termos do documento. Por exemplo, dada a consulta “especialista”, em adição ao número de vezes que o termo aparece no documento, é verificado se os termos relacionados relacionados ocorrem no documento como “busca” e “recuperação”.

O corpus de treinamento consiste em um uma grande quantidade de consultas (618.644.170) com um vocabulário do tamanho de 2.748.230 palavras. Esses dados foram extraídos dos logs do período de 19 de Agosto de 2014 até 25 de Agosto de 2014[3]. Os dados são transformados utilizando um modelo de *Continuous Bag-of-Words* (CBOW) (MIKOLOV et al., 2013) que é otimizado maximizando o log condicional da probabilidade da palavra aparecer dado o contexto de palavras de uma dada janela em volta da palavra inicial.

Devido a capacidade desse modelo de capturar a ocorrência de palavras, o CBOW é um útil para modelar a proximidade de um documento. Os espaços de *embedding* aprendidos também são úteis pois contêm conhecimentos sobre propriedades distributivas das palavras. Com essa motivação é proposto uma função de classificação simples mas eficaz chamada de *Dual Embedding Space Model* (DESM).

$$DESM(Q, D) = \frac{1}{|Q|} \sum_{q_i \in Q} \frac{q_i^T \bar{D}}{\|q_i\| \left\|\bar{D}\right\|} \tag{17}$$

$$\bar{D} = \frac{1}{|D|} \sum_{d_j \in D} \frac{d_j}{\|d_j\|} \tag{18}$$

A Equação (17) mostra o modelo proposto por Mitra et al. (2016) em que Q representa a consulta, q_i são os termos de cada consulta e $\bar{D}$ é dada pela Equação (18) em que $\bar{D}$ é o centróide de todos os vetores normalizados para as palavras do documento.

Os dados para validar as equações também foram extraídos de logs do Bing e foi comparado com os *baselines BM25* e *LSA*. Os resultados foram promissores e DESM superou os *baselines* propostos ao comparar *nDCG*.

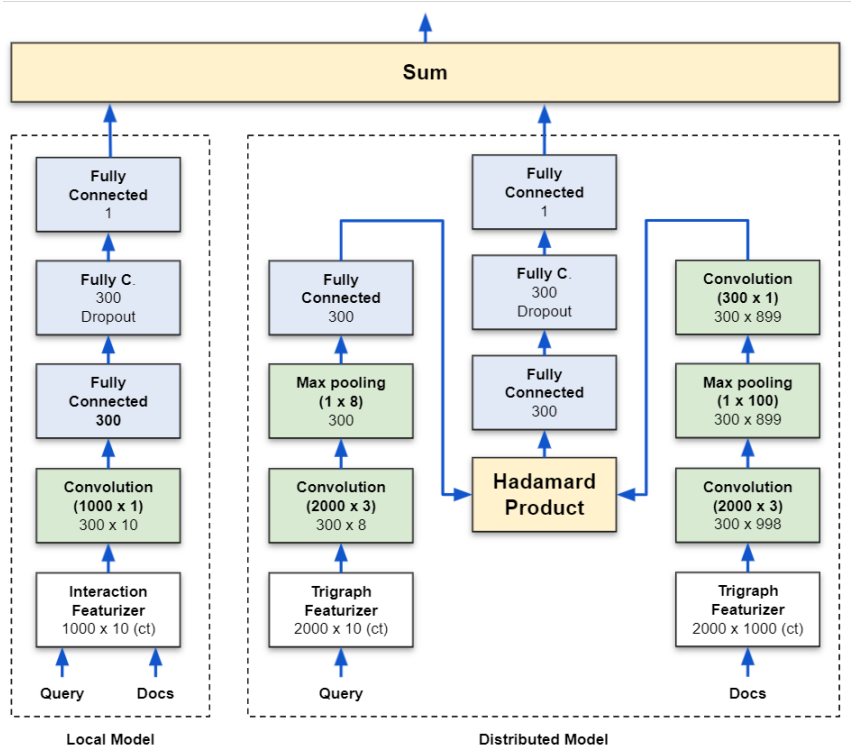
Em Mitra, Diaz e Craswell (2017) foca-se em conjunto de documentos que podem ser longos, como corpo de um texto, diferente de diversos outros trabalhos que comumente trabalham com textos curtos como títulos. Nos desafios com textos curtos há problemas relacionados com incompatibilidade de vocabulário, já com textos longos os documentos podem conter uma mistura de diversos tópicos, a correspondência em diferentes partes do documento não são uniformemente relevantes e a proximidade dos termos é importante.

É importante para RI diferentes pontos de correspondência, a correspondência exata, em que é relevante se o termo é raro ou novo, frequência e posições são bons

[3] Bing dataset: http://research.microsoft.com/projects/DESM.

indicadores de relevância e a proximidade dos termos. Já em correspondência inexatas deve-se focar em relacionamento com sinônimos, evidências sobre o que é o documento bem como proximidade e posição. Mitra, Diaz e Craswell (2017) lida com isso propondo um modelo do tipo *Duet* que utiliza de duas formas de representação dos dados, uma representação local cuja representação é feita através de um vetores *one-hot* com uma correspondência exata, e uma representação distribuída cujo textos são representados por um vetor denso (*embeddings*) que leva em contraposição dos termos e sua ordenação.

Figura 1 – Arquitetura Duet



Fonte: Elaboração de Mitra, Diaz e Craswell (2017)

A arquitetura *Duet* como pode ser visto na Figura 1 possui diversas entradas de valores sendo cada ramo principal com objetivo de atingir as propostas:

- **Local model:** Aprender uma rede neural para estimar a relevância de padrões exatos de correspondência;
- **Distributed model:** Aprender um *embedding* de texto baseado na correspondência da consulta com o documento no espaço do embedding.

Enquanto o modelo local opera com dados com uma matriz de interação lexical o modelo distribuído utiliza *n-graphs*, ou seja, cada termo é separado em conjuntos de caracteres formando um grande vetor com os 2.000 *n-graphs* mais comuns. Os dados passam pelas diversas camadas do modelo e se agrupam na última camada somando o escore atribuído pelo modelo local e modelo distribuído. Parte dos hiperparâmetros e da

arquitetura do modelo *Dual Embedding LSTM* utilizados nesta Dissertação foram inspirados em Mitra, Diaz e Craswell (2017).

Como é comum em modelos neurais, é necessário uma grande quantidade de dados para treinamento do modelo e foram usados dados de log de buscas do *Bing* rotulados por humanos nas relações positivas de documentos positivos e consulta e para relação de documentos negativos e consulta foram selecionados aleatoriamente entre o corpus.

Treinar modelos do zero pode ser extremamente caro computacionalmente, ainda mais quando se trata de modelos grandes e profundos como BERT (DEVLIN et al., 2018) e GPT-3 (BROWN et al., 2020) e somente grandes empresas com muito poder de processamento conseguem treiná-los a partir do zero. Uma maneira de se aproveitar desses poderosos modelos é usá-los pré-treinados, realizar uma transferência de conhecimento do modelo para novos dados, *fine-tuning* ou adaptar de alguma forma seu funcionamento para outra tarefa. O trabalho (NOGUEIRA; JIANG; LIN, 2020) propõe uma adaptação *Sequence-to-Sequence* (SUTSKEVER; VINYALS; LE, 2014) utilizando T5 (RAFFEL et al., 2019) que consiste em transformar esse modelo em um problema de classificação; *Sequence-to-Sequence* baseia-se em modelos neurais cujo dado de entrada formado por uma sequência de valores e seu valor a ser previsto como outra sequência, comumente usado em tarefas de tradução; dessa maneira a adaptação proposta tem como objetivo gerar rótulos de maneira a ser possível interpretar como probabilidade da classificação ser relevante ou não.

Os testes aplicados ao popular desafio MS MARCO (BAJAJ et al., 2016) e na trilha *Robust* (VOORHEES et al., 2004) de TREC 2004 e foi comparado com *BM25*, aplicando a técnica proposta utilizando *T5* e *BERT*. As propostas superam os modelos de classificação, especialmente em casos em que há uma escassez de dados de treinamento

Em Dehghani et al. (2017) os autores propuseram um modelo de rede neural profunda fracamente supervisionada. Os testes foram feitos em dois conjuntos de dados, sendo um de notícias e outro com dados gerais da internet, para as consultas foram utilizados logs de consultas do provedor de serviços AOL[4]. Os pseudo-julgamentos de relevância são gerados utilizando *BM25* (ROBERTSON; ZARAGOZA et al., 2009), sendo selecionados 1.000 documentos com maior escore para cada consulta e 1.000 outros documentos negativos amostrados aleatoriamente. No experimento são propostas três arquiteturas de rede, sendo uma *point-wise* e duas *pair-wise*, três representações dos dados de entrada, sendo uma representação densa com vários atributos calculados, uma representação esparsa com *bag-of-words* e por fim um vetor *embedding* aprendido durante o treinamento. A combinação da arquitetura *pair-wise* com a representação de *embedding* como entrada superaram a técnica de *baseline*.

[4] AOL: https://www.aol.com/

O tutorial proposto por Mitra, Craswell et al. (2018) apresenta conceitos básicos e intuições por trás dos modelos neurais aplicados em RI, revisando arquiteturas de redes neurais recentes, passando por modelos não supervisionados de representações de termos, modelos de *embedding* de termos para RI, aprendizado supervisionado para ranqueamento e redes neurais profundas, apontando seus pontos positivos e negativos, e por fim discutindo sobre possíveis direções futuras.

Todos estes trabalhos foram importantes e serviram de base para o conhecimento a ser aplicado na Dissertação.

4 Caracterização dos Dados

Neste capítulo é descrita a coleção de teste utilizada, apresentando as características dessa base, como foi gerada (Seção 4.1) e sobre o gabarito para uso em atividades de busca de especialistas (Seção 4.2).

4.1 Informações Gerais

Esta Dissertação utiliza dados extraídos da Plataforma Lattes, uma plataforma muito importante para todos pesquisadores brasileiros, pois seu uso é obrigatório para participação em diversos editais de fomento, em que é necessário manter seus dados o mais atualizado possível. Em se tratando de conteúdo, a Plataforma Lattes provê informações gerais sobre o pesquisador como nome, informações de contato; informações sobre publicações científicas como artigos em periódicos e artigos em publicações em anais de congresso; informações profissionais. Como é uma plataforma gratuita qualquer pessoa pode cadastrar seu currículo, incluindo estrangeiros.

Devido ao seu amplo uso e adoção pela maioria das agências de fomento, institutos de pesquisa brasileira e universidades, os dados curriculares da Plataforma Lattes se tornaram também amplamente usados como coleção de informações para compreender a ciência brasileira (SANTIAGO; DIAS; AFFONSO, 2020; GOMES; DIAS; MOITA, 2018; DIAS et al., 2018; DIAS et al., 2016; AFFONSO; DIAS; SANTIAGO, 2020).

Como esta Dissertação propõe um arcabouço em que se utiliza a fraca supervisão em seu treinamento é necessário uma base contendo um gabarito para validar as metodologias propostas. Para isso, foi utilizado o conjunto de dados apresentado em Mangaravite et al. (2016) que consiste em uma coleção focada em busca de especialistas extraídos da Plataforma Lattes.

A Tabela 1 sumariza as estatísticas básicas da base de dados que foi extraída da plataforma em 2014 e 2015. Optou-se pela seleção dos candidatos à especialistas limitando ao contexto de pessoas com doutorado concluído, pois esse grupo é responsável por mais de 72% das publicações de artigos cadastrados nos currículos da Plataforma Lattes. O número total de documentos era de 16.526.452 e entre os doutores totalizando em 11.942.014. Ou seja, 223.853 pessoas cadastradas na plataforma, totalizando em 6,54% dos currículos, produziram 72,26% de todo material científico.

Entre os currículos, vários deles não possuem registros de produção científica, se excluirmos esse grupo de pessoas restam um total de 206.697 doutores representando 21,19% do total de 975.470 pessoas que possuem pelo menos uma associação com um

Tabela 1 – Estatísticas gerais da coleção LExR

	Doutores	**Lattes (2015)**	**Percentual**
Documentos	11.942.014	16.526.452	72,26%
Candidatos	223.853	3.423.548	6,54%
Candidatos com pelo menos 1 Associação	206.697	975.470	21,19%
Associações (Candidato x Documento)	21.015.538	30.128.338	69,75%
Média de Associações por Documento	1,76	1,82	
Média de Associações por Candidato	101,67	30,89	
Média de Documentos por Candidatos	57,78	16,94	
Consultas (qrel)	235		
Julgamentos de relevância	1.635		

documento, com uma média de 1,76 associações por documento e cada doutor com uma média de 57,78 associações com documentos.

O conteúdo da coleção de testes LExR (MANGARAVITE et al., 2016) possui apenas metadados bibliográficos contendo uma lista de documentos que é composto de trabalhos publicados em anais de eventos (50,58%); artigos completos em periódicos (23,48%); apresentações de artigos em eventos (19,90%); e livros completos ou capítulos (6,04%). Contendo o título da publicação, palavras chaves, ano de publicação, lista de autores.

Apenas 52,5% das relações de co-autoria possuem o vínculo explícito do identificador dos currículos da plataforma, para solucionar o problema de desambiguação de autores foi utilizado uma técnica simples de remover caracteres especiais dos títulos, duplos espaços, padronizar em caixa baixa e juntamente com o ano de publicação tentar encontrar esse identificador para relacionar os autores ao documento, dessa maneira totalizando em mais de 21 milhões de associações.

4.2 Gabarito para busca de especialistas

Com acesso à informações de contato extraídos da Plataforma Lattes Mangaravite et al. (2016) entraram em contato com os candidatos a especialistas e enviou a eles um questionários para identificação de possíveis especialistas, esses rótulos utilizados para formulação do questionário usou a abordagem proposta por Ribeiro et al. (2015) no qual é descrito o problema de *expert profiling* que tenta sumarizar os conhecimentos dos candidatos.

Foram 57.852 rótulos avaliados pelo questionário e após uma filtragem em que mantiveram aqueles que foram selecionados por pelo menos três candidatos, totalizaram 4.105 consultas. Até esse momento os candidatos fizeram auto-avaliação de suas especialidades e 1.348 pessoas participaram da pesquisa.

Uma segunda etapa foi feita na qual desses 1.348, 514 participaram, o objetivo nessa etapa é um candidato a especialista avaliar os demais. Então uma última filtragem foi

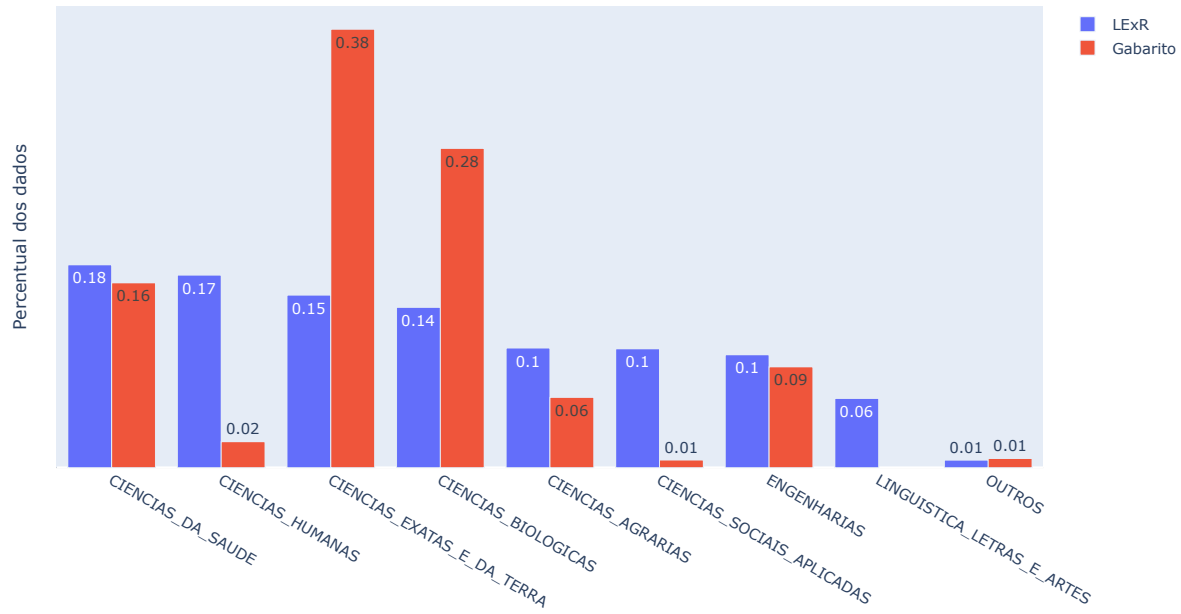
feita, permanecendo apenas consultas com pelo menos três candidatos a especialista e com pelo menos duas respostas nessa segunda etapa. No fim o gabarito é composto por 235 consultas e 1.635 relações de consulta/candidato a especialista, também conhecido como *qrel*.

A classificação proposta possui a seguinte descrição de acordo com o escore atribuído:

- **1 = Parcialmente relevante:** O pesquisador possui conhecimento na área;
- **2 = Relevante:** O pesquisador é um especialista na área;
- **3 = Fortemente relevante:** O pesquisador é uma das principais referências na área;

Outro dado importante é que todas as consultas composta no gabarito estão em inglês, apesar de 61% das publicações estarem em português (MANGARAVITE et al., 2016), 51% das publicações dos participantes do gabarito estão em inglês, seguido por 33% em português, 5,4% e, espanhol e 6,6% os demais idiomas. Logo optou-se por manter as consultas em inglês também.

Figura 2 – Percentual de dados por grande área



Fonte: Elaboração de Mangaravite et al. (2016)

Ao compararmos os especialistas contidos no gabarito podemos verificar o percentual de pessoas separadas por grande área (Figura 2). Podemos ver que Ciência Exatas e da Terra possui 38% dos especialistas do gabarito enquanto se compararmos em toda base de dados de doutores essa grande área representa 15%. Ciências Humanas e Ciências Sociais Aplicadas juntos representam 3%, uma quantidade pequena, evidenciando um

ponto negativo desse conjunto de dados que possui muito pouco registros para essas áreas serem avaliadas e a grande área de Linguística, Letras e Artes não conta com nenhum especialista no gabarito.

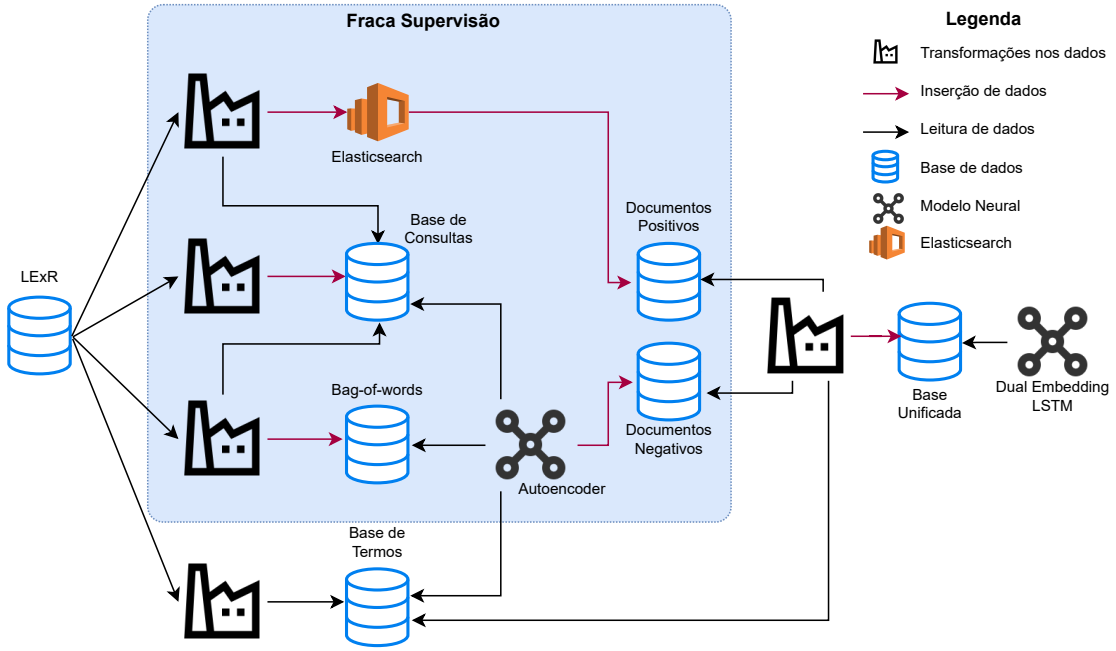
Apesar desse ponto negativo, LExR é uma base de dados muito importante e única para podermos avaliar possíveis modelos de busca de especialistas utilizando a Plataforma Lattes como fonte de dados.

5 Metodologia

Neste capítulo é apresentado o arcabouço proposto, inicialmente são apresentados os Diferenciais Propostos (Seção 5.1) seguindo pelas Suposições e Restrições (Seção 5.2) da metodologia, após são descrito as Transformações nos Dados (Seção 5.3), em seguida é descrito sobre a Fraca Supervisão (Seção 5.4) com a geração e seleção de documentos com relacionamento positivos e negativos com as consultas e por fim é apresentado o modelo *Dual Embedding LSTM* (Seção 5.5)

A Figura 3 sumariza o arcabouço proposto.

Figura 3 – Estrutura da metodologia proposta



Fonte: Elaboração do autor

5.1 Diferenciais Propostos

No Capítulo 3 foi descrito trabalhos próximos ao que está sendo apresentado aqui. Em Dehghani et al. (2017) é proposto que se treine modelo fracamente supervisionado para busca tradicional em que o objetivo é re-ranquear documentos, aqui propomos extrapolar essa aplicação para busca de especialistas, um problema mais complexo pois documentos são utilizados para representar de alguma maneira quantificável a especialidade dos candidatos

Para tanto foi usado o conjunto de teste proposto em Mangaravite et al. (2016), uma base de dados que até então foi pouco explorada e inédita em aplicações utilizando redes neurais, isso se dá pelo fato da base de teste contar uma quantidade muito pequena de julgamento de relevâncias *(*qrel) tornando um treinamento inviável se esse dado for usado diretamente para o treino.

Outra contribuição da metodologia proposta é a seleção não aleatória de documentos cujo relacionamento é negativo dada uma consulta, como visto em Mitra, Diaz e Craswell (2017) no qual foi proposto o uso de julgamentos de não relevância gerados por humanos para os documentos se mostrou mais efetivo que uma seleção aleatória.

Logo as contribuições desta Dissertação se destacam:

- Uma seleção de documentos com julgamentos de irrelevância utilizando técnica de aprendizado não supervisionado e similaridade do cosseno;
- O uso de fraca supervisão para busca de especialistas utilizado dados da Plataforma Lattes;
- Um modelo neural denominado Dual Embedding LSTM que combina os benefícios de uma representação de embedding e a ordem e sequência dos termos em camadas recorrentes.

5.2 Suposições e Restrições

Apesar das diversas vantagens de modelos neurais como se beneficiar de grandes volumes da dados e dados com volumetria crescente e a vantagem de generalização diversas restrições e suposições são necessárias para o bom funcionamento do arcabouço proposto:

- Como redes neurais profundas tendem a ter um alto custo computacional tanto para seu treinamento quanto para o seu uso em ambiente de produção a maioria dos modelos propostos incluindo este utilizam alguma técnica de busca tradicional e rápida para trazer uma sub-amostragem de documentos e com essa seleção é feito um novo ranqueamento desses documentos;
- Redes neurais quando utilizam de camadas recorrentes possui restrições em treinamento e uso de computação paralela devido a dependência temporal entre os termos que precisam ser inseridos um a um no modelo em ordem;
- Os dois modelos neurais do arcabouço utilizam termos indexados, ou seja, caso haja a presença de novas palavras será necessário um novo treinamento;
- O conjunto de teste é pequeno comparado ao conjunto de dados ocasionando em valores baixos nas métricas de avaliação do sistema de busca;

Diante disso estes se tornam alguns restritivos da pesquisa que podem ser explora-

dos em trabalhos futuros.

5.3 Transformações nos Dados

Como mencionado no Capítulo 4 o conjunto de dados utilizado nesta Dissertação consiste em um grande arquivo no formato json cujo, cada linha corresponde a um documento. Para agilizar a geração de dados dos modelos o primeiro passo foi agrupar documentos por autor como pode ser visto na Figura 4 em que foi criado um dicionário cuja chave principal é a identificação de cada candidato a especialista, o item contido na chave é uma lista de documentos com informações de título, ano de publicação, uma lista de palavras chaves, lista de autores e caso haja, informações do DOI. A estrutura se estende por todos os documentos de cada autor.

Figura 4 – Estrutura da base de dados agrupada por autor


```
{"3527197809276361": [
    {"title": "Using Taxonomies for Product Recommendation",
     "year": 2012,
     "keywords": ["Recommender systems", "Product recommendation", "Taxonomies"],
     "authors": ["2034607422210997", "3405503472010994", "3527197809276361",
     "6261175351521953"],
     "DOI": ""},
    {"title": "Learning to expand queries using entities",
     "year": 2014,
     "keywords": [],
     "authors": ["1162362624079364", "3405503472010994", "3527197809276361",
     "4737852130924504", "4935788335854516"],
     "DOI": "http://dx.doi.org/10.1002/asi.23084"},
    {"..."} /*Demais Documentos*/
],
 "4737852130924504" : [
    "..." /*Demais Documentos*/
]
"..." /*Demais Doutores*/
}
```


Fonte: Elaboração do autor

A partir de uma única fonte de dados diversas transformações e novas fontes de dados foram geradas para atender a necessidade em cada etapa do processo, a lista a seguir resume cada base de dados geradas bem como sua aplicação:

- Lista de Termos: Usada em diversas etapas do processo, incluindo indexação e frequências (Subseção 5.3.1);
- *Bag-of-words* dos candidatos: totaliza os termos de cada candidato, usado na etapa de treinamento do modelo de *Deep auto-encoder* (Seção 5.4);
- Base de candidatos transformados via Encoder: Gerados pelo encoder e usado para selecionar candidatos de acordo com a distância da similaridade do cosseno (Seção 5.4);
- Base de dados de consultas (*query*): Base usada tanto para seleção de documentos positivos quando documentos negativos na etapa de fraca supervisão (Seção 5.4);

- Lista de documentos positivos e negativos para cada consulta: Usado no treinamento do Dual Embedding LSTM (Seção 5.5);
- Documentos indexados: Usado no treinamento do Dual Embedding LSTM (Seção 5.5);

Logo, seguindo algumas transformações iniciais são apresentadas na Subseção 5.3.1 que descreve as transformações a nível de termos e a Subseção 5.3.2 descreve uma breve análise sobre o resultado dessas transformações.

5.3.1 Identificação dos Termos

Primeiro passo na seleção dos termos é estabelecer a regra para identificação dos mesmos. Assumimos a separação de termos usando a separação por espaço e pontuações sendo o hífen excluído dessa regra. Por exemplo a frase: *"Real-time video super-resolution on smartphones with deep learning, mobile ai 2021 challenge: Report"* seria gerado uma lista com os seguintes termos *"Real-time", "video", "super-resolution", "on", "smartphones", "with", "deep", "learning", ",", "mobile", "ai", "2021", "challenge", ":", "Report"*.

Após isso, é necessário normalizar os termos, padronizando todos para caixa baixa e por fim calcular algumas métricas para o corpus de documentos. Primeiramente é calculada a frequência geral de cada termo, somando-se a quantidade de vezes que o termo é usado como pode ser visto na Equação (19) em que a frequência do termo tn é dada pela somatória de vezes que o termo é encontrado entre todos os termos de todos documentos, sendo D o corpus de documentos agrupados por candidato a especialista.

$$f_{tn} = \sum_{d_i \in D} \sum_{tn \in d_i} 1 \tag{19}$$

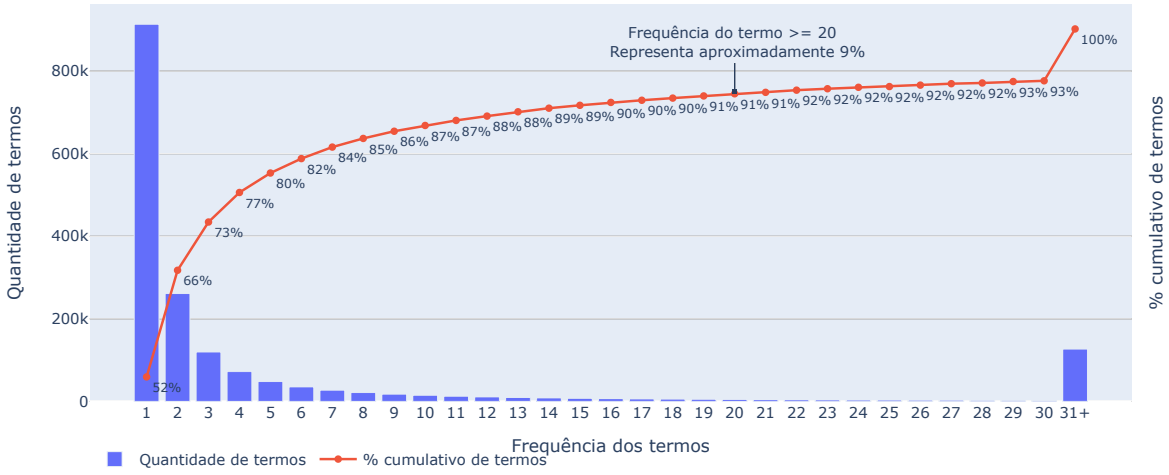
A próxima métrica é a quantidade de documentos que cada termo aparece. Esse valor é importante ao calcularmos a ponderação TF-IDF (Subsubseção 2.1.1.1), uma ponderação que valorizará termos menos comuns, ou seja, como nessa etapa consideramos um documento como um candidato, quanto menor a quantidade de candidatos que usam maior o peso daquele termo, o cálculo pode ser visto na Equação (20), em que df_{tn} é a frequência de documentos do termo tn.

$$df_{tn} = \sum_{d_i \in D} \begin{cases} 1 & \text{se } tn \in d_i \\ 0 & \text{caso contrário} \end{cases} \tag{20}$$

Podemos avaliar então os valores e frequência dos termos, a Figura 5 mostra um gráfico de Pareto em que totaliza a quantidade de termos dada uma frequência, no primeiro eixo y representado pelas barras e o percentual de dados que esse valor representa no total, no gráfico esse valor é dado no segundo eixo y com a linha. Podemos ver então que

cerca de 52% dos 1.767.752 totais de termos foram localizados apenas uma vez, podendo ser em sua grande maioria erros de digitação. Nessa representação é possível visualizar o fenômeno de cauda longa em que a maioria dos dados estão localizados no grupo com menor frequência.

Figura 5 – Pareto da das quantidades e frequência dos termos



Fonte: Elaboração do autor

Como uma das restrições citadas anteriormente, redes neurais precisam de grandes volumes de dados para seu aprendizado, para evitar problemas de *underfitting* realizamos um filtro nos termos cuja frequência seja inferior a 20, ou seja, desconsideramos os termos usados 19 vezes ou menos em todo corpus de documentos. Dessa maneira além de favorecer aos modelos neurais também diminuímos o risco de possíveis erros de digitação entre os termos indexados.

Tabela 2 – Resumo sobre os termos

Variável	**Valor**
Total de termos	1.767.752
Total termos selecionados	164.216
% termos Selecionados	9,2%
Menor Frequência	20
Maior Frequência (de)	12.462.911

Com isso, podemos sumarizar a seleção de termos como pode ser visto na Tabela 2, na qual foram localizados 1.767.752 termos, que, após filtrar pela frequência maior que 20 sobraram 164.216 termos, representando 9,2% de todos os termos. Assim sendo, a menor frequência do termo sendo 20 e a maior 12.462.911 com a palavra “de”.

5.3.2 Análise dos Termos

Com os termos filtrados, podemos analisar quais são os termos mais comuns e quais termos possuem maior escore via TF-IDF, na Tabela 3 foram excluídas palavras vazias (*stop words*) e pontuações. Podemos ver que os cinco termos mais comuns são “educação”, “avaliação”, “análise”, “ensino” e “saúde”, podemos ver também a quantidade de candidatos que utilizaram esse termo (*df*) e o escore TF-IDF de cada um deles.

Tabela 3 – Ranque de termos geral ordenado pela Frequência

Ranque	Termo	Frequência	Qtd. Candidatos	TF IDF
1	educacao	724163	59680	16.761490
2	avaliacao	572317	101123	9.477967
3	analise	511442	112091	8.043983
4	ensino	467978	59877	16.176100
5	saude	467819	43124	20.460771
6	estudo	466418	113576	7.815801
7	formacao	335274	51362	17.714524
8	brasil	303746	71116	13.469014
9	historia	295253	35869	22.059668
10	desenvolvimento	294820	74680	12.821339
11	producao	265680	60794	15.284857
12	social	258303	43930	19.299185
13	trabalho	244544	39986	20.381531
14	tratamento	235862	47291	18.246411
15	qualidade	234537	51790	17.114373
16	efeito	205167	56658	15.830364
17	sistema	204795	67345	13.714768
18	caso	202839	58752	15.372101
19	rio	199933	39737	20.126910
20	meio	198072	96406	9.302007

Em contrapartida, ao ordenar os termos pelo escore TF-IDF é possível avaliar os termos com maior impacto em todo corpus na Tabela 4. Os termos “imigração/colonização” e “lide-fgv” apareceram em apenas 1 candidato cada mas com uma grande frequência.

Podemos repetir essa mesma análise agrupando termos pelas 9 grandes áreas e assim identificar os termos com maior impacto em cada uma das áreas que são Ciências Exatas e da Terra, Ciências da Saúde, Ciências Agrárias, Engenharias, Ciências Biológicas, Ciências Sociais Aplicadas, Ciências Humanas na Tabela 5 Linguística, Letras e Artes, Outros na Tabela 6. Dessa maneira é possível notar que os termos com maior escore TF-IDF são bem diferentes de uma grande área para outra.

Ainda analisando os termos por grande área podemos verificar a similaridade entre elas de acordo com as palavras usadas pelos candidatos a especialista. Primeiramente, utilizando o *bag-of-words* descrito na Seção 5.4 localizamos para cada grande área o centróide referente a cada uma, ou seja, foi calculado a média do vetor de termos para cada e assim, a partir do centróide foi comparado entre os demais a partir da similaridade do cosseno, cuja Equação (4) pode ser conferida.

Tabela 4 – Ranque de termos geral ordenado pelo TF IDF

Ranque	Termo	Frequência	Qtd. Candidatos	TF IDF
1	imigracao/colonizacao	165	1	62,491714
2	lide-fgv	136	1	60,126028
3	benedictus	220	4	58,535502
4	sterechemistry	191	4	57,001426
5	aguaruna	175	4	56,051950
6	portuguesa-ensino-professor	127	2	55,930316
7	myxomycetes	934	62	55,480977
8	airglow	931	69	54,723591
9	teletandem	723	51	54,689589
10	rorschach	3.810	276	54,572971
11	otimidade	637	45	54,445721
12	tokamak	1.710	142	54,217819
13	pkhd1	179	6	54,193915
14	palestra/conferencia	109	2	54,165655
15	geolinguistica	994	81	54,141092
16	crianca-de-risco	178	6	54,135387
17	alib	606	45	54,025034
18	arteterapia	2.429	203	53,988263
19	liddy	82	1	53,933878
20	chiaffarelli	82	1	53,933878

A Tabela 7 exibe a matriz de comparação de uma grande área com as demais. Para cada coluna foi destacado em negrito o maior valor encontrado, isso significa que, quanto maior o valor, maior o conjunto de termos comuns usados entre as grandes áreas. Podemos ver que para grande área “Ciências Exatas e da Terra” a maior similaridade está com “Engenharias”, totalizando em um valor de 0,75, sendo o maior valor possível igual 1 é possível concluir que as áreas são próximas ao compararmos os termos utilizados em títulos e palavras chaves dos artigos. Outra dupla com grande similaridade se dá entre “Ciências Humanas” e “Ciências Sociais Aplicadas” com uma similaridade de 0,67.

Uma outra análise interessante é que a grande área de “Linguística, Letras e Artes” é a que possui a menor similaridade com as demais sendo as maiores similaridades com “Ciências Humanas” e “Ciências Sociais Aplicadas” com as respectivas similaridades de 0,56 e 0,37. Dessa maneira podemos concluir que há interseções entre diversas áreas e algumas são distantes entre si ao compararmos as médias de uso de termos.

5.4 Fraca Supervisão

As técnicas baseadas em redes neurais profundas têm sido usadas em diversas áreas da tecnologia, lidando de maneira muito efetiva em se tratando de dados não estruturados como texto e imagens. A recuperação de informação trabalha com esses tipos de dados, mas aplicar redes neurais nesse tipo de problema pode não ser trivial, ainda mais quando redes neurais ficam cada vez mais profundas e necessitando de mais dados rotulados.

Tabela 5 – Ranque de termos por Grande Área e ordenado pelo TF-IDF - Parte 1

	Ciências Exatas e da Terra				Ciências da Saúde			
Ranque	**Termo**	**Freq.**	**Cand.**	**TF IDF**	**Termo**	**Freq.**	**Cand.**	**TF IDF**
1	airglow	2524	69	62,70	sirolimus-eluting	1586	34	64,20
2	icrf	559	11	62,25	zotarolimus	670	19	60,48
3	decimetric	1111	30	61,97	drug-eluting	2437	97	59,77
4	tokamak	4454	142	61,19	stents	17510	472	59,42
5	tcabr	1802	65	60,45	intra-stent	1459	65	58,75
6	alfven	2563	96	60,23	odontoblastoides	919	39	58,5
7	ionospheric	3815	153	59,44	neointimal	1310	65	57,88
8	astrometria	1233	54	58,71	reestenose	4006	194	57,82
9	mesospheric	1046	47	58,32	in-stent	984	47	57,81
10	ionosphere	2923	141	58,17	intracoronario	3097	167	57,24
11	optogalvanica	454	16	57,91	sirolimus	4051	216	57,01
12	magnetocaloric	2999	150	57,87	reline	1124	64	56,76
13	meteor	1072	54	57,56	eluting	988	61	56,04
14	f-region	1218	64	57,40	vestasync	372	16	56,03
15	sterechemistry	192	4	57,0	safyre	437	21	55,90
	Ciências Agrárias				**Engenharias**			
Ranque	**Termo**	**Freq.**	**Cand.**	**TF IDF**	**Termo**	**Freq.**	**Cand.**	**TF IDF**
1	lacunatus	433	15	57,85	graded-channel	330	9	58,22
2	acarophenax	427	15	57,72	zvt	538	23	57,24
3	pre-antrais	3773	193	57,45	flextensional	346	14	56,12
4	nageli	811	40	57,27	diamante-cvd	841	51	55,94
5	charoles	3151	169	57,26	acilase	756	46	55,74
6	aphanothece	985	54	56,86	astrodinamica	801	51	55,54
7	gelatinosus	303	13	55,27	constructal	547	31	55,51
8	trichogramma	6764	398	55,14	castables	629	41	54,93
9	tospovirus	2290	170	54,95	manganito	248	10	54,78
10	enduro	1084	82	54,73	junctionless	223	9	54,29
11	bhv-5	1925	149	54,71	auto-reducao	538	37	54,25
12	desflurane	461	29	54,41	utbox	234	10	54,20
13	desflurano	468	30	54,33	relap5	475	33	53,88
14	duddingtonia	916	72	54,30	fd-fd	126	3	53,87
15	orius	1217	99	54,30	asfalticas	2296	197	53,83
	Ciências Biológicas				**Ciências Sociais Aplicadas**			
Ranque	**Termo**	**Freq.**	**Cand.**	**TF IDF**	**Termo**	**Freq.**	**Cand.**	**TF IDF**
1	nifa	1346	52	59,71	lide-fgv	136	1	60,12
2	muscarum	790	28	59,42	palestra/conferencia	108	2	54,05
3	myxomycetes	1407	62	58,80	gqvt	358	26	52,81
4	uchb	878	41	57,78	documentaria	1064	138	50,96
5	rhynchosciara	1384	74	57,39	insumo-produto	2154	293	50,33
6	seropedicae	4712	239	57,19	folkcomunicacao	1003	148	50,04
7	ucha	611	28	57,13	macromarketing	296	37	49,09
8	parabrachial	1022	55	57,04	subsidiarias	710	133	48,24
9	willistoni	1316	74	56,99	telejornalismo	2379	423	48,13
10	glurp	497	22	56,79	controladoria	2832	487	48,09
11	herbaspirillum	5604	290	56,69	pergulas	90	5	47,83
12	stannous	1930	115	56,69	biblioteconomia	2729	500	47,66
13	vaccinia	4752	256	56,67	elacom	138	15	46,96
14	av3v	1064	61	56,65	ciberjornalismo	219	34	46,95
15	d-2-hydroxyglutaric	607	30	56,63	midiatizacao	930	218	46,85

Em muitas aplicações possuir muitos dados rotulados de maneira rica pode ser muito dispendioso, tornando assim um gargalo na evolução de novas técnicas que só são possíveis de avançar quando é aplicável modelos não supervisionados em que não são necessários esse tipo de dados. Entretanto, não é uma tarefa fácil para RI o uso de modelos não supervisionados. Para resolver esse problema podemos usar modelos com fraca supervisão (*weak supervision*).

Um modelo de fraca supervisão na literatura está relacionado a modelos cujos dados

Tabela 6 – Ranque de termos por Grande Área e ordenado pelo TF-IDF - Parte 2

	Ciências Humanas				Outros			
Ranque	**Termo**	**Freq.**	**Cand.**	**TF IDF**	**Termo**	**Freq.**	**Cand.**	**TF IDF**
1	imigracao/colonizacao	165	1	62.49	samambaia-preta	137	42	41.82
2	bbt-br	359	10	58.45	airship	76	23	39.42
3	benedictus	218	4	58.43	relatvity	25	1	39.39
4	rorschach	4889	276	56.22	laeonereis	131	68	39.09
5	crianca-de-risco	180	6	54.25	adiantiformis	79	30	38.61
6	librada	104	2	53.62	acuta	140	89	38.29
7	habilidades/superdotacao	1029	104	52.67	inulinase	185	136	38.24
8	viking	201	11	52.198	hortela-rasteira	43	8	38.21
9	surdocegueira	601	62	51.90	rumohra	79	33	38.19
10	integralismo	953	113	51.52	hortela-japonesa	63	24	37.54
11	scotus	331	29	51.47	l-glutamico	75	38	37.13
12	siona	67	1	51.46	znco	46	15	36.49
13	delitivas	329	29	51.42	y-7571	52	22	36.14
14	pfister	573	66	51.12	nageli	67	40	35.95
15	visigoda	187	12	51.02	nova-unicamp	37	10	35.87

	Linguística, Letras e Artes			
Ranque	**Termo**	**Freq.**	**Cand.**	**TF IDF**
1	lacunatus	433	15	57.85
2	acarophenax	427	15	57.72
3	pre-antrais	3773	193	57.45
4	nageli	811	40	57.27
5	charoles	3151	169	57.26
6	aphanothece	985	54	56.86
7	gelatinosus	303	13	55.27
8	trichogramma	6764	398	55.14
9	tospovirus	2290	170	54.95
10	enduro	1084	82	54.73
11	bhv-5	1925	149	54.71
12	desflurane	461	29	54.41
13	desflurano	468	30	54.33
14	duddingtonia	916	72	54.30
15	orius	1217	99	54.30

Tabela 7 – Similaridade entre o centroide de cada Grande Área

	Ciências Exatas e da Terra	**Ciências da Saúde**	**Ciências Agrárias**	**Engenharias**	**Ciências Biológicas**	**Ciências Sociais Aplicadas**	**Ciências Humanas**	**Outros**	**Linguística, Letras e Artes**
Ciências Exatas e da Terra	1,0	0,39	0,43	**0,75**	0,52	0,32	0,28	**0,68**	0,17
Ciências da Saúde	0,39	1,0	0,38	0,37	0,57	0,34	0,40	0,49	0,20
Ciências Agrárias	0,43	0,38	1,0	0,41	0,57	0,27	0,22	0,58	0,12
Engenharias	**0,75**	0,37	0,41	1,0	0,40	0,37	0,25	0,65	0,14
Ciências Biológicas	0,52	**0,57**	0,57	0,40	1,0	0,24	0,24	0,62	0,13
Ciências Sociais Aplicadas	0,32	0,34	0,27	0,37	0,24	1,0	**0,67**	0,59	0,37
Ciências Humanas	0,28	0,40	0,22	0,25	0,24	**0,67**	1,0	0,53	**0,56**
Outros	0,68	0,49	**0,58**	0,65	**0,62**	0,59	0,53	1,0	0,28
Linguística, Letras e Artes	0,17	0,20	0,12	0,14	0,13	0,37	0,56	0,28	1,0

rotulados contém ruído ou possuem valores incorretos e quando um modelo supervisionado utiliza dados em que os rótulos foram gerados utilizando alguma outra metodologia automática ou utilizando algum tipo de heurística (HOFFMANN et al., 2011). Nesta Dissertação é

assumida a segunda definição.

Desse modo, como o objetivo final é treinar um modelo neural que necessita de uma lista de consultas; uma lista de documentos que possuem um relacionamento positivo, ou seja, documentos cujos julgamentos são de relevância; também são necessários documentos com relacionamento negativo dada uma consulta, ou seja, julgamento de irrelevância.

Como modelos neurais são gulosos com dados rotulados e estamos lidando com fraca supervisão, se faz necessária a obtenção de pseudo-julgamentos de relevância, a seguir é apresentada técnicas de extração de, dada uma consulta, obter documentos positivos e negativos juntamente com seus rótulos. Ou seja,

$$S = f(q, d),$$

em que S significa o score do pseudo-julgamento, q a consulta e d documento, sendo documentos variando entre relevantes $+d$ e não relevantes $-d$.

A preparação dos dados para um modelo com fraca supervisão se dá primeiramente na seleção de consultas (Subseção 5.4.1) para o modelo, seguindo dos documentos positivos (Subseção 5.4.2) e negativos (Subseção 5.4.3)

5.4.1 Consultas

Precisamos de um grande número de consultas para conseguirmos treinar modelos neurais e dentro da base de dados utilizada temos um conjunto de informações que tendem a condensar e resumir em tópicos os assuntos abordados pelos artigos. As palavras chaves descrevem de forma clara e objetiva os principais assuntos que permeiam o documento (YI; CHOI, 2012; STUBBS, 2010; EL-ARINI; GUESTRIN, 2011). Dessa maneira, esse conjunto de dados se mostra uma ótima opção para serem usados como conjunto de consultas.

Como a Plataforma Lattes possui campos abertos para os próprios usuários preencherem, alguns erros podem ser encontrados, como por exemplo o campo de palavras chaves serem preenchidos com resumos ou outros metadados que não são relacionados ao objetivo do campo. Para mitigar esse problema, filtramos palavras-chaves cujo tamanho da string excedia 100 caracteres.

Outro filtro realizado foi selecionar apenas as palavras chaves que ocorrem pelo menos uma vez no conjunto de todos os títulos de todos documentos, dessa maneira, mitigamos possíveis erros de digitação entre essas palavras, e garantindo pelo menos uma correspondência exata dentro do conjunto de dados possíveis.

Após todos os filtros, o conjunto de dados de consultas totalizaram em 906.010 registros na qual uma pequena amostra pode ser vista na Figura 6, em que temos um conjunto com as consultas para serem usadas em outras etapas do processo.

Figura 6 – Amostra de consultas

```
{"deep learning",
 "information retrieval",
 "hidrogenacao de polidienos",
 "analise da variancia.",
 "praticas discursivas midiaticas",
 {"..."} /*Demais Consultas e documentos*/
}
```

Fonte: Elaboração do autor

5.4.2 Documentos Positivos

Nesse ponto do processo chamaremos de documento cada conjunto de título de artigo, palavras chaves, ano de publicação, lista de autores e um identificador único. Com a lista de consultas detalhada anteriormente o processo de seleção de dados para fraca supervisão segue-se primeiramente escolhendo documentos com os maiores escores para cada.

Para gerar esse pseudo-rótulo utilizamos a técnica baseada em Modelo de Linguagem com suavização e distribuição de Dirichlet (ZHAI; LAFFERTY, 2017) cuja equação pode ser conferida Equação (9), um modelo tradicional e muitas vezes trazendo resultados melhores que BM25 (BENNETT; SCHOLER; UITDENBOGERD, 2008; ROBERTSON; ZARAGOZA et al., 2009) ou similares pois a sua forte motivação probabilística se mostra empiricamente melhor.

Para agilizar esse processo foi utilizada a ferramenta Elasticsearch (GORMLEY; TONG, 2015), um mecanismo de análise e busca, de código aberto, gratuito e que aproveita de recursos de computação distribuída. Possuindo em seu core as técnicas tradicionais mais famosas como LM e BM25, possui ainda possibilidade de criação de novas técnicas. Outra grande vantagem é que ele cria de maneira passiva todas as tabelas de índices, o que o torna extremamente rápido em sua execução.

O primeiro passo consiste em definir a configuração da criação dos índices, seguindo de inserir na ferramenta os documentos em formato json e por fim, realizar as todas as consultas utilizando um arquivo de configuração próprio.

A configuração de indexação usada consiste em:

1. Passar todos os termos para caixa baixa;
2. ignorar palavras vazias;
3. Cálculo de similaridade: *LMDirichlet*.

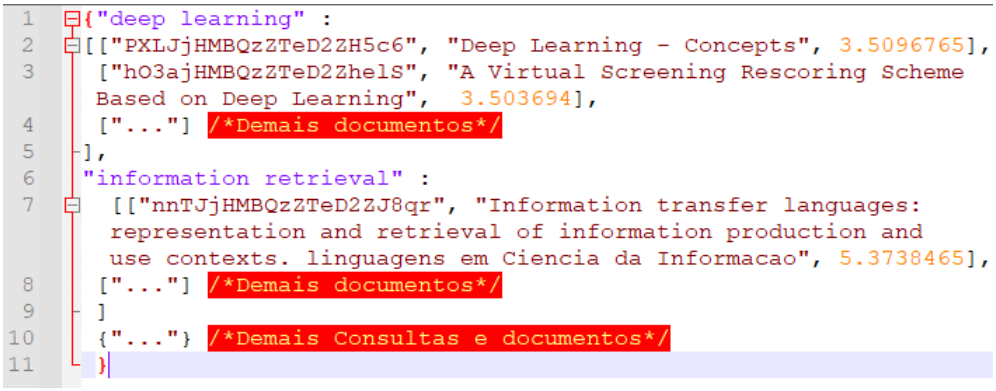
A configuração das consultas:

1. Correspondência considerando vários campos (Título e Palavras-chave) no modo *phrase*;

2. *slop* igual a 9, ou seja priorizar a sequência dos termos da consulta de maneira a considerar até 9 termos de distância da procurada;
3. Selecionar até 20 documentos com o maior score.

O Elasticsearch retorna arquivos JSON contendo como conteúdo os documentos relevantes, assim esses dados são registrados criando uma base na qual temos um dicionário cujas chaves são as consultas e o valor é uma lista de documentos com identificador único retornado pelo Elasticsearch, título do documento e escore como pode ser visto na Figura 7.

Figura 7 – Amostra de documentos positivos



Fonte: Elaboração do autor

5.4.3 Documentos Negativos

Como descrito anteriormente, ao treinar modelos neurais para problemas de de RI é extremamente importante definir um conjunto de dados que será usado em conjunto com as consultas e com documentos positivos, assim o modelo consegue diferenciar quais documentos são bons para consulta e quais não.

Em geral esses documentos negativos são selecionados aleatoriamente do corpus de documentos, mas como pode ser visto em Mitra, Diaz e Craswell (2017), os julgamentos de irrelevância entre consulta e documentos superam a opção aleatória. Assim sendo, seguindo a mesma lógica da fraca supervisão, aqui é proposto um método para selecionar o candidato mais distante da consulta e com isso, utilizar os seus documentos como irrelevante para a consulta.

A primeira opção para trabalhar com essa proposta é utilizar o *bag-of-words* de cada candidato e com isso utilizar esse grande e espaço vetor para calcular as similaridades do cosseno, dessa maneira teremos dois problemas, os vetores serem esparsos, ou seja, a maioria dos valores são zeros e por ele ser um vetor grande torna o processamento mais lento. Então, para encontrar os documentos mais distantes de cada consulta utilizamos um *deep autoencoder* para aprender uma representação dos candidatos em um espaço

dimensão reduzido e latente (SALAKHUTDINOV; HINTON, 2007), as consultas também são transformados nesse novo espaço dimensional e com isso, calcula-se a similaridade do cosseno entre a consulta e cada candidato, selecionando assim os candidatos mais distantes e extraímos seus documentos como amostras negativas igualando o número de documentos positivos e negativos para cada consulta.

O primeiro passo para essa etapa é criar uma base de dados em que cada linha representa um candidato e cada coluna será o escore calculado via TF-IDF de cada termo, esse grande vetor ou *bag-of-words* pode ser conferido na amostra da Figura 8, é registrado a identificação do candidato como chave e o valor é um dicionário de termos juntamente com seu escore. Optamos por essa representação para poupar alocação de memória da grande matriz que seria formada e em tempo de treinamento cada candidato é convertido em *bag-of-words*.

Figura 8 – Amostra de bag-of-words

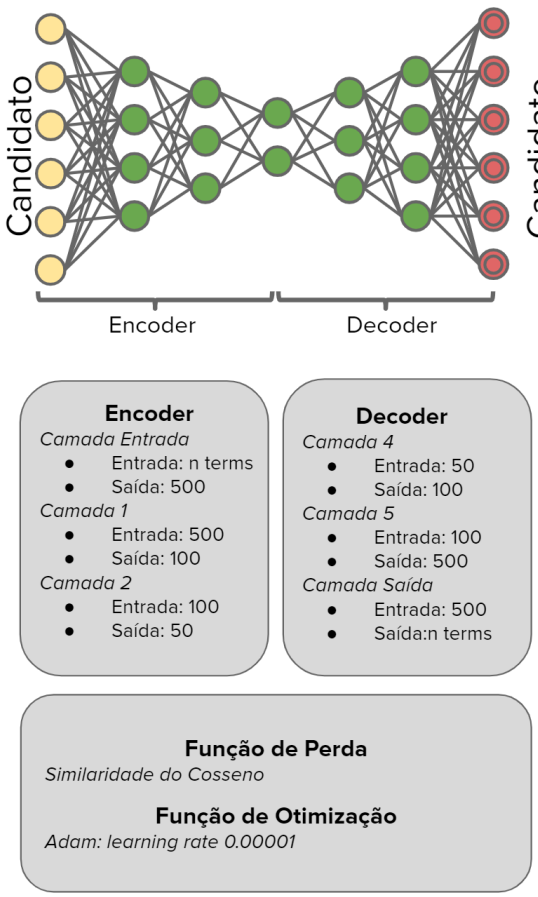
```
{"2628343082052753": {
    "estudo": 27,
    "da": 88,
    "mortalidade": 1,
    "dos": 40,
    "medicos": 31,
    {"..."}}, /*Demais termos*/
"3527197809276361": {
    "approaches": 1,
    "for": 58,
    "recommender": 7,
    "systems": 13,
    "learning": 6,
    {"..."}}, /*Demais termos*/
{"..."} /*Demais candidatos*/
}
```

Fonte: Elaboração do autor

Com esses dados, um modelo de *autoencoder* é treinado, sua arquitetura pode ser vista na Figura 9 em que podemos ver um formato de ampulheta, o objetivo é que, a cada camada o modelo possa aprender uma representação condensada dos dados, ou seja, ele reduz a cada passo a quantidade de colunas até chegar no fim do *encoder* no qual a dimensão é aumentada até o tamanho original do dado. O modelo então aprenderá ao calcular o erro da diferença entre o dado que entre e o dado que sai a partir da similaridade do cosseno entre os dois vetores.

A arquitetura é composta de 6 camadas, sendo as 3 primeiras denominadas *encoder* e as demais como *decoder*, sendo *encoder* possuindo 500, 100 e 50 neurônios e *decoder* possuindo a mesma sequência mas invertida. Entre a primeira e segunda camada e entre a penúltima e última foi utilizada a regularização chamada dropout que atua reduzindo o *overfitting* do modelo ao aleatoriamente desativando um percentual dos neurônios enquanto o treinamento é realizado, isso tenta garantir que os neurônios não se especializem o

Figura 9 – Arquitetura Deep Autoencoder



Fonte: Elaboração do autor

suficiente ao ponto de piorar o funcionamento da rede.

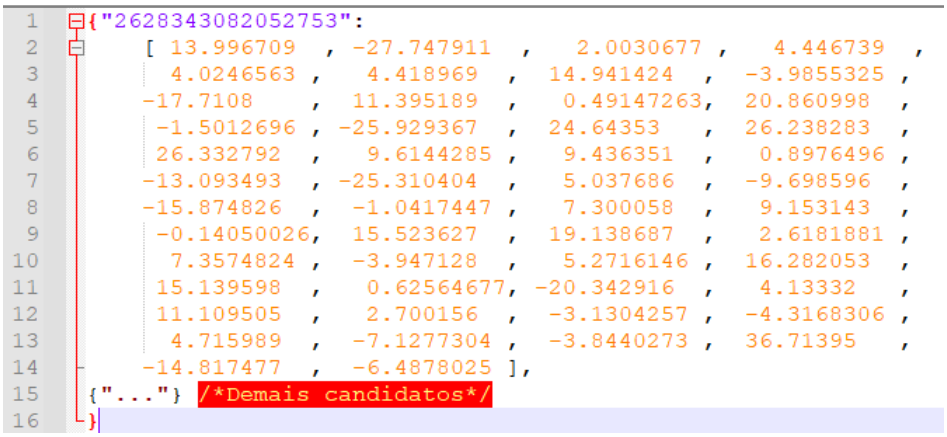
Após o treinamento do modelo, separamos a *encoder* e o utilizamos para transformar o dado de entrada para um espaço dimensão reduzido de 50 valores para cada candidato. Cada candidato é transformado nessa nova dimensão e os valores registrados como pode ser visto na amostra Figura 10. O conjunto de consultas também são transformados e assim, cada consulta é comparada com todos os candidatos e é calculada a similaridade do cosseno entre cada. Por fim, ordenando de maneira crescente e selecionando os candidatos mais distantes para cada consulta.

Então os documentos dos candidatos mais distantes são extraídos e registrados como sendo documentos com julgamento de irrelevância para a consulta, a quantidade de documentos extraídos é a mesma da quantidade de documentos positivos e indexada de maneira similar aos documentos positivos.

Logo, em resumo essa esse processo passa pelas etapas:

1. Agrega-se os documentos por autor, gerando um vetor de *bag-of-words* para cada candidato.

Figura 10 – Amostra dos candidatos após a transformação



Fonte: Elaboração do autor

2. Esse vetor é utilizado no treinamento do *autoencoder*;
3. Erro de reconstrução calculado pela similaridade do cosseno entre o vetor de entrada e o vetor retornado pela saída do *autoencoder*;
4. Após treinado, utiliza-se o *encoder* para transformar todos os candidatos e consultas em um espaço dimensional reduzido;
5. Compare-se a similaridade do cosseno entre cada consulta com cada candidato;
6. Dos candidatos mais distantes extrai-se seus documentos.

5.5 Dual Embedding LSTM

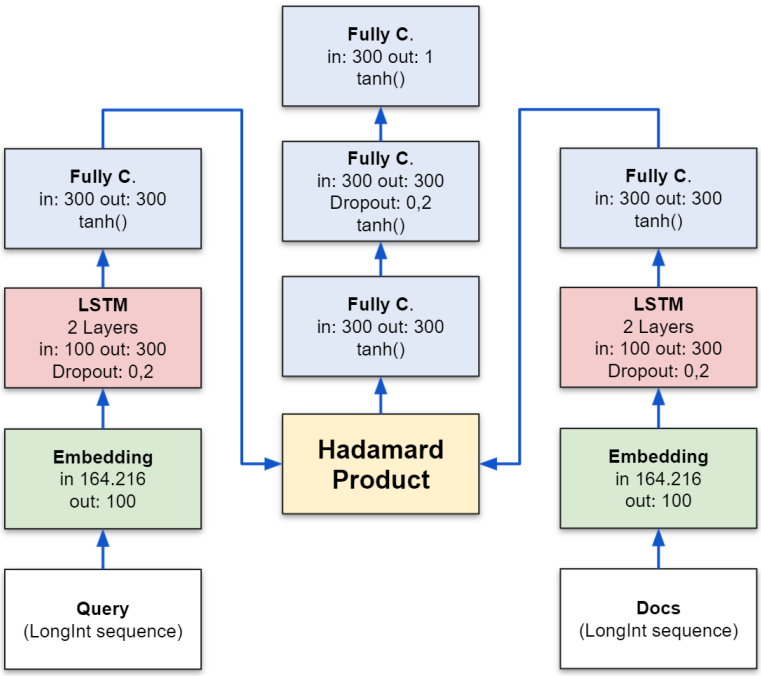
Com todos os dados e pseudo-julgamento de relevância e irrelevância segue-se então para a etapa de treinar um reclassificador, de documentos que irá receber informações da consulta e de documentos e no fim gerará um escore para essa relação. A intuição por trás da arquitetura *Dual Embedding LSTM* consiste em unir recursos de um modelo com correspondência exata, levando em conta a sequência e posição dos termos e modelo com representação distribuída em que os documentos são representados através de um vetor denso, então é uma arquitetura *Dual* pois possui dois nós de entrada que se unem posteriormente, *Embedding* pois transformam de alta dimensão para um espaço dimensional menor e *LSTM* ao fazer uso de camadas recorrentes com um controle de memória entre os termos para então gerar uma única representação vetorial para o documento e outra para a consulta.

A arquitetura *Dual Embedding LSTM* proposta pode ser vista na Figura 11, na qual possui dois nós de entrada, o primeiro para as consultas e o segundo para documentos. A entrada se dá passando termo a termo em formato de *Longint* que varia de zero até o tamanho máximo do vocabulário, esse termo é convertido em formato *one-hot* em que um grande vetor de zeros é formado e o valor correspondente àquele termo é preenchido com

1.

Cada termo segue passando por uma camada de textitEmbedding que reduzirá para um vetor de 100 dimensões que um a um passam para um conjunto de duas camadas *LSTM* (GERS; SCHMIDHUBER; CUMMINS, 1999). Camadas *LSTM* possuem capacidade de memorizar informações úteis e remover informações que podem atrapalhar o modelo. Esses cálculos são realizados em camadas internas de cada célula de *LSTM* com cálculos de sigmóide e todos os termos da sentença são usados nessa etapa se acumulando. Nessa etapa também é utilizada a regularização *Dropout* com valor de 20%. Quando o último termo é usado, as camadas de *LSTM* geram um vetor de 300 dimensões para toda a consulta e outro vetor para os documentos no outro nó do modelo que seguem para uma camada totalmente conectadas cuja a camada de ativação é *tanh*.

Figura 11 – Arquitetura Dual Embedding LSTM
Cada caixa representa uma camada sendo a informação contida em in a dimensão de entrada e out a de saída, em alguns casos temos camadas de ativação como $tanh$ e regularização $dropout$.



Fonte: Elaboração do autor

Após a consulta e o documento passarem pelas transformações, camadas recorrentes e totalmente conectadas, as informações de ambos são unidades através do produto Hadamard, que consiste em multiplicar valor a valor entre dois vetores ou matrizes. Se inspirando em modelos da literatura como Mitra, Diaz e Craswell (2017) o modelo segue com três camadas totalmente conectadas cuja camada de ativação é *tanh*, na penúltima camada há uma etapa de regularização com Dropout configurado em 20%. A última camada possui apenas uma dimensão variando entre -1 e 1

Como função de otimização foi utilizado *Adam* (KINGMA; BA, 2014) com uma taxa de aprendizado de $5E-5$ e regularização *L2*. A função de perda utilizada foi *MSE* calculada tentando prever se dada uma consulta e um documento eles possuem um relacionamento positivo ou negativo.

5.6 Re-ranqueamento

Após todas as etapas, o arcabouço proposto finaliza com uma etapa de em que o objetivo é re-ranquear. Como mencionado na Seção 5.2 devido ao alto consumo de hardware dos modelos neurais a melhor opção para se aplicar em produção em problemas de RI é utilizando uma técnica mais rápida para selecionar uma quantidade reduzida de documentos para serem aplicados o modelo proposto.

Assim sendo, para aproveitar o arcabouço na etapa de documentos positivos, é proposto utilizar o *Elasticsearch* com a configuração realizada anteriormente para selecionar os 2.000 documentos com o melhor escore e assim, aplicar o *Dual Embedding LSTM* para re-ranquear esses documentos, acumular os escores e por fim determinar quais os candidatos possuem o maior escore para uma dada consulta.

6 Análise e Discussão dos Resultados

Neste capítulo é apresentada a configuração de execução do experimento (Seção 6.1) e os resultados obtidos do arcabouço proposto (Seção 6.2).

6.1 Configuração e execução do experimento

O treinamento e avaliação utilizam dados da coleção *LExR* (MANGARAVITE et al., 2016), uma coleção pública extraída da Plataforma Lattes nos anos de 2014 e 2015. Esses dados foram transformados e organizados como apresentados no Capítulo 5.

O *autoencoder* profundo descrito na Subseção 5.4.3 foi treinado com 80% dos candidatos e os outros 20% utilizados para avaliar a performance dos hiperparâmetros. Após definida a melhor configuração todo o conjunto de dados foi usado para realizar o treinamento final do modelo e a cada iteração do treino são usados 20 amostras por vez (*batch size*).

Para o modelo *Dual Embedding LSTM* descrito na Seção 5.5 obteve no final das etapas de fraca supervisão os seguintes totais de dados com os pseudo julgamento de relevância e irrelevância:

- 164.216 Termos
- 906.010 Consultas
- 4.071.628 Documentos utilizados
- 18.120.200 Triplas de Consulta + Documento + Escore (incluindo metade positivos e metade negativos)

Esses dados foram separados em 80% para treinamento e 20% para validação do treinamento e a cada iteração são usados 10 amostras de dados (*batch size*), para testar o modelo proposto foi utilizado o gabarito do *LExR* com 235 tópicos de consulta e 1.635 julgamentos de relevância criados por humanos. A avaliação final se dá pela métrica nDCG, um forte indicador de modelos de RI que leva em conta a presença de documento relevante, a ordem resultante da busca e o peso dos documentos.

As arquiteturas de redes neurais propostas foram desenvolvidas utilizando *Pytorch* e máquina utilizada para todo experimento, incluindo servidor Elasticsearch e treinamento de dois modelos neurais possui seguinte configuração:

- CPU AMD Ryzen 5 3600X
- 32GB Memória RAM DDR4
- NVIDIA GeForce RTX 2070 com 8GB vRAM

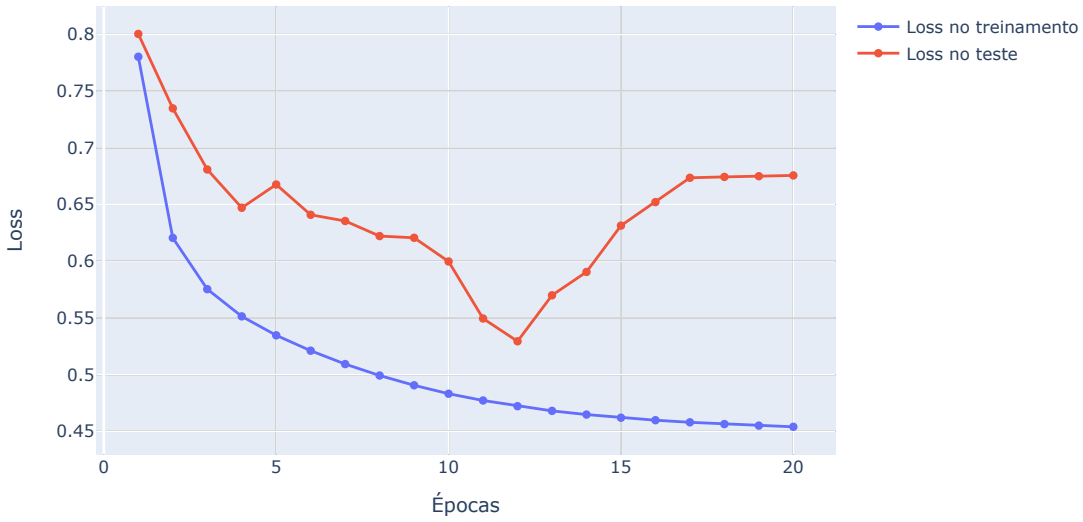
6.2 Resultados

Nesta seção são exibido os resultados obtidos pelos modelos proposto divididos em resultados obtidos pelo autoencoder profundo (Subseção 6.2.1) e os resultados obtidos pelo *Dual Embedding LSTM* (Subseção 6.2.2)

6.2.1 Resultados autoencoder profundo

A evolução do treinamento do *autoencoder* profundo pode ser vista na Figura 12 em que no eixo x são registradas as épocas variando de 1 até 20 e o eixo y com os valores da função de perda. A linha azul mostra o comportamento dos dados de treinamento e a linha vermelha os dados de teste. É possível notar que enquanto a perda do treinamento reduz a perda no teste oscila e a partir da época 12 começa a aumentar, exibindo um comportamento de *overfitting*, assim sendo o modelo final foi treinado por 12 épocas.

Figura 12 – Gráfico da função de perda ao longo das épocas

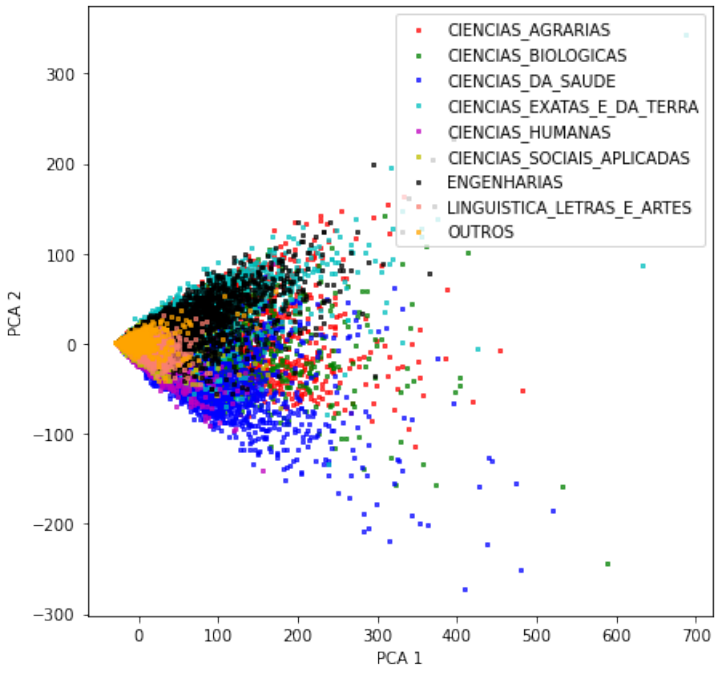


Fonte: Elaboração do autor

Com isso o *encode* é separado do modelo e usado para transformar todos os dados necessários. Para avaliar essa transformação utilizamos um modelo de redução de dimensionalidade PCA (Principal Component Analysis (ABDI; WILLIAMS, 2010)) para ser possível criar um gráfico e visualizarmos se há algum padrão nos dados.

A Figura 13 exibe essa visualização em que os eixos são os valores gerados pelo modelo e as cores representam cada grande área. É possível notar uma leve separação desses dados, ainda mais se compararmos de maneira isolada algumas grandes áreas como

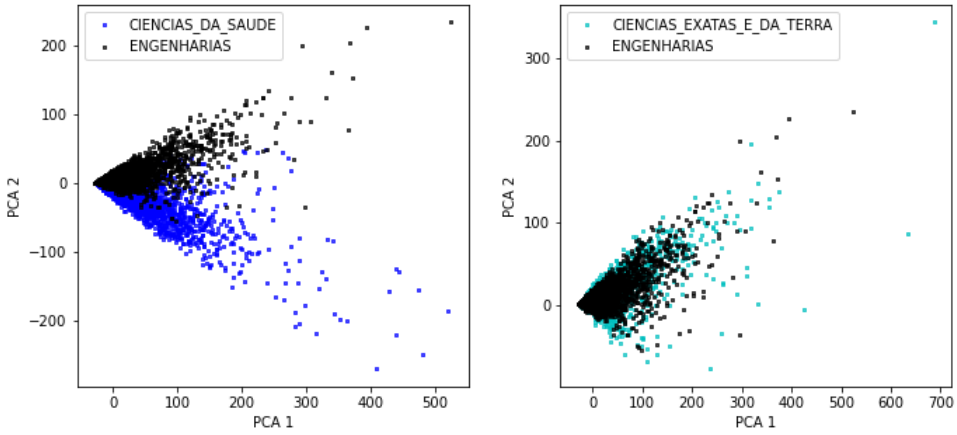
Figura 13 – PCA dos embeddings dos candidatos em todas grandes áreas



Fonte: Elaboração do autor

pode ser visto na Figura 14 em que se observa uma clara separação entre as grandes áreas de Ciências da Saúde e Engenharias; já com áreas que possuem interseções podemos ver uma distribuição parecida ao comparar Ciências Exatas e da Terra com Engenharias.

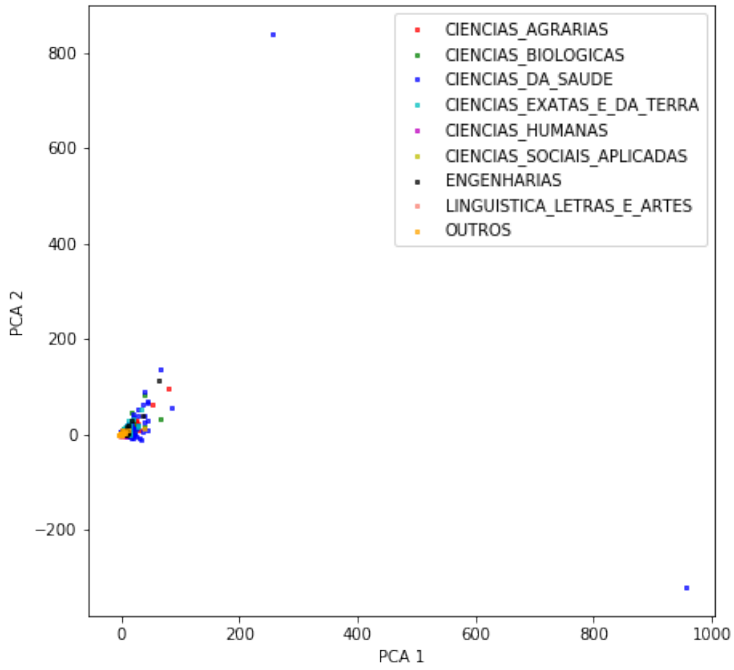
Figura 14 – PCA dos embeddings dos candidatos
Comparando a grande área de Engenharias com Ciência da Saúde e Ciências Exatas e da Terra



Fonte: Elaboração do autor

Na Figura 15 podemos ver como é o comportamento dos grandes grupos caso seja utilizado os dados do *bag-of-words* diretamente no treinamento do PCA. Como é um dado com alta dimensionalidade o PCA não consegue realizar uma boa separação dos dados agrupando praticamente todos os pontos em um único cluster.

Figura 15 – PCA dos bag-of-words dos candidatos em todas grandes áreas



Fonte: Elaboração do autor

6.2.2 Resultados Dual Embedding LSTM

Foram treinados dois conjuntos de modelos *Dual Embedding LSTM* diferenciando pela formação dos dados de fraca supervisão para os documentos negativos, o primeiro grupo foi treinado utilizando documentos negativos selecionados através do uso do *autoencoder* e o segundo grupo composto por documentos selecionados aleatoriamente, sendo que para serem analisados foram treinados três modelos com três conjuntos diferentes de dados de documentos negativos selecionados aleatoriamente e retirada a média entre os três modelos para fins de avaliação.

A Tabela 8 registra os resultados obtidos ao longo das épocas para o modelo que utiliza o *autoencoder* e a média de 3 modelos que usam dados aleatórios para os documentos negativos. A coluna nDCG@10 considera o cálculo do nDCG para os 10 primeiros resultados da busca comparada com o gabarito, já a perda e acurácia são métricas retiradas da validação do modelo com os dados de teste. Podemos ver que o uso de *autoencoder* aprimorou o treinamento do modelo, atingindo um nDCG@10 de 0,191 e colaborando para que ele atinja um resultado mais elevado se comparado com as médias dos documentos aleatório cujo maior valor registrado foi de 0,170 para o nDCG@10.

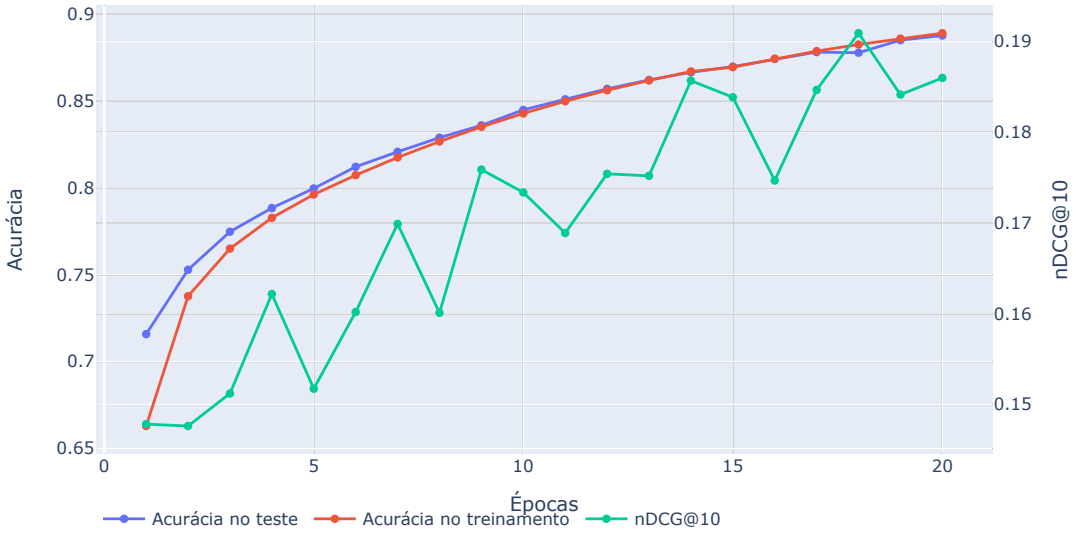
A Figura 16 exibe a evolução do modelo final proposto em que o eixo x representa as épocas, o primeiro eixo y a esquerda com o valor da acurácia do modelo e o segundo eixo y com o valor de nDCG@10. As linhas azuis e vermelhas representam respectivamente a acurácia do teste e treinamento, já a linha verde representa o valor de nDCG utilizando o

Tabela 8 – Evolução do modelo por época

	Autoencoder			Média 3 Aleatórios		
Época	**nDCG@10**	**Perda**	**Acurácia**	**nDCG@10**	**Perda**	**Acurácia**
1	0,148	0,002	0,716	0,134	0,002	0,717
2	0,148	0,002	0,753	0,143	0,002	0,745
3	0,151	0,002	0,775	0,141	0,002	0,758
4	0,162	0,001	0,788	0,144	0,002	0,772
5	0,152	0,001	0,800	0,138	0,002	0,782
6	0,16	0,001	0,812	0,143	0,001	0,791
7	0,17	0,001	0,821	0,154	0,001	0,799
8	0,16	0,001	0,829	0,143	0,001	0,807
9	0,176	0,001	0,836	0,150	0,001	0,815
10	0,173	0,001	0,845	0,154	0,001	0,821
11	0,169	0,001	0,851	0,152	0,001	0,827
12	0,175	0,001	0,857	0,159	0,001	0,831
13	0,175	0,001	0,862	0,162	0,001	0,836
14	0,186	0,001	0,866	**0,170**	**0,001**	**0,839**
15	0,184	0,001	0,870	0,159	0,001	0,837
16	0,175	0,001	0,874	0,153	0,001	0,836
17	0,185	0,001	0,878	0,154	0,001	0,841
18	**0,191**	**0,001**	**0,878**	0,152	0,001	0,840
19	0,184	0,001	0,885	0,149	0,001	0,838
20	0,186	0,001	0,888	0,153	0,001	0,837

segundo eixo y. Pode-se observar que apesar de oscilar o nDCG tende a crescer ao longo das épocas, chegando ao seu maior valor na época 18.

Figura 16 – Gráfico da função da acurácia e nDCG@10 ao longo das épocas



Fonte: Elaboração do autor

Por fim, a Tabela 9 exibe a comparação dos melhores modelos da literatura com os melhores modelos propostos neste trabalho, assim sendo, temos 3 modelos baseados em teoria da informação proposto por Mangaravite e Santos (2016), o modelo base usado no

Elasticsearch tento para selecionar os documentos positivos para fraca supervisão quanto para selecionar os documentos serão re-ranqueados (LM Dirichlet) e dois modelos usando *Dual Embedding LSTM* sendo um com documentos negativos selecionados aleatoriamente o outro com documentos selecionados utilizando o *autoencoder*. Podemos ver que os três modelos apresentados superam a literatura comparada e o arcabouço completo proposto proporciona o melhor resultado de nDCG@10.

Tabela 9 – Performance entre diferentes modelos.

Método	nDCG@10
Inf. Theoretic ρKL (MANGARAVITE; SANTOS, 2016)	0.135
Inf. Theoretic ρH - ψDC (MANGARAVITE; SANTOS, 2016)	0.146
Inf. Theoretic ρH - ψSDC (MANGARAVITE; SANTOS, 2016)	0.164
LM Dirichlet	**0.178**
Dual Embedding LSTM + Aleatório	**0.170**
Dual Embedding LSTM + Autoencoder	**0.191**

7 Conclusões

Buscando avançar a ciência em buscas de especialistas, este trabalho propõe um arcabouço que se utiliza da fraca supervisão para treinar uma rede neural profunda. Uma proposta inédita em vários aspectos como uso de redes neurais em buscas de especialistas em especial na área acadêmica bem como a apresentação de métodos para gerar pseudo julgamentos de relevância a irrelevância e uma seleção para um conjunto de consultas.

Os resultados são positivos, todos os métodos testados apresentam uma melhora sobre os *baselines* da literatura comparados, assim sendo, a fraca supervisão pode ser uma opção em se tratando do treinamento de modelos neurais na busca de especialistas na base de dados da Plataforma Lattes

Outro avanço foi a proposta de geração de documentos com julgamentos de irrelevância ou documentos negativos para as consultas. O padrão da literatura consiste em selecionar aleatoriamente documentos entre todo o corpus, e ao compararmos a média de três modelos treinados usando documentos aleatórios podemos ver uma melhoria ao usar um *autoencoder* para transformar os candidatos em um espaço dimensional reduzido e selecionar os documentos dos candidatos mais distantes para cada consulta.

Por fim, o modelo *Dual Embedding LSTM* juntamente com o uso do *autoencoder* conseguiram superar todas as técnicas da literatura comparadas, de maneira a proporcionar as vantagens de modelos com representação distribuída em camadas que unem o resultado das LSTM e fazendo uso de representação exata ao considerar a sequência de aparições dos termos.

7.1 Publicações

A seguir, são apresentadas as publicações elaboradas a partir dos estudos realizados nesta Dissertação, inicialmente explorando os aspectos da representação dos candidatos as especialistas, criando a proposta de seleção de documentos com *autoencoder*, abaixo seguem os trabalhos completos publicados em anais de congressos internacionais que incluem apresentações:

- SOUSA, S. J. ; DIAS, T. M. R. ; PINTO, A. L. . A STRATEGY FOR EXPERT RECOMMENDATION FROM OPEN DATA AVAILABLE ON THE LATTES PLATFORM. In: **16th International Conference on Information Systems and Technology Management - CONTECSI2019**, 2019, São Paulo. Anais do 16th International Conference on Information Systems and Technology Management-CONTECSI2019. São Paulo: USP, 2019. v. 01. p. 189-198.

- SOUSA, S. J. ; DIAS, T. M. R. ; PINTO, A. L. . Uma estratégia para recomendação de especialistas a partir de dados abertos disponíveis na Plataforma Lattes. In: **IX Encuentro Ibérico de la Asociación de Educación e Investigación en Ciencia de la Información de Iberoamérica y el Caribe (EDICIC)**, 2019, Barcelona. Anais do IX Encuentro Ibérico de la Asociación de Educación e Investigación en Ciencia de la Información de Iberoamérica y el Caribe (EDICIC). Barcelona: Universidade de Barcelona, 2019. v. 01. p. 50-60.

Por fim, os resultados finais foram publicados e apresentados no DIONE 2021, sendo o texto publicado como capítulo de Livro:

- DE SOUSA, Sergio Jose; DIAS, Thiago Magela Rodrigues; PINTO, Adilson Luiz. Neural Weak Supervision Model for Search of Specialists in Scientific Data Repository. In: **EAI DIONE 2021- 2nd EAI International Conference on Data and Information in Online Environments**.
- DE SOUSA, Sergio Jose; DIAS, Thiago Magela Rodrigues; PINTO, Adilson Luiz. Neural Weak Supervision Model for Search of Specialists in Scientific Data Repository. In: **International Conference on Data and Information in Online**. Springer, Cham, 2021. p. 286-296.

7.2 Trabalhos Futuros

A partir do estudo apresentado, diversas novas frentes de pesquisas podem surgir a partir deste trabalho, como por exemplo:

- Propor e verificar outras arquiteturas neurais utilizando os dados de fraca supervisão gerados, assim reduzindo a restrição de ser necessário grandes volumes de dados rotulados por humanos;
- Pesquisas visando aprimorar a seleção de dados de fraca supervisão, como por exemplo testar os novos modelos de embedding agnóstico de linguagem como BERT[1] e similares;
- Propor outras funções de agregação de escore dos documentos para compor o escore final dos candidatos;
- Expandir o arcabouço proposto para outras bases de busca de especialistas e checar o desempenho em outros cenários de dados.

[1] LaBSE Language-agnostic BERT Sentence Embedding : https://dblp.uni-trier.de/

Referências

ABDI, H.; WILLIAMS, L. J. Principal component analysis. **Wiley interdisciplinary reviews: computational statistics**, Wiley Online Library, v. 2, n. 4, p. 433–459, 2010. Citado na página 44.

AFFONSO, F.; DIAS, T. M. R.; SANTIAGO, M. de O. A strategy for co-authorship recommendation: Analysis using scientific data repositories. In: SPRINGER. **International Conference on Data and Information in Online**. [S.l.], 2020. p. 167–178. Citado na página 22.

BAJAJ, P. et al. Ms marco: A human generated machine reading comprehension dataset. **arXiv preprint arXiv:1611.09268**, 2016. Citado na página 20.

BALOG, K.; AZZOPARDI, L.; RIJKE, M. D. Formal models for expert finding in enterprise corpora. In: **Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.: s.n.], 2006. p. 43–50. Citado na página 13.

BALOG, K. et al. Broad expertise retrieval in sparse data environments. In: ACM. **Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.], 2007. p. 551–558. Citado na página 1.

BALOG, K. et al. Expertise retrieval. **Foundations and Trends® in Information Retrieval**, Now Publishers, Inc., v. 6, n. 2–3, p. 127–256, 2012. Citado 3 vezes nas páginas 1, 5 e 16.

BALOG, K.; RIJKE, M. de. Finding experts and their eetails in e-mail corpora. In: **Proceedings of the 15th international conference on World Wide Web**. [S.l.: s.n.], 2006. p. 1035–1036. Citado 2 vezes nas páginas 12 e 13.

BENNETT, G.; SCHOLER, F.; UITDENBOGERD, A. A comparative study of probabilistic and language models for information retrieval. In: **Proceedings of the nineteenth conference on Australasian database-Volume 75**. [S.l.: s.n.], 2008. p. 65–74. Citado na página 36.

BERENDSEN, R. et al. On the assessment of expertise profiles. **Journal of the American Society for information Science and Technology**, Wiley Online Library, v. 64, n. 10, p. 2024–2044, 2013. Citado na página 1.

BROWN, T. B. et al. Language models are few-shot learners. **arXiv preprint arXiv:2005.14165**, 2020. Citado na página 20.

CHI, M. T.; GLASER, R.; FARR, M. J. **The nature of expertise**. [S.l.]: Psychology Press, 2014. Citado na página 1.

CRASWELL, N.; VRIES, A. P. de; SOBOROFF, I. Overview of the trec 2005 enterprise track. In: **Trec**. [S.l.: s.n.], 2005. v. 5, p. 1–7. Citado na página 1.

DEHGHANI, M. et al. Neural ranking models with weak supervision. In: ACM. **Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval**. [S.l.], 2017. p. 65–74. Citado 2 vezes nas páginas 20 e 26.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018. Citado na página 20.

DIAS, P. M. et al. Um estudo sobre a produção científica do conjunto de docentes dos programas de pós-graduação área interdisciplinar no brasil. **Pesquisa Brasileira em Ciência da Informação e Biblioteconomia; Vol. 13, No 1 (2018)**, v. 24, n. 2, 2018. Citado na página 22.

DIAS, T. M. R.; MOITA, G. F. A method for the identification of collaboration in large scientific databases. **Em Questão**, Universidade Federal do Rio Grande do Sul, v. 21, n. 2, p. 140–161, 2015. Citado na página 2.

DIAS, T. M. R. et al. Caracterização dos bolsistas de produtividade em pesquisa do cnpq a partir de dados da plataforma lattes. **ENCONTRO BRASILEIRO DE BIBLIOMETRIA E CIENTOMETRIA**, v. 5, p. A90, 2016. Citado na página 22.

DIGIAMPIETRI, L. A.; FERREIRA, J. E. Desambiguação de nomes de autores para a identificação automática de perfis acadêmicos. **Em Questão**, v. 24, n. 2, p. 37–54, 2018. Citado na página 15.

EL-ARINI, K.; GUESTRIN, C. Beyond keyword search: discovering relevant scientific literature. In: **Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.: s.n.], 2011. p. 439–447. Citado na página 35.

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. Learning to forget: Continual prediction with lstm. IET, 1999. Citado na página 41.

GOMES, J. O.; DIAS, T. M. R.; MOITA, G. F. Uma análise temporal dos principais tópicos de pesquisa da ciência brasileira a partir das palavras-chave de publicações científicas. **Pesquisa Brasileira em Ciência da Informação e Biblioteconomia; Vol. 13, No 1 (2018)**, v. 24, n. 2, 2018. Citado na página 22.

GONÇALVES, R.; DORNELES, C. F. Automated expertise retrieval: a taxonomy-based survey and open issues. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 52, n. 5, p. 1–30, 2019. Citado 2 vezes nas páginas 16 e 17.

GORMLEY, C.; TONG, Z. **Elasticsearch: the definitive guide: a distributed real-time search and analytics engine**. [S.l.]: "O'Reilly Media, Inc.", 2015. Citado na página 36.

GUO, J. et al. A deep relevance matching model for ad-hoc retrieval. In: ACM. **Proceedings of the 25th ACM International on Conference on Information and Knowledge Management**. [S.l.], 2016. p. 55–64. Citado na página 17.

HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. **science**, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006. Citado na página 3.

HOFFMANN, R. et al. Knowledge-based weak supervision for information extraction of overlapping relations. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1**. [S.l.], 2011. p. 541–550. Citado na página 34.

HUSAIN, O. et al. Expert finding systems: A systematic review. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 9, n. 20, p. 4250, 2019. Citado na página 16.

JÄRVELIN, K.; KEKÄLÄINEN, J. Cumulated gain-based evaluation of ir techniques. **ACM Transactions on Information Systems (TOIS)**, ACM New York, NY, USA, v. 20, n. 4, p. 422–446, 2002. Citado na página 14.

JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. **Journal of documentation**, MCB UP Ltd, 1972. Citado 2 vezes nas páginas 6 e 7.

KEIKHA, M.; GERANI, S.; CRESTANI, F. Relevance stability in blog retrieval. In: **Proceedings of the 2011 ACM Symposium on Applied Computing**. [S.l.: s.n.], 2011. p. 1119–1123. Citado na página 15.

KEIKHA, M.; GERANI, S.; CRESTANI, F. Temper: A temporal relevance feedback method. In: SPRINGER. **European Conference on Information Retrieval**. [S.l.], 2011. p. 436–447. Citado na página 15.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014. Citado na página 42.

LANE, J. Let's make science metrics more scientific. **Nature**, Nature Publishing Group, v. 464, n. 7288, p. 488, 2010. Citado na página 2.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Citado na página 2.

LI, X. et al. A network approach to expertise retrieval based on path similarity and credit allocation. **Journal of Economic Interaction and Coordination**, Springer, p. 1–33, 2021. Citado na página 16.

LIN, S. et al. A survey on expert finding techniques. **Journal of Intelligent Information Systems**, Springer, v. 49, n. 2, p. 255–279, 2017. Citado na página 1.

LIU, X.; CROFT, W. B. Statistical language modeling for information retrieval. **Annual Review of Information Science and Technology**, Wiley Online Library, v. 39, n. 1, p. 1–31, 2005. Citado na página 11.

LUHN, H. P. A statistical approach to mechanized encoding and searching of literary information. **IBM Journal of research and development**, IBM, v. 1, n. 4, p. 309–317, 1957. Citado na página 7.

MACDONALD, C.; OUNIS, I. Voting for candidates: adapting data fusion techniques for an expert search task. In: **Proceedings of the 15th ACM international conference on Information and knowledge management**. [S.l.: s.n.], 2006. p. 387–396. Citado 2 vezes nas páginas 12 e 13.

MANGARAVITE, V.; SANTOS, R. L. On information-theoretic document-person associations for expert search in academia. In: ACM. **Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval**. [S.l.], 2016. p. 925–928. Citado 4 vezes nas páginas 4, 16, 47 e 48.

MANGARAVITE, V. et al. The lexr collection for expertise retrieval in academia. In: ACM. **Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval**. [S.l.], 2016. p. 721–724. Citado 5 vezes nas páginas 22, 23, 24, 27 e 43.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2013. p. 3111–3119. Citado na página 18.

MITRA, B.; CRASWELL, N. et al. An introduction to neural information retrieval. **Foundations and Trends® in Information Retrieval**, Now Publishers, Inc., v. 13, n. 1, p. 1–126, 2018. Citado 2 vezes nas páginas 2 e 21.

MITRA, B.; DIAZ, F.; CRASWELL, N. Learning to match using local and distributed representations of text for web search. In: INTERNATIONAL WORLD WIDE WEB CONFERENCES STEERING COMMITTEE. **Proceedings of the 26th International Conference on World Wide Web**. [S.l.], 2017. p. 1291–1299. Citado 6 vezes nas páginas 18, 19, 20, 27, 37 e 41.

MITRA, B. et al. A dual embedding space model for document ranking. **arXiv preprint arXiv:1602.01137**, 2016. Citado 2 vezes nas páginas 17 e 18.

MOREIRA, T. H. J. **Genealogia acadêmica brasileira: uma caracterização da relação orientador-orientado no Brasil**. Tese (Doutorado) — Dissertação (Mestrado em Modelagem Matemática e Computacional). Belo . . . , 2018. Citado na página 15.

NOGUEIRA, R.; JIANG, Z.; LIN, J. Document ranking with a pretrained sequence-to-sequence model. **arXiv preprint arXiv:2003.06713**, 2020. Citado na página 20.

OPITZ, D.; MACLIN, R. Popular ensemble methods: An empirical study. **Journal of artificial intelligence research**, v. 11, p. 169–198, 1999. Citado na página 13.

RAFFEL, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. **arXiv preprint arXiv:1910.10683**, 2019. Citado na página 20.

RIBEIRO, I. S. et al. On tag recommendation for expertise profiling: A case study in the scientific domain. In: **Proceedings of the eighth ACM international conference on web search and data mining**. [S.l.: s.n.], 2015. p. 189–198. Citado na página 23.

ROBERTSON, S.; ZARAGOZA, H. 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. Citado 2 vezes nas páginas 20 e 36.

ROBERTSON, S.; ZARAGOZA, H.; TAYLOR, M. Simple bm25 extension to multiple weighted fields. In: **Proceedings of the thirteenth ACM international conference on Information and knowledge management**. [S.l.: s.n.], 2004. p. 42–49. Citado na página 10.

ROBERTSON, S. E. The probability ranking principle in ir. **Journal of documentation**, v. 33, n. 4, p. 294–304, 1977. Citado na página 9.

ROSENBLATT, F. **The perceptron, a perceiving and recognizing automaton Project Para**. [S.l.]: Cornell Aeronautical Laboratory, 1957. Citado na página 2.

SALAKHUTDINOV, R.; HINTON, G. Semantic hashing. **RBM**, v. 500, n. 3, p. 500, 2007. Citado 3 vezes nas páginas 8, 9 e 38.

SALTON, G.; WONG, A.; YANG, C.-S. A vector space model for automatic indexing. **Communications of the ACM**, ACM New York, NY, USA, v. 18, n. 11, p. 613–620, 1975. Citado na página 6.

SANTIAGO, M. de O.; DIAS, T. M. R.; AFFONSO, F. Characterization of women's scientific participation in brazil. In: SPRINGER. **International Conference on Data and Information in Online**. [S.l.], 2020. p. 210–221. Citado na página 22.

SMALHEISER, N. R.; TORVIK, V. I. Author name disambiguation. **Annual review of information science and technology**, Wiley Online Library, v. 43, n. 1, p. 1–43, 2009. Citado na página 15.

STUBBS, M. Three concepts of keywords. **Keyness in texts**, Amsterdam: John Benjamins, p. 21–42, 2010. Citado na página 35.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2014. p. 3104–3112. Citado na página 20.

TANG, J. et al. Arnetminer: extraction and mining of academic social networks. In: ACM. **Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.], 2008. p. 990–998. Citado na página 1.

VOORHEES, E. M. et al. Overview of trec 2004. In: **Trec**. [S.l.: s.n.], 2004. Citado na página 20.

WILKINSON, R.; HINGSTON, P. Using the cosine measure in a neural network for document retrieval. In: **Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.: s.n.], 1991. p. 202–210. Citado na página 8.

YI, S.; CHOI, J. The organization of scientific knowledge: the structural characteristics of keyword networks. **Scientometrics**, Springer, v. 90, n. 3, p. 1015–1026, 2012. Citado na página 35.

ZAHEDI, M. et al. Time sensitive blog retrieval using temporal properties of queries. **Journal of Information Science**, SAGE Publications Sage UK: London, England, v. 43, n. 1, p. 103–121, 2017. Citado na página 15.

ZHAI, C.; LAFFERTY, J. A study of smoothing methods for language models applied to ad hoc information retrieval. In: ACM. **ACM SIGIR Forum**. [S.l.], 2017. v. 51, n. 2, p. 268–276. Citado 3 vezes nas páginas 3, 11 e 36.