

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA**

LEONARDO SANTIAGO BENITEZ PEREIRA

**COMPARAÇÃO DE TÉCNICAS DE RECUPERAÇÃO DE INFORMAÇÃO
APLICADAS A UM CONJUNTO DE CHAMADOS DE SUPORTE DE TI**

FLORIANÓPOLIS, 2022.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA**

LEONARDO SANTIAGO BENITEZ PEREIRA

**COMPARAÇÃO DE TÉCNICAS DE RECUPERAÇÃO DE INFORMAÇÃO
APLICADAS A UM CONJUNTO DE CHAMADOS DE SUPORTE DE TI**

Este trabalho foi julgado adequado para obtenção do título de Engenheiro Eletrônico em dezembro de 2022 e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Eletrônica do Instituto Federal de Educação Ciência, e Tecnologia de Santa Catarina.

Orientador:
Prof. Robinson Pizzio, doutor em Engenharia Elétrica

FLORIANÓPOLIS, 2022.

Ficha de identificação da obra elaborada pelo autor.

Pereira, Leonardo Santiago Benitez
**Comparação de técnicas de Recuperação de Informação
aplicadas a um conjunto de chamados de suporte de TI** / Leonardo
Santiago Benitez Pereira; orientação de Robinson
Pizzio. - Florianópolis, SC, 2022.
52 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. Bacharelado
em Engenharia Eletrônica. Departamento Acadêmico
de Eletrônica.
Inclui Referências.

1. Recuperação de Informação. 2. Aprendizado de Máquina.
3. Processamento de Linguagem Natural. 4. Chamados
de suporte. 5. Tecnologias de Informação. I. Pizzio,
Robinson. II. Instituto Federal de Santa Catarina.
III. Comparação de técnicas de Recuperação de Informação
aplicadas a um conjunto de chamados de suporte de

**COMPARAÇÃO DE TÉCNICAS DE RECUPERAÇÃO DE INFORMAÇÃO
APLICADAS A UM CONJUNTO DE CHAMADOS DE SUPORTE DE TI**

LEONARDO SANTIAGO BENITEZ PEREIRA

Este trabalho foi julgado adequado para obtenção do título de Engenheiro Eletrônico e aprovado na sua forma final pela banca examinadora do Curso Engenharia Eletrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 26 de dezembro, 2022.

Orientador:

Prof. Robinson Pizzio, Dr.

IFSC

Banca Examinadora:

Prof. Carlos Gontarski Speranza, Dr.

IFSC

Prof. Samir Bonho, Me.

IFSC

RESUMO

Este trabalho compara diversas técnicas de Recuperação de Informação para implementar um *software* que facilite o trabalho de analistas de suporte de Tecnologias de Informação. Tal *software* deve permitir o fácil acesso às ações remediativas utilizadas no passado, por meio da busca por chamados de suporte similares a um novo chamado recebido. A pesquisa foi executada na empresa Skaylink, a partir de uma base de dados da própria empresa onde são descritos chamados de suporte já finalizados. Foram comparadas onze técnicas de Recuperação de Informação, incluindo diversas abordagens e abrangendo desde técnicas clássicas até o estado da arte em pesquisa. Identificou-se que o melhor resultado foi obtido com a técnica *Sentence-BERT*, na sua variação multi-idioma *distiluse-base-multilingual-cased-v1*, onde 78,7% das recomendações realizadas pelo modelo foram consideradas relevantes. Além disso, este trabalho disponibilizou de forma gratuita e irrestrita o conjunto de dados utilizado, descreveu em detalhes a implementação de cada técnica e explorou as condições que afetam os resultados de cada modelo. Por fim, este estudo demonstrou a praticabilidade de um sistema de recuperação de chamados de suporte ao implementar um protótipo viável mínimo.

Palavras-chave: Recuperação de Informação. Aprendizado de Máquina. Processamento de Linguagem Natural. Chamados de suporte. Tecnologias de Informação.

ABSTRACT

This work compares several Information Retrieval techniques to implement a software that facilitates the work of Information Technology support analysts. Such software should allow easy access to corrective actions used in the past by searching for support tickets similar to a new incoming ticket. The research was carried out at the company Skylink, using a company's own database that describes support calls that have already been finalized. Eleven Information Retrieval techniques were compared, including several approaches and ranging from classic methods to state of the art research. The best results were obtained with the Sentence-BERT technique, in its multi-language variation distilluse-base-multilingual-cased-v1, where 78.7% of the recommendations made by the model were considered relevant. Furthermore, this work made the dataset used freely and completely available. It also described in detail the implementation of each technique and explored the conditions affecting the results of each model. Finally, this study demonstrated the practicality of a support ticket recovery system by implementing a minimal viable prototype.

Keywords: Information Retrieval. Machine Learning. Natural Language Processing. Support Tickets. Information Technologies.

LISTA DE FIGURAS

Figura 1 – Ilustração de uma rede <i>Multilayer-Perceptron</i>	19
Figura 2 – Mecanismo de atenção	20
Figura 3 – Arquitetura <i>Transformers</i>	21
Figura 4 – <i>Embeddings</i> gerados pela rede BERT para uma frase ilustrativa . .	23
Figura 5 – Arquitetura típica de um sistema de recuperação de informação . .	29
Figura 6 – Distribuição de idiomas dos chamados	31
Figura 7 – Arquitetura do protótipo	41
Figura 8 – Tela de <i>login</i> do protótipo	42
Figura 9 – Tela de visualização de chamados	42
Figura 10 – Tela de cadastro de chamados	42
Figura 11 – Tela de chamados recuperados pelo protótipo	43

LISTA DE QUADROS

Quadro 1 – Exemplos de similaridade entre palavras	33
Quadro 2 – Comparação das técnicas implementadas	35
Quadro 3 – Segmentação por lingua inglesa, conjunto de controle	37
Quadro 4 – Segmentação por lingua inglesa, conjunto sob teste	37
Quadro 5 – Segmentação por lingua portuguesa, conjunto de controle	38
Quadro 6 – Segmentação por lingua portuguesa, conjunto sob teste	38
Quadro 7 – Segmentação por categorias facilmente distinguíveis, conjunto de controle	39
Quadro 8 – Segmentação por categorias facilmente distinguíveis conjunto sob teste	39
Quadro 9 – Comparação das técnicas, utilizando a anotação por <i>clustering</i> . .	40

LISTA DE ABREVIATURAS E SIGLAS

IR	Recuperação de Informação
TF-IDF	<i>Term Frequency - Inverse Document Frequency</i>
LDA	<i>Latent Dirichlet Allocation</i>
BERT	<i>Bidirectional Encoder Representations from Transformers</i>
RDS	<i>Amazon Relational Databases</i>
EC2	<i>Amazon Elastic Compute Cloud</i>
EBS	<i>Amazon Elastic Block Store</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Justificativa	12
1.2	Definição do Problema	12
1.3	Objetivo Geral	12
1.4	Objetivos Específicos	12
1.5	Estrutura do Trabalho	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Recuperação da Informação	14
2.1.1	Modelo de espaço vetorial	14
2.1.2	Medidas de similaridade	15
2.1.3	Anotação de dados	16
2.1.4	Métricas de avaliação	16
2.2	Aprendizado de Máquina	17
2.2.1	Tipos de aprendizado	17
2.2.1.1	<i>Supervisionado</i>	17
2.2.1.2	<i>Não-supervisionado</i>	17
2.2.1.3	<i>Auto-supervisionado</i>	17
2.2.2	Redes Neurais	18
2.2.2.1	<i>Arquitetura Multilayer-Perceptron</i>	18
2.2.2.2	<i>Arquitetura Transformer</i>	19
2.2.2.3	<i>Large Language Models</i>	21
2.3	Técnicas de Recuperação de Informação	22
2.3.1	Sistemas especialistas	22
2.3.2	<i>Term Frequency - Inverse Document Frequency (TF-IDF)</i>	22
2.3.3	BM25	22
2.3.4	Latent Dirichlet Allocation (LDA)	22
2.3.5	Word2vec e derivados	23
2.3.6	<i>Bidirectional Encoder Representations from Transformers (BERT)</i> e derivados	23
2.4	Trabalhos relacionados	24
2.4.1	Recuperação de conteúdos diversos	24
2.4.2	Recuperação de chamados de suporte	25
2.4.3	Outras aplicações similares para chamados de suporte	26
3	METODOLOGIA	27
3.1	Métodos Aplicados	27
3.1.1	Identificação da melhor técnica	28
3.1.2	Avaliação do protótipo	29
4	APRESENTAÇÃO DOS RESULTADOS	30
4.1	Características do conjunto de dados	30
4.2	Implementação das técnicas	31
4.2.1	Sistema especialista	32
4.2.2	TF-IDF	32
4.2.3	BM25	32
4.2.4	LDA	32
4.2.5	Word2vec e derivados	33

4.2.6	BERT e derivados	33
4.3	Identificação da melhor técnica	34
4.4	Experimentos exploratórios	35
4.4.1	Segmentando os chamados por idioma	36
4.4.2	Segmentação por categorias facilmente distinguíveis	37
4.4.3	Anotações de relevância por <i>clustering</i>	39
4.5	Desenvolvimento do protótipo	40
4.6	Avaliação do protótipo	43
5	CONSIDERAÇÕES FINAIS	45
5.1	Sugestões para trabalhos futuros	46
	REFERÊNCIAS	47
	APÊNDICES	50
	APÊNDICE A – QUESTIONÁRIO APLICADO	51

1 INTRODUÇÃO

As Tecnologias de Informação (TI) já se tornaram parte do cotidiano das pessoas, as quais se mostram cada vez mais dependentes delas, seja para fins educacionais, sociais, econômicos ou profissionais (STAIR; REYNOLDS, 2009). De certa forma, espera-se que os recursos tecnológicos "simplesmente funcionem" e estejam disponíveis o tempo inteiro, de forma que indisponibilidades ou problemas nesses recursos tecnológicos acabam gerando disrupção na rotina.

Dentro do âmbito empresarial/institucional, as chamadas "equipes de suporte de TI" são as responsáveis pelo funcionamento adequado de recursos de TI (AL-HAWARI; BARHAM, 2021). Quando um usuário enfrenta algum problema com recursos de TI, ele descreve o seu problema abrindo um chamado de suporte em um sistema de gerenciamento. Então, um Analista de TI fica responsável por resolver o problema, comunicar-se com o usuário e registrar no sistema de gerenciamento quais ações foram realizadas para resolver o problema (ZHOU *et al.*, 2016).

O conhecimento acumulado nessas bases de dados é um ativo valioso para as empresas, pois pode ser usado para melhorar o uso de tecnologias digitais dentro da empresa. Porém, buscar essas informações nas bases de dados dos sistemas de gerenciamento é tecnicamente complicado e demorado (FIALHO, 2006). Além disso, Silva e Vasconcelos (2020) apontam que as equipes de suporte de TI também enfrentam problemas como a sobrecarga de chamados, rotatividade das equipes, uso de ferramentas tecnológicas inadequadas e/ou ultrapassadas, falta de mão de obra qualificada, entre outros, dificultando ainda mais a utilização dessas bases de dados.

Uma área de conhecimento que pode facilitar o uso dessas bases é a de Recuperação de Informação (RI), cujo objetivo é encontrar documentos de natureza não-estruturada (geralmente texto) que satisfaça uma necessidade de informação, a partir de uma grande coleção de materiais (geralmente armazenada em computadores) (MANNING; RAGHAVAN; SCHÜTZE, 2008). Tais técnicas podem permitir a busca por chamados de suporte na base de dados, facilitando ao analista de TI encontrar as informações necessárias para resolver um novo chamado (MUNI *et al.*, 2017).

Nesse contexto, esta pesquisa compara diversas técnicas de RI para implementar um *software* que facilite o trabalho de analistas de suporte. Tal *software* deve permitir o fácil acesso às ações remediativas aplicadas a chamados similares anteriores, o que Muni *et al.* (2017) apontam que pode economizar muito esforço do analista e, dessa forma, aprimorar consideravelmente o serviço prestado pelas equipes de TI. O projeto foi realizado na empresa Skylink, a partir de uma base de dados da própria empresa onde são descritos chamados de suportes já realizados e indicadas as soluções aplicadas a esses chamados.

1.1 Justificativa

O serviço de suporte de TI é fundamental para empresas que dependem de recursos de TI, e a velocidade de resolução de chamados impacta diretamente os usuários (ZAIDI *et al.*, 2021). Portanto, é importante que qualquer *software* utilizado na resolução de chamados seja devidamente desenvolvido e testado, de forma que o seu uso seja de real utilidade para os analistas.

Manning, Raghavan e Schütze (2008) analisam que a área de Recuperação de Informação "se desenvolveu como uma disciplina altamente empírica, e requer cuidadosa e bem pensada avaliação para demonstrar a performance superior de uma nova técnica para uma certa coleção de documentos" (MANNING; RAGHAVAN; SCHÜTZE, 2008, p. 139, tradução nossa). Dessa forma, a relevância deste trabalho está em viabilizar a implementação de um *software* que atenda, de forma satisfatória, o analista de suporte, por consequência, melhorando o serviço prestado aos usuários de TI.

1.2 Definição do Problema

Muitos chamados possuem resolução idêntica, de forma que basta ao analista identificar se um problema similar já foi resolvido no passado (MUNI *et al.*, 2017). Para isso, o analista pode perguntar a colegas, ler históricos de conversas, buscar diretamente na base de chamados já resolvidos, entre outros. Tal processo pode consumir muito tempo do analista, atrasando a resolução do problema e prejudicando o usuário.

A técnica mais adequada para facilitar esse processo é aquela que permita ao analista encontrar rapidamente informação que possa ser usada para resolver um novo chamado. Dessa maneira, a questão-problema que orienta este Trabalho de Conclusão de Curso é: qual a técnica mais adequada para implementar um sistema que busque por chamados de suporte passados, os quais sejam similares a um novo chamado de suporte recebido?

1.3 Objetivo Geral

Comparar técnicas de Recuperação de Informação para buscar por chamados de suporte similares a um novo chamado recebido, facilitando assim o trabalho de analistas de suporte.

1.4 Objetivos Específicos

Para alcançar o objetivo geral, foram traçados os seguintes objetivos específicos:

- a) entender as técnicas de Recuperação de Informação aplicáveis ao domínio do problema;

- b) escolher e implementar técnicas para resolver o problema;
- c) definir a metodologia para comparar diferentes soluções para o problema;
- d) utilizar a base de dados da Skaylink para avaliar as diferentes soluções para o problema;
- e) implementar um *software* protótipo com a melhor técnica;
- f) testar o protótipo em condições reais e avaliar se esse auxilia na resolução de chamados de suporte.

1.5 Estrutura do Trabalho

Este trabalho está estruturado em cinco capítulos, sendo o primeiro deles o da introdução, onde é apresentado o tema da pesquisa e os principais conceitos associados ao domínio do problema, bem como os objetivos e a relevância do trabalho. Em seguida, inicia-se a Fundamentação Teórica, necessária para a escolha e implementação das técnicas utilizadas durante o desenvolvimento. No capítulo 3, é exposta a metodologia utilizada, a qual precede o capítulo dos resultados obtidos e, por fim, são apresentadas as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

Para a melhor compreensão desta pesquisa, o capítulo da Fundamentação Teórica foi dividido em tópicos e subtópicos referentes à área de conhecimento em que o estudo se insere, as técnicas utilizadas, e às publicações utilizadas como referência para a implementação prática do sistema.

2.1 Recuperação da Informação

A Recuperação da Informação (RI) é uma área interdisciplinar que tem por objetivos "encontrar materiais (geralmente documentos) de natureza não-estruturada (geralmente texto) que satisfaça uma necessidade de informação a partir de uma larga coleção de documentos (geralmente armazenada em computadores)" (MANNING; RAGHAVAN; SCHÜTZE, 2008, pg. 1, tradução nossa)

Singhal (2001) aponta que os primeiros sistemas eram baseados em consultados booleanas, onde o usuário expressava sua necessidade de informação por meio de complexas combinações de palavras com operadores lógicos AND, OR e NOT. Esses sistemas evoluíram para lidar com consultas em linguagem natural, onde o usuário brevemente descreve a informação que está buscando, ao ponto de se tornarem a principal forma de acessar informação (principalmente na internet) (MANNING; RAGHAVAN; SCHÜTZE, 2008).

Além da área de RI propriamente dita, existem diversos outros ramos de estudo que convergem para objetivos similares: Sistemas de Recomendação Baseados em Conteúdo (que têm por objetivo recomendar materiais baseado nas características do material e das preferências do usuário) (RICCI *et al.*, 2011), *Semantic Text Similarity* (que objetiva determinar o grau de similaridade entre duas frases (REIMERS; GUREVYCH, 2019)), entre outros. Ao longo deste trabalho, utiliza-se a nomenclatura usual da área de RI, com menções a outras áreas quando necessário.

Por ser uma área que se estuda desde a década de 50, há diversos modelos para implementar sistemas de RI, sendo os principais: modelo de espaço vetorial, modelo probabilístico e modelo de inferência em rede (SINGHAL, 2001).

2.1.1 Modelo de espaço vetorial

O modelo de espaço vetorial representa documentos como vetores em um espaço vetorial comum (MANNING; RAGHAVAN; SCHÜTZE, 2008). Tradicionalmente, palavras são utilizadas como dimensões independentes desse espaço, de forma que documentos são representados por vetores esparsos com muitas dimensões (SINGHAL, 2001). Entretanto, técnicas mais recentes utilizam representações densas, também

chamadas de *word embedding*, cujas dimensão desse espaço vetorial não possuem uma interpretação clara (MIKOLOV *et al.*, 2013).

Segundo Mulsa e Spanakis (2020), o vector que representa cada palavra pode ser calculado apenas a partir da palavra em si (sendo, portanto, não-contextual), ou ser calculado considerando as demais palavras que a circundam no documento em análise (as chamadas representações contextuais). As técnicas do segundo tipo costumam levar a resultados melhores, sendo amplamente usadas em aplicações reais.

Quanto ao idioma, as representações podem ser uni-idioma ou multi-idioma. No segundo caso, os *word embedding* da mesma palavra em diversos idiomas são todos alinhados, ou seja, significados similares são mapeados próximos no espaço vetorial (REIMERS; GUREVYCH, 2020).

O modelo de espaço vetorial faz parte de um conjunto de métodos conhecidos como representacionais (também conhecidos como *latent semantic models*, ou ainda *distributional semantic models*), em oposição a métodos interativos (também conhecidos como *cross-encoding models*) (GUO *et al.*, 2020). Enquanto no primeiro é gerado um vetor, o segundo compara textos um a um. Métodos interativos implicam claramente em uma dificuldade de operacionalizar o sistema, pois o número de comparações que devem ser realizadas cresce de forma fatorial com o número de documentos a serem comparados.

2.1.2 Medidas de similaridade

Por se tratarem de vetores, as representações podem ser comparadas com medidas tradicionais de distância ou de similaridade. A premissa fundamental dessa comparação é a de que a proximidade de dois pontos nesse espaço está relacionada com a similaridade semântica dos dois documentos (MANNING; RAGHAVAN; SCHÜTZE, 2008) e, portanto, a tarefa de recuperar informação se resume a buscar pontos próximos no espaço vetorial formado pelos *embeddings*.

A métrica mais usual para comparar *embeddings* é a Similaridade do Cosseno, Equação 1, que possui a desejável propriedade de apenas produzir valores entre 0 e 1 (MITKOV, 2003).

$$\text{Similaridade do Cosseno}(A, B) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \quad (1)$$

Quando as dimensões do vetor não são ponderadas (isto é, são apenas valores 0 ou 1, verdadeiro ou falso), é comum utilizar a métrica de Similaridade de Jaccard (MITKOV, 2003) Dados dois conjuntos, A e B, essa métrica é calculada pela Equação 2.

$$\text{Similaridade de Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

2.1.3 Anotação de dados

Elaborar conjuntos de dados apropriados para tarefas de RI é desafiador, tanto pela subjetividade de julgar a relevância de documentos quanto pela dificuldade de o pesquisador se lembrar de todos os documentos de uma coleção que poderiam ser relevantes para uma certa necessidade de informação.

Manning, Raghavan e Schütze (2008) sugerem a solução de *pooling*: analisar os possíveis documentos relevantes apenas em um pequeno subconjunto da coleção inteira. Tal conjunto deve ser obtido por meio de algum método predefinido, como a utilização de um sistema de RI já existente, e que se deseja aprimorar.

Outra possibilidade é anotar apenas pares de sentenças (REIMERS; GUREVYCH, 2019). Nessa técnica, o conjunto de dados consiste apenas de documentos tomados dois a dois, e de um julgamento do pesquisador de quão similares os dois documentos são (ou, de forma análoga, quão relevante um documento é para o outro).

Independente das técnicas utilizadas para facilitar a compilação de um conjunto de dados, Manning, Raghavan e Schütze (2008) apontam que "50 necessidades de informação têm sido usualmente consideradas suficientes" (MANNING; RAGHAVAN; SCHÜTZE, 2008, pg. 140, tradução nossa).

2.1.4 Métricas de avaliação

De posse de um conjunto de dados anotados, é possível avaliar a qualidade de um sistema de RI utilizando métricas padronizadas. Tais métricas se dividem em dois grupos principais: não-ranqueadas e ranqueadas.

Métricas não-ranqueadas não levam em consideração a ordem das recomendações (BAI *et al.*, 2019). As mais utilizadas são Precisão (Equação 3) e Revocação (Equação 4)

$$\text{Precisão} = \frac{\text{Nº de itens relevantes recomendados}}{\text{Nº total de itens recomendados}} \quad (3)$$

$$\text{Revocação} = \frac{\text{Nº de itens relevantes recomendados}}{\text{Nº total de itens relevantes}} \quad (4)$$

Já as métricas ranqueadas consideram a ordenação dos resultados (BAI *et al.*, 2019), de forma que um documento muito relevante deve aparecer primeiro nos

resultados, enquanto documentos parcialmente relevantes devem aparecer por último. Tais métricas são mais complexas de serem implementadas e interpretadas, e não serão mais abordadas no decorrer deste trabalho.

2.2 Aprendizado de Máquina

As técnicas de Aprendizado de Máquina permitem construir um modelo para previsão de uma saída baseada em uma ou mais entradas utilizando dados históricos, que representem instâncias do problema (FACELI *et al.*, 2011).

2.2.1 Tipos de aprendizado

As técnicas de Aprendizado de Máquina podem ser divididas em diversas categorias, sendo que as principais delas são Aprendizado Supervisionado, Aprendizado Não-supervisionado e Aprendizado Auto-supervisionado (CHOLLET, 2017).

2.2.1.1 Supervisionado

O aprendizado supervisionado consiste em mapear dados de entrada para uma saída conhecida, a partir de um conjunto de exemplos elaborados por humanos (CHOLLET, 2017). Essa saída pode ser um valor contínuo (nesse caso, o problema é chamado de regressão) ou categórico (os chamados problemas de classificação) (RUSSELL; NORVIG, 2010).

2.2.1.2 Não-supervisionado

No aprendizado não-supervisionado não há um conjunto de exemplo elaborado por humanos (FACELI *et al.*, 2011). Ou seja, não há um atributo-alvo a ser mapeado, e o objetivo é identificar padrões nos dados. Essas técnicas não requerem grande conhecimento prévio sobre o problema, e permitem inferir padrões naturais (livres de tendências do pesquisador). Dentre os vários tipos de problemas não-supervisionados, os mais utilizados são os de *clusterização* e de redução de dimensionalidade (FACELI *et al.*, 2011).

2.2.1.3 Auto-supervisionado

O aprendizado auto-supervisionado não possui exemplos elaborados por humanos, mas possui um atributo alvo (CHOLLET, 2017). Ou seja, pode-se pensar como sendo um aprendizado supervisionado, porém sem seres humanos. O atributo alvo é gerado pelos próprios dados, usualmente utilizando algum algoritmo heurístico, por exemplo: prever o próximo *frame* de um vídeo, a próxima palavra de um texto, entre outros (CHOLLET, 2017).

2.2.2 Redes Neurais

Redes Neurais Artificiais (RNA) são um conjunto de modelos computacionais que tomam como inspiração a estrutura e o funcionamento do sistema nervoso (FACELI *et al.*, 2011). Os primeiros estudos na área datam de 1943, com McCulloch e Pitts, que propuseram um modelo matemático de um neurônio (FACELI *et al.*, 2011). Chollet (2017) aponta que modelos atuais de RNAs são muito diferentes das primeiras redes neurais, com uma grande variedade de tipos de neurônios, tipos de interconexões entre neurônios, algoritmos de aprendizado, entre outros.

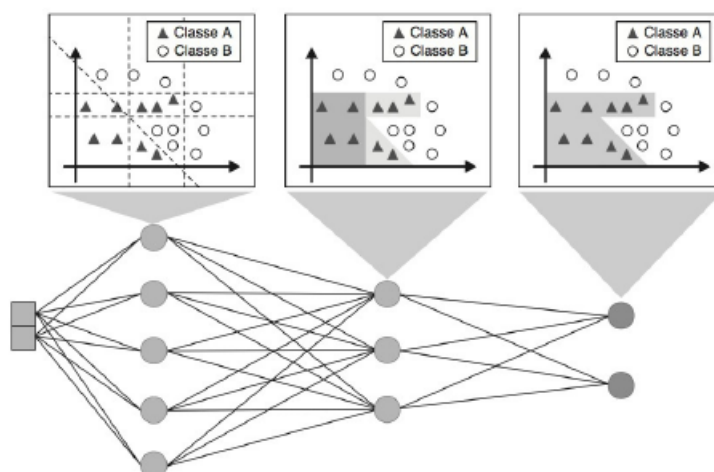
2.2.2.1 Arquitetura Multilayer-Perceptron

Uma das possíveis arquiteturas é a chamada *Multilayer-Perceptron*. Tal rede é composta por diversos "neurônios artificiais" interconectados, sendo que cada neurônio implementa a função da Equação 5, onde y é a saída do neurônios, f uma função não linear (chamada usualmente de função de ativação), w_0 o viés, \mathbf{X} o vetor de entrada, e \mathbf{W} o vetor de pesos da rede neural (FACELI *et al.*, 2011). O processo de ajustar os valores de \mathbf{W} para obter o comportamento desejado é chamado de treinamento.

$$y = f(w_0 + \mathbf{X}^T \mathbf{W}) \quad (5)$$

As saídas dos neurônios formam as entradas de outros neurônios, que estão organizados em camadas, e as camadas intermediárias são chamadas de camadas ocultas (FACELI *et al.*, 2011). A Figura 1 ilustra uma *Multilayer-Perceptron* com duas entradas, duas camadas ocultas, e dois neurônios de saída, treinada para diferenciar as amostras do tipo A (representadas por triângulos) das amostras do tipo B (representadas por círculos), e que a cada camada adquire uma melhor capacidade de diferenciação.

Russell e Norvig (2010) apontam que "com uma única, suficientemente grande camada oculta, é possível representar qualquer função contínua da entrada com precisão arbitrária; com duas camadas ocultas, até funções descontínuas podem ser representadas" (RUSSELL; NORVIG, 2010, p. 732, tradução nossa).

Figura 1 – Ilustração de uma rede *Multilayer-Perceptron*

Fonte: Faceli *et al.* (2011).

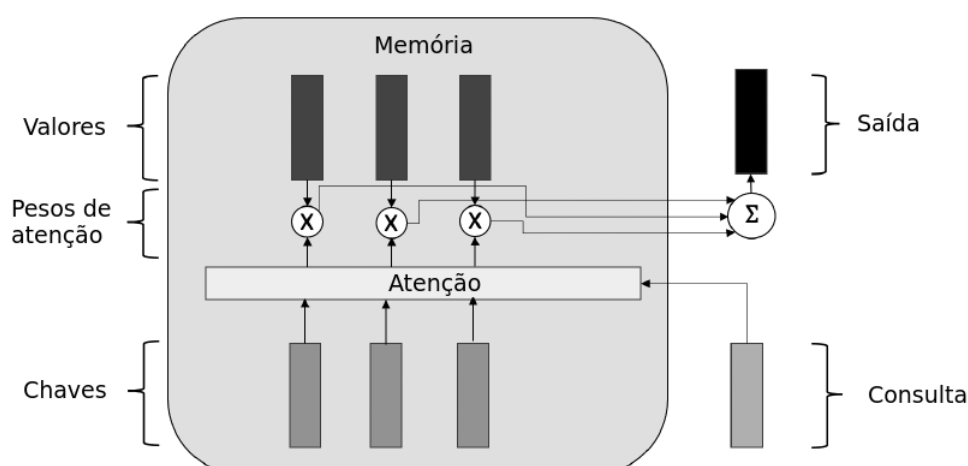
2.2.2.2 Arquitetura Transformer

Outra arquitetura muito utilizada é a chamada *Transformer*, que difere da *Multilayer-Perceptron* devido a duas ideias principais: uma organização de codificadora-decodificadora e um mecanismo de Auto-Atenção. Dessa forma, a rede *Transformer* consegue lidar de forma efetiva com entradas e saídas de tamanho variável, como, por exemplo, quando se deseja traduzir uma frase em português para uma frase em inglês (VASWANI *et al.*, 2017).

As redes neurais organizadas na forma de codificadora-decodificadora possuem as suas camadas claramente divididas entre uma porção codificadora (que converte a entrada de tamanho variável em um vetor de tamanho fixo) e uma porção decodificadora (que converte o vetor de tamanho fixo em uma saída de tamanho variável) (KAMATH; GRAHAM; EMARA, 2022).

O mecanismo de atenção funciona como um pequeno sistema de recuperação de informação de curto prazo, armazenando as informações vistas até agora no texto colocado na entrada da rede neural. Dada uma Consulta (uma palavra sobre a qual a rede precisa de mais informações, por exemplo), o mecanismo procura por Valores (palavras passadas armazenadas), que são representados por seus vetores de Chave, retornando uma saída que contém a informação contextual desejada (Figura 2) (KAMATH; GRAHAM; EMARA, 2022). Uma variação desse mecanismo, chamada de Auto-Atenção, elabora as Consultas, Valores e Chaves a partir do próprio texto de entrada, aprendendo a gerar suas representações automaticamente (KAMATH; GRAHAM; EMARA, 2022). Nessa variação, os vetores Consulta, Valor e Chave são obtidos por meio da multiplicação dos vetores *embeddings* de entrada com matrizes W_Q , W_V , e W_K , treinadas de forma conjunta com a rede *Transformer* de forma análoga aos neurônios da rede *perceptron* (KAMATH; GRAHAM; EMARA, 2022).

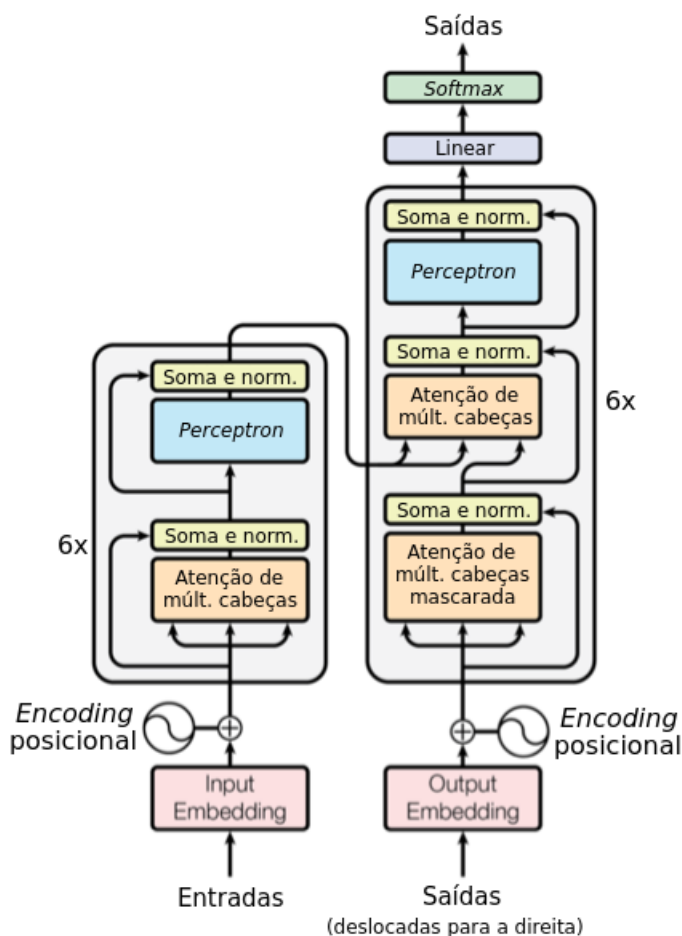
Figura 2 – Mecanismo de atenção



Fonte: Adaptado de Kamath, Graham e Emara (2022).

A arquitetura *Transformer* utiliza 6 codificadores e 6 decodificadores em paralelo (Figura 3), cada um com os seus mecanismos próprios de Auto-Atenção (VASWANI *et al.*, 2017). Além disso, Kamath, Graham e Emara (2022) apontam que a arquitetura utiliza diversos "truques" para aumentar a sua capacidade de aprendizado, entre eles: o mecanismo de Auto-Atenção é repetido 8 vezes para cada sequência, sendo assim chamado de Auto-Atenção de Múltiplas Cabeças; as Conexões Residuais formam um caminho alternativo para cada vetor, "curto-circuitando" as representações e facilitando o treinamento; as camadas de normalização garantem que cada vetor possui média zero e variância um, com o mesmo objetivo; camadas *Perceptron* são intercaladas com o mecanismo de Auto-Atenção; a camada linear expande as representações vetoriais densas para o número de palavras do dicionário (as palavras da língua portuguesa, por exemplo); a camada *softmax* gera uma distribuição probabilística (qual é a próxima palavra mais provável, na sequência de saída); por fim, as próprias saídas produzidas a cada iteração são colocadas novamente na entrada do decodificador, gerando uma recursão.

Figura 3 – Arquitetura Transformers



Fonte: Adaptado de Vaswani *et al.* (2017).

2.2.2.3 Large Language Models

A arquitetura *Transformer* é mais rápida do que arquiteturas utilizadas anteriormente, facilitando o treinamento em conjuntos de dados maiores (KAMATH; GRAHAM; EMARA, 2022). Tal inovação permitiu o treinamento de modelos utilizando corpos de texto gigantescos, como, por exemplo, todos os livros já escritos ou quase todo o conteúdo público da internet (KAMATH; GRAHAM; EMARA, 2022). Para isso, é necessário utilizar técnicas de aprendizado auto-supervisionado, pois esses conjuntos não possuem um atributo de saída definido por humanos.

Os chamados *Large Language Models* são redes neurais treinadas de forma auto-supervisionada em grandes conjuntos de dados (KAMATH; GRAHAM; EMARA, 2022). Uma vez treinados, esses modelos "aprendem o idioma" e podem ser reutilizados com muita facilidade, adicionando uma nova camada de neurônios a sua saída e brevemente treinando apenas a camada adicionada (DEVLIN *et al.*, 2019).

2.3 Técnicas de Recuperação de Informação

Manning, Raghavan e Schütze (2008) apontam que diversas técnicas podem ser utilizadas para implementar sistemas de RI, cada uma com os seus pontos fortes e pontos fracos.

2.3.1 Sistemas especialistas

Os sistemas especialistas são construídos com a ajuda de entendedores do domínio do problema, desenvolvendo programas que incorporam o seu conhecimento (LIU; GEGOV; COCEA, 2016). O programa pode ser total ou parcialmente aprendido a partir dos dados, por meio da indução de regras ou, então, definido manualmente pelo especialista (LIU; GEGOV; COCEA, 2016).

2.3.2 Term Frequency - Inverse Document Frequency (TF-IDF)

A técnica de TF-IDF permite gerar uma representação vetorial para um documento de texto, onde cada posição de vetor corresponde a uma palavra do conjunto de palavras conhecidas. O valor de cada termo t no documento d é calculado considerando quantas vezes t aparece em d , ponderado pelo número de outros documentos nos quais t também aparece, de forma que palavras muito frequentes automaticamente recebem um peso menor. Formalmente, o peso de cada posição do vetor é dado pela Equação 6, onde $tf_{(t,f)}$ é o número de ocorrência de t em d , N é o número de documentos na coleção e $df_{(t)}$ é o número de documentos em que o termo t aparece (MITKOV, 2003).

$$\text{TF-IDF}_{(t,d)} = tf_{(t,f)} * \log_{10} \left(\frac{N}{df_{(t)}} \right) \quad (6)$$

2.3.3 BM25

A técnicas de *Okapi BM25 Weighting Scheme*, usualmente conhecida apenas por BM25, utiliza métodos probabilísticos para ordenar os documentos a serem recuperados, a partir dos valores de *Term Frequency* e *Inverse Document Frequency*. Tal como a técnica original de TF-IDF, a BM25 não considera a ordenação das palavras na frase, apenas a sua presença no documento (MANNING; RAGHAVAN; SCHÜTZE, 2008).

2.3.4 Latent Dirichlet Allocation (LDA)

A técnica de LDA também possui origem probabilística, mas não utiliza a técnica de TF-IDF. Ao invés disso, LDA pode ser entendido como uma forma de *clusterização* não-exclusiva (quando uma determinada amostra pode pertencer a mais de um *cluster*, com diferentes graus de pertencimento a cada um), onde cada *cluster*

representa um tópico/assunto que foi identificado na coleção de documentos. O vetor formado pelo grau de pertencimento de um documento para com cada *cluster* pode ser utilizado para fins de recuperação de informação, de maneira similar ao vetor de TF-IDF (MANNING; RAGHAVAN; SCHÜTZE, 2008).

2.3.5 Word2vec e derivados

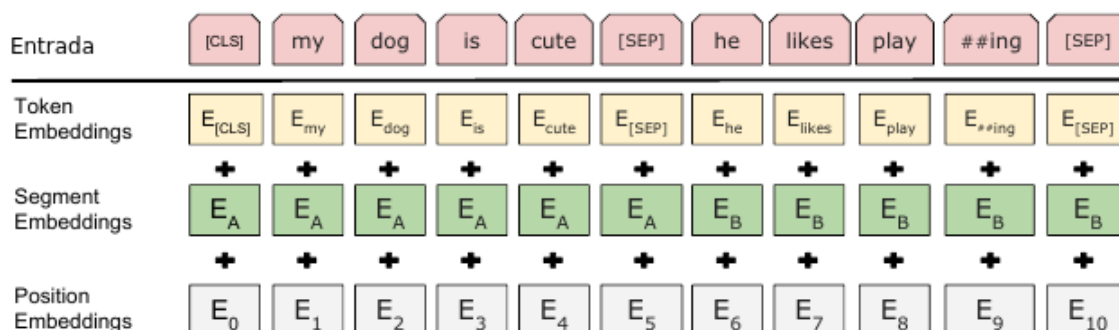
Word2vec é uma técnica baseada em redes neurais, que tem por objetivo aprender de forma não-supervisionada um *embedding* para cada palavra. A tarefa utilizada para treinar a rede consiste em prever a próxima palavra em uma frase, e a rede neural consiste de apenas uma camada *perceptron* (MIKOLOV *et al.*, 2013).

Uma das variações dessa rede é chamada de Doc2vec, que leva em consideração o documento inteiro ao calcular o *embedding* de uma palavra (DELUCIA; MOORE, 2020). Dessa forma, Doc2vec passa a ser considerada uma representação contextual, enquanto word2vec é não-contextual.

2.3.6 Bidirectional Encoder Representations from Transformers (BERT) e derivados

A rede neural BERT utiliza a arquitetura transformer, porém com modificações que a tornam mais fácil de ser adaptada para diversas tarefas após sua etapa inicial de treinamento não-supervisionado. BERT utiliza apenas a parte codificadora da rede transformer, gerando 3 vetores para cada palavra do texto (e adicionalmente 3 vetores para o texto como um todo): *token embedding* (referente apenas à palavra, portanto não-contextual), *segment embedding* (referente à função da palavra na frase, portanto contextual) e *position embedding* (referência apenas a posição da palavra na frase) (KAMATH; GRAHAM; EMARA, 2022), conforme ilustrado no exemplo da Figura 4.

Figura 4 – Embeddings gerados pela rede BERT para uma frase ilustrativa



Fonte: Adaptado de Devlin *et al.* (2019)

A versão mais utilizada da rede BERT, chamada de BERT-Base, utiliza 12 conjuntos de codificadores *Transformer*, cujas saídas são um vetor de tamanho fixo com 768 posições, e cada um possuindo 12 cabeças de Auto-Atenção. A etapa não-

supervisionada do treinamento é utilizada combinando duas tarefas: prever uma palavra que foi ocultada da frase (de forma similar à word2vec) e prever a próxima frase de um texto. Dessa forma, a rede BERT aprende a trabalhar com texto de forma muito genérica, e pode ser facilmente adaptada para outras tarefas (VASWANI *et al.*, 2017).

Objetivando melhorar a qualidade dos resultados obtidos para áreas específicas, diversas variações da BERT foram desenvolvidas (KAMATH; GRAHAM; EMARA, 2022). Uma delas foi apresentada por Reimers e Gurevych (2019), chamada *Sentence-BERT*, que melhora a geração de *embeddings* para tarefas como RI.

2.4 Trabalhos relacionados

Diversas publicações já apresentaram sistemas similares ao proposto no presente trabalho. Nesta sessão, alguns desses trabalhos serão discutidos, realçando as suas similaridades e peculiaridades.

2.4.1 Recuperação de conteúdos diversos

De forma geral, há um longo histórico de aplicações de RI para recomendar diversos tipos de documentos à usuários que estão utilizando um sistema de informação. Cezar *et al.* (2021) apresentam um sistema de recomendação de artigos científicos, utilizando a técnica de TF-IDF para vetorizar o resumo dos artigos e manualmente perguntando ao usuário quais são as suas área/termos de interesse. O trabalho utilizou os artigos do Simpósio Brasileiro de Sistemas de Informação, vetorizou e comparou similaridades utilizando o software *ElasticSearch*, e apresentou as recomendações para o usuário por meio de uma página web.

O trabalho de Al-Ghossein *et al.* (2018) utiliza a técnica de *Latent Dirichlet Allocation* para complementar um sistema de recomendação baseado em *Collaborative Filtering*. O sistema proposto foi testado para recomendação de notícias e filmes, mas pode ser aplicado de forma similar a outros tipos de conteúdos digitais.

O modelo de espaço vetorial também é utilizado em buscadores web, como apresentado na publicação de Yin *et al.* (2016) do buscador Yahoo. Nesse trabalho, foram utilizados *embeddings* contextuais para relacionar a semântica de consultas frequentes (para as quais é mais fácil aplicar técnicas tradicionais de recomendação) com consultas raras (para as quais é preciso levar em conta o conteúdo das páginas, pois há poucos dados de *clicks* reais de usuários). Esse mecanismo de enriquecimento semântico das consultas, em conjunto com outras técnicas apresentadas no mesmo artigo, foram avaliadas tanto em conjuntos de dados predefinidos quanto com a utilização do buscador por usuários reais.

2.4.2 Recuperação de chamados de suporte

Há diversos trabalhos recentes utilizando técnicas de RI no domínio de chamados de suporte de TI, recorrendo a uma variedade grande de metodologias e técnicas.

O trabalho de Muni *et al.* (2017) utilizou o texto da descrição do chamado pré-processado com técnicas de *lemmatization* e remoção de *stop words*, então aplicou TF-IDF e técnicas de redução de dimensionalidade para obter uma representação vetorial do chamado e, por fim, comparou os vetores com Similaridade do Cosseno. A anotação dos dados para treinamento foi realizada com um sistema já existente, e um analista verificou a qualidade das recomendações no final do processo.

Em Feng, Senapati e Liu (2022), utilizaram-se *embeddings* da rede BERT e derivados (especificamente: RoBERTa, DistilBERT e DistilRoBERTa) para permitir a busca semântica de chamados. Adicionalmente, utiliza modelos supervisionados para classificar o grupo/setor da empresa que deve se responsabilizar pelo chamado e, também, o analista que deve atender ao chamado.

O Centro de Pesquisa em Sistemas de Alta Performance do *Los Alamos National Laboratory* publicou no trabalho de DeLucia e Moore (2020) os seus métodos para automaticamente classificar chamados e sugerir chamados similares. Foram utilizados 70 mil chamados, cujo texto foi pré-processado (remoção de *stop words*, conversão para *lower case*, entre outros) e, então, vetorizado (utilizando 3 técnicas diferentes: *Latent Dirichlet Allocation*, *Latent Semantic Analysis*, e *Doc2Vec*). O sistema foi inicialmente avaliado por meio da comparação com dois sistemas existentes (a funcionalidade "*more like this*" do *software ElasticSearch* e um sistema especialista que compara o percentual de palavras em comum); assim, 200 chamados foram utilizados para serem manualmente avaliados de forma qualitativa por um especialista.

Quando não há um sistema anterior para utilizar como comparativo (como os dois trabalhos anteriores, de Muni *et al.* (2017) e DeLucia e Moore (2020)), é comum utilizar um conjunto pequeno de avaliação. Em Dyrhovden, Norvang e Sund (2021) utilizaram-se apenas 5 chamados, porém as recomendações foram avaliadas em duas dimensões diferentes: se as recomendações pertenciam a uma área/categoria similar (exemplos de categorias desse trabalho são "edição de vídeo" e "interface gráfica") e se as recomendações possuíam a mesma característica funcional (nesse trabalho utilizou-se funções como "iniciar", "salvar", entre outros). A partir dessa metodologia de avaliação, foram comparadas 4 técnicas de vetorização (Word2Vec, Doc2Vec, TF-IDF, e BERT, utilizando apenas modelos não retreinados), em que Doc2Vec e BERT obtiveram os melhores resultados.

2.4.3 Outras aplicações similares para chamados de suporte

Para além da recuperação de chamados similares, há diversas aplicações de RI, Aprendizado de Máquina e área similares para facilitar a resolução de chamados de suporte.

No trabalho de Al-Hawari e Barham (2021), 1585 chamados foram utilizados para treinar um modelo que classificava o chamado em uma de 13 categorias. Utilizou-se como entrada a concatenação dos campos de título, descrição de comentários; além disso, foi usado o pré-processamento com *steaming* e TF-IDF para representar os chamados, e 4 modelos foram testados: um sistema baseado em regras, J48, *Naive Bayes* e SMO.

Wahba, Madhavji e Steinbacher (2020) também testou diversas técnicas para classificar chamados de suporte, porém focando nas técnicas de vetorização ao invés das técnicas de classificação. Testaram-se TF-IDF e word2vec treinado em 3 conjuntos de dados diferentes (*Google News*, um conjunto de dados sobre engenharia de software, e o próprio conjunto de chamados utilizados para a classificação). A base de chamados utilizada possuía 1,6 milhões de chamados, distribuídos em 32 categorias.

Diversas empresas também desenvolveram soluções para seus sistemas internos de gerenciamento de chamados, como a Uber (MOLINO; ZHENG; WANG, 2018). Em Molino, Zheng e Wang (2018), é apresentado um sistema que realiza a classificação de chamados para definir *templates* de repostas que o analista pode utilizar. Os modelos resultantes foram avaliados com usuários reais e levaram a uma diminuição de 10% no tempo de resolução do chamado.

3 METODOLOGIA

Este trabalho possui natureza aplicada que, segundo Severino (2016), busca gerar de conhecimento para utilização prática e imediata a um problema específico. A aplicação prática que orienta esta pesquisa é facilitar o trabalho do analista de suporte, tomando como base um conjunto de dados da empresa Skaylink. Tal conjunto é composto por informações de 20356 chamados de suporte realizados entre os anos de 2017 e 2022, registrados durante a prestação de serviços para uma empresa cliente da Skaylink. Os dados foram anonimizados antes da realização deste trabalho, de forma que nenhuma informação pessoal estivesse presente.

Quanto aos objetivos, esta pesquisa pode ser considerada exploratória. Gil (2008) aponta que esse tipo de pesquisa levanta informações sobre um determinado assunto, mapeando as condições de manifestação desse assunto. Como há muitas técnicas disponíveis para facilitar o trabalho do analista, esta pesquisa mapeia e compara as diversas possibilidades, uma etapa fundamental para o prosseguimento do trabalho.

Para tanto, utilizou-se uma abordagem quali-quantitativa, que utiliza tanto a mensuração quantitativa de parâmetros associados ao assunto quanto da interpretação do pesquisador na análise dos resultados (SEVERINO, 2016). Serão definidas métricas objetivas para a comparação estatística das técnicas, bem como a análise da experiência subjetiva dos usuários do protótipo desenvolvido.

Realizou-se também uma pesquisa bibliográfica para a construção do referencial teórico e para a implementação prática das técnicas. Tal tipo de pesquisa é realizada a partir de materiais disponíveis de pesquisas anteriores, como livros e artigos de pesquisadores devidamente registrados (GIL, 2008).

3.1 Métodos Aplicados

A partir do estudo das técnicas de RI disponíveis para atender aos objetivos, escolheu-se comparar diversas técnicas que são representativas das principais abordagens existentes para o problema:

- a) representando abordagens tradicionais que são altamente dependentes dos conhecimentos do desenvolvedor sobre o domínio, desenvolveu-se um Sistema Especialista;
- b) representando abordagens estatísticas bem consolidadas na área de RI, utilizou-se TF-IDF;
- c) representando abordagens probabilísticas bem consolidadas na área de RI, utilizou-se BM25 e LDA;

- d) representando abordagens de redes neurais com *embeddings* não contextuais, utilizou-se *Word2vec* treinado em uma base de idioma inglês, *Word2vec* inteiramente treinado com a base de dados da Skaylink, e a sua variação contextual *Doc2vec* também treinado com a base de dados da Skaylink;
- e) representando abordagens de redes neurais com *embeddings* contextuais, utilizou-se BERT treinado com dados multi-idioma, *Sentence-BERT* treinado com dados multi-idioma, *Sentence-BERT* treinado com dados apenas da língua inglesa e *Sentence-BERT* inicialmente treinado com dados multi-idioma e então retreinado com a base de dados da Skaylink.

Adicionalmente, implementou-se a técnica de seleção aleatória - na qual os chamados são selecionados aleatoriamente - para facilitar a interpretação dos valores obtidos. A comparação das cinco técnicas se deu em duas etapas sequenciais: identificação da melhor técnica e avaliação do protótipo.

3.1.1 Identificação da melhor técnica

O processo de identificar os chamados similares também é chamado de rotulagem de dados (FACELI *et al.*, 2011); neste trabalho, foram rotulados 300 chamados, escolhidos para serem representativos do conjunto completo de dados (selecioneando manualmente 30 chamados de cada uma das 10 categorias mais frequentes do conjunto). Dividiram-se os 300 chamados em 3 subgrupos de 100 chamados, para facilitar a rotulagem. Para cada chamado, indicou-se manualmente quais eram os outros 5 chamados mais parecidos com ele (dentro o subgrupo).

Para simplificar rotulagem, utilizou-se a ferramenta gráfica Miro, colocando-se o texto de cada chamado em cartão disposto em um plano bidimensional, de forma que chamados parecidos sejam posicionados próximos. Realizou-se o procedimento supracitado com 3 analistas rotulando os dados de forma independente, cada um responsável por um subgrupo.

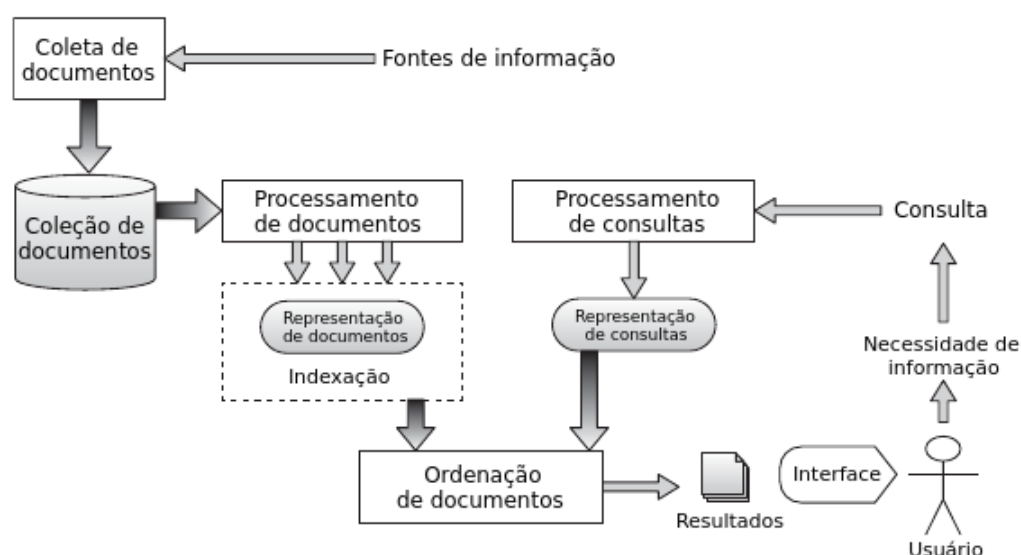
Então, o modelo em avaliação deu 5 recomendações para cada um dos 100 chamados de cada subconjunto, e avaliou-se utilizando a métrica de precisão. Para medir a precisão, considerou-se como "documento relevante" os 5 chamados manualmente indicados durante a rotulagem, isto é, os 5 chamados mais próximos no plano bidimensional.

Depois disso, calculou-se a média da precisão entre os 3 conjuntos rotulados, e a melhor técnica foi escolhida como sendo a com maior precisão média.

3.1.2 Avaliação do protótipo

Utilizando a técnica identificada anteriormente como sendo a mais adequada, implementou-se um *software* protótipo. Esse foi alimentado inicialmente com os 20356 chamados da base de dados atual, e permite o registro de novos chamados de suporte. Quando um chamado é criado, o protótipo gera 5 recomendações de chamados anteriores similares ao novo chamado aberto. O novo chamado é armazenado pelo protótipo, para ser utilizado como base histórica para recomendações futuras. O protótipo disponibiliza um botão para permitir ao analista dar o seu *feedback*, indicando se as recomendações lhe foram úteis ou não. Ou seja, o protótipo possui os principais elementos da arquitetura de referência apresentada por Mitkov (2003) (Figura 5).

Figura 5 – Arquitetura típica de um sistema de recuperação de informação



Fonte: Adaptado de Mitkov (2003).

Utilizou-se o protótipo no trabalho diário da Skylink. Os *feedbacks* coletados foram armazenados de forma anônima e utilizados para avaliar se a técnica escolhida de fato é capaz de auxiliar o analista na resolução de incidentes.

Por fim, disponibilizou-se um questionário online e anônimo para os analistas, conforme detalha o Apêndice A. As respostas foram analisadas para entender se o comportamento do protótipo foi satisfatório quando utilizado em condições reais, bem como para permitir o aprimoramento do sistema em trabalhos futuros.

4 APRESENTAÇÃO DOS RESULTADOS

Neste capítulo serão discutidos os resultados obtidos ao longo desta pesquisa, especificamente: a análise da base de dados a ser utilizada no trabalho, a utilização de técnicas de RI para recomendar chamados similares nessa base, e o protótipo desenvolvido para validar o sistema proposto.

4.1 Características do conjunto de dados

Cada chamado é descrito por nove variáveis: *external_ID* (identificação do chamado no sistema de gerenciamento), *title* (título do chamado, conforme informado pelo usuário que abriu o incidente), *description* (descrição do chamado, também informado pelo usuário), *category* (categoria do incidente), *date_open* (data de abertura do chamado), *date_close* (data de finalização do chamado), *location* (escritório do qual o usuário faz parte), *solution* (solução que foi aplicada ao chamado) e *analysts* (grupo de analistas responsáveis pela resolução do chamado). Ao receber o chamado, o analista não recebe as variáveis *category*, *date_close* e *solution*, que são adicionadas aos registros apenas após a finalização do chamado.

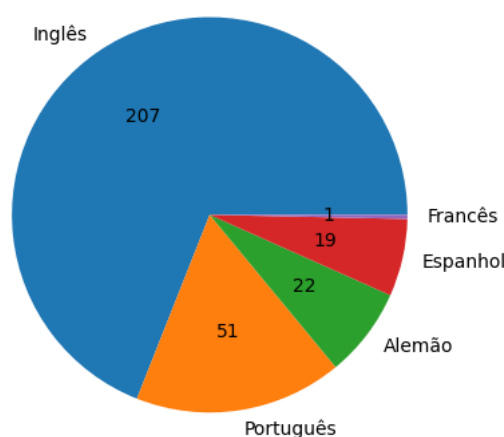
Para os fins deste trabalho, concatenaram-se os campos *title* e *description* e descartaram-se todos os demais, de forma que o sistema utilizasse apenas as informações fornecidas pelo usuário no momento da abertura do incidente. As 15 principais categorias presentes do conjunto de dados, ordenadas por quantidade de incidentes, são as seguintes: *Fileservices*, *Active Directory*, *Computer Services*, *Access Control*, *End-Of-Life*, *O365*, *Exchange*, *Create Account*, *Data Center*, *Identity Management*, *Fileshared*, *Telecom*, *Printer*, *Software general* e *Security*.

Os chamados são escritos, principalmente, em inglês; porém, muitos são escritos em alemão, espanhol, português e potencialmente em outros idiomas. A Figura 6 apresenta a distribuição de chamados por idioma, onde percebe-se que o conjunto de dados é altamente desbalanceado. Além disso, muitos chamados são escritos com erros gramaticais e abreviações. O conjunto de *title* e *description* totaliza, em média, 224 caracteres. Para fins de ilustração, a seguir apresenta-se um chamado típico (o campo *external_ID*, os nomes e as datas foram utilizados de forma fictícia para preservar a identidade do usuário original):

- a) *external_ID*: ABC123456;
- b) *title*: Acesso a arquivos;
- c) *description*: Bom dia. Preciso ter aceso ao computador do Leonardo Benitez. O memso foi desligado da empresa e preciso dos arquivos que ficaram na área de trabalho. pois se tratam de p-lanilhas de controle e tambem aos e-mails. O mesmo assinou a carta de autorização. Obrigada;

- d) *category*: Fileservice;
- e) *date_open*: 2022-01-01 10:23:19.000;
- f) *date_close*: 2022-01-02 09:15:56.000;
- g) *location*: BRLM;
- h) *solution*: O cliente conseguiu acesso e copiou os arquivos para a sua máquina;
- i) *analysts*: Leonardo Pereira.

Figura 6 – Distribuição de idiomas dos chamados



Fonte: Elaboração própria (2022).

O conjunto de 300 chamados utilizados está disponibilizado em Pereira (2022). Antes de os dados serem publicados, todas as informações pessoais e/ou sensíveis foram removidas (substituídas por uma *tag* indicando o conteúdo original, por exemplo: a frase "este texto foi escrito por Leonardo" é convertida para "este texto foi escrito por [NAME]"). A remoção foi realizada em três etapas: primeiro, a ferramenta baseada em Aprendizado de Máquina *AWS Comprehend PII Removal* foi utilizada; então, aplicou-se uma sequência de expressões regulares próprias; finalmente, todos os chamados foram manualmente verificados.

4.2 Implementação das técnicas

Todas as técnicas foram implementadas utilizando a linguagem de programação *Python*. O vetor gerado por cada uma das técnicas foi comparado utilizando a métrica de Similaridade do Cosseno (Equação 1). As únicas exceções foram o sistema especialista e o BM25, que utilizaram métricas de similaridade próprias. O sistema, então, recomenda os chamados com maior similaridade dentre o conjunto de recomendações possíveis.

Todas as técnicas foram implementadas seguindo uma estrutura de programação orientada a objetos, de forma que expuseram a mesma interface genérica. Assim, elas foram avaliadas seguindo a mesma implementação de métricas e metodologias.

4.2.1 Sistema especialista

Desenvolveu-se um sistema que, partir da presença ou não de certos termos, gerava um conjunto de "etiquetas" para cada documento. Utilizaram-se como termos 116 jargões de TI, complementados com mais 141 sinônimos. Visando aumentar o número de etiquetas identificadas para cada documento, preprocessou-se o texto da seguinte forma:

- a) convertendo para letras minúsculas;
- b) removendo os caracteres especiais —, ., ,, !, ?, _, e *;
- c) convertendo os caracteres para a sua representação mais próxima em *unicode*.

Após o pré-processamento, comparou-se o conjunto de etiquetas resultantes utilizando a métrica de Similaridade de Jaccard (Equação 2).

4.2.2 TF-IDF

Para a técnica de TF-IDF, utilizou-se a implementação da biblioteca *Sklearn* (na sua versão 0.23.2). O dicionário foi calculado com base com nos 20056 chamados que não foram utilizados para avaliação. Utilizou-se o mesmo pré-processamento do sistema especialista, aliado à remoção de *stop words* para a língua inglesa. Foram utilizadas as 500 palavras mais frequentes do dicionário, resultando em um vetor de 500 posições.

4.2.3 BM25

Foi adotada a implementação da biblioteca *rank_bm25* (versão 0.2.2). Utilizou-se o mesmo pré-processamento do sistema especialista e mantiveram-se os parâmetros padrão de $k1 = 1.5$, $b = 0.75$ e $epsilon = 0.25$. Como a técnica de BM25 requer uma métrica de comparação própria, utilizou-se, também, a implementação da biblioteca.

4.2.4 LDA

Para implementar a técnica de LDA, foi escolhida a biblioteca *Gensim* (versão 4.3.0). Utilizou-se o mesmo pré-processamento do sistema especialista, e a técnica foi configurada para 300 tópicos. Tal dimensão foi escolhida pois é o mesmo valor dos vetores da *Word2vec*, tornando a comparação mais justa, além de ser o valor utilizado no trabalho de Muni *et al.* (2017) e de ser um valor usual na literatura.

4.2.5 Word2vec e derivados

Foi adotada as implementações da biblioteca *Gensim*. Para a técnica de *Word2vec*, utilizou-se um modelo treinado no conjunto de dados *Google News*, que possui aproximadamente 100 bilhões de palavras da língua inglesa. Tal modelo produz um vetor de 300 dimensões.

Também treinou-se uma rede *Word2vec* nova, apenas com os 20056 chamados que não foram empregados para a avaliação (que totalizaram aproximadamente 1.6 milhões de palavras), também com 300 dimensões. Foi mantido o mesmo pré-processamento do sistema especialista. Para exemplificar o efeito do treinamento com base própria, foram escolhidas três palavras relacionadas a chamados de suporte (email, *problem* e trabalho) e utilizou-se a representação vetorial dos *embeddings* para verificar as cinco palavras mais parecidas. Os resultados são apresentados no Quadro 1, onde é possível perceber que a consulta por "problema" provê resultados que são mais condizentes com a atividade de suporte de TI, enquanto a consulta por "trabalho" nem sequer foi encontrada no modelo de língua inglesa (pois, naturalmente, essa palavra não existe na língua inglesa, na qual o modelo foi treinado).

Quadro 1 – Exemplos de similaridade entre palavras

Palavra	Conjunto <i>Google News</i>	Conjunto da <i>Skaylink</i>
email	e_mail, emails, E_mail, Email, roycimagala@hotmail.com	emails, mail, mails, messages, [x]
problem	problems, dilemma, prob_lem, conundrum, probem	issue, having, something, problems,
trabalho	Não encontrado	perfil, compra, aparelho, momento, verificacao

Fonte: Elaboração própria (2022).

Treinou-se *Doc2vec* com os mesmos 20056 chamados, técnicas para pré-processar os dados e dimensões do vetor de saída. O treinamento foi realizado com os parâmetros de *window* = 10, *min_count* = 1 e *epochs* = 100, obtidos após brevemente testar diversos valores e medir a precisão obtida. Não foi utilizado um modelo já treinado pois não havia um modelo disponível na biblioteca *Gensim*, e não foi encontrada outra fonte confiável para obter um modelo.

4.2.6 BERT e derivados

Para o modelo BERT original, foi adotada a implementação *BERT-base-multilingual-cased* da biblioteca *HuggingFace* (versão 4.25.1), que foi treinado com 104 idiomas, tomando-se como *embedding* a saída do *token* especial [CLS], que possui dimensão de 768 elementos. A biblioteca *SentenceTransformers* foi utilizada para os modelos *Sentence-BERT*, na versão 2.2.2. O modelo multi-idioma utilizado foi o

distiluse-base-multilingual-cased-v1, que foi treinado com 15 idiomas (incluindo inglês, alemão, espanhol e português), e provê um vetor de 512 dimensões. Ao retrainar a rede com os dados da Skaylink, tomou-se como base esse mesmo modelo de treinou-se por 3 épocas, com os 20056 não usados para a avaliação. O modelo de língua inglesa escolhido foi o *all-mpnet-base-v2*, que fornece um vetor de 768 dimensões.

O modelo BERT e suas variações costumam apresentar resultados melhores quando o texto não é pré-processado (KAMATH; GRAHAM; EMARA, 2022); portanto, utilizou-se o sempre texto original.

4.3 Identificação da melhor técnica

Conforme descrito no Capítulo 3, a similaridade entre os chamados foi calculada a partir da posição dos cartões, que foram exportados para arquivos do tipo CSV (um arquivo para cada subgrupo). Para cada modelo, esses arquivos CSV foram carregados um a um, e o modelo identificou os chamados mais similares para cada chamado; então, avaliou-se o modelo nesse subgrupo e, por fim, calculou-se a média das métricas de avaliação para o modelo.

A métrica de precisão (Equação 3) foi escolhida para comparar os resultados. Adicionalmente, criou-se a métrica de "acurácia pelo-menos-um" (formalizada na Equação 7, onde N é o número de chamados, y_i é o conjunto de chamados relevantes e \hat{y}_i é o conjunto de chamados recomendados), com o objetivo de possuir uma métrica que melhor reflita a percepção do analista quanto à qualidade da técnica. Tal métrica considera como um "acerto" caso qualquer um dos cinco chamados recuperados seja de fato um chamado relevante. A métrica de revocação (Equação 4) não foi utilizada pois, como se decidiu que cada técnica recuperaria cinco chamados e havia também sempre cinco chamados considerados relevantes, revocação e precisão possuem por definição sempre o mesmo valor.

$$\text{Acurácia}_{\text{pelo-menos-um}}(y, \hat{y}) = \frac{\sum_i^N \lambda(y_i, \hat{y}_i)}{N} \quad (7)$$

em que

$$\lambda(a, b) = \begin{cases} 1, & |a \cap b| > 0 \\ 0 & \end{cases} \quad (8)$$

Os resultados foram sumarizados no Quadro 2. Percebe-se que a técnica de *Sentence-BERT* multi-idioma apresenta o melhor resultado, com 35,1% de precisão e 78,7% de acurácia pelo-menos-um (ou seja, três a cada quatro vezes o sistema recomenda ao menos um chamado anterior que se parece com o chamado em análise). *Sentence-BERT* também é a técnica mais recente, publicada em 2019 (REIMERS;

GUREVYCH, 2019), sendo, portanto, esperado que ela apresentaria os melhores resultados.

Duas técnicas surpreenderam pela sua baixa precisão: BERT multi-idioma (17,2%) e *Doc2vec* (5,8%). Apesar de ambos obterem resultados melhores do que a seleção aleatória (5,5%), a literatura indica que a utilização dessas técnicas usualmente resulta em precisões comparáveis a de técnicas clássicas como TF-IDF (29,6%). Para o *Doc2vec*, uma possível explicação é que o treinamento gerou um super ajuste (também conhecido como *overfit*) ao conjunto, visto o baixo número de chamados utilizados, (20056) e por não ter sido tomado como base um modelo previamente treinado (ou seja, o treinamento foi realizado "do zero"). Quanto ao BERT multi-idioma, uma possível explicação foi a escolha do *token* especial [CLS] para como *embedding*, pois há outros métodos possíveis para extrair *embeddings* de uma rede BERT (DEVLIN *et al.*, 2019). Não se pode descartar que a baixa precisão se deva também a erros de programação ou à má escolha de parâmetros durante o treinamento.

Quanto ao efeito obtido pelo retreinamento das redes neurais (a partir de um modelo base existente), houve resultados divergentes. Embora o retreinamento da *Word2vec* tenha melhorado a sua precisão, o retreinamento da *Sentence-BERT* multi-idioma não obteve o mesmo resultado, e ambas performaram de forma similar. Tal fato pode ser explicado, pois redes BERT precisam de um grande volume de dados para serem treinadas, e a utilização de conjuntos pequenos pode, inclusive, ocasionar a rede “esquecer” parte do que aprendeu anteriormente.

Quadro 2 – Comparação das técnicas implementadas

Nome	Acurácia pelo-menos-um	Precisão
BM25	59,0%	23,7%
BERT multi-idioma	50,0%	17,2%
<i>Doc2vec</i>	27,3%	5,8%
LDA	66,3%	20,9%
Seleção aleatória	26,0%	5,5%
<i>Sentence-BERT</i> inglês	74,3%	30,1%
<i>Sentence-BERT</i> multi-idioma	78,7%	35,1%
<i>Sentence-BERT</i> retreinado	78,7%	32,7%
Sistema especialista	42,7%	17,2%
TF-IDF	69,0%	29,7%
<i>Word2vec</i> inglês	58,3%	23,4%
<i>Word2vec</i> retreinado	68,7%	26,2%

Fonte: Elaboração própria (2022).

4.4 Experimentos exploratórios

Segundo Gil (2008), a pesquisa exploratória serve de preparação para a pesquisa explicativa. Objetivando permitir a futura explicação dos resultados obtidos

e identificação das causas para o comportamento de cada técnica implementada, esta seção apresenta os resultados obtidos com diversas variações da metodologia utilizada neste trabalho. Para cada experimento, serão apresentados uma descrição dos resultados e possíveis interpretações, de forma a guiar a elaboração posterior de trabalhos explicativos.

Em muitos dos experimentos, o conjunto de dados de avaliação teve de ser reduzido. Como as métricas utilizadas são influenciadas pelo tamanho do conjunto de documentos a ser buscado, comparou-se sempre os resultados com um subconjunto de n chamados selecionados aleatoriamente, onde n corresponde ao número de chamados utilizados durante o experimento.

4.4.1 Segmentando os chamados por idioma

Os dados utilizados possuem chamados de suporte em vários idiomas, da mesma forma que o trabalho diário de muitos analistas de suporte de TI. Tal característica impacta diretamente a aplicação de técnicas de RI, na medida em que tornam o problema mais desafiador. Para mensurar quantitativamente o impacto do idioma na qualidade dos resultados, realizaram-se dois experimentos, selecionando apenas os chamados de idioma inglês e de idioma português. Para permitir essa seleção, aplicou-se uma técnica de classificação de idioma (aprendizado supervisionado, utilizando a biblioteca *Spacy*), verificou-se manualmente os resultados e corrigiram-se os erros.

Foram selecionados apenas os chamados da língua inglesa (207 chamados), e avaliou-se o sistema com as mesmas métricas do capítulo anterior. Os resultados foram compilados no Quadro 3 e no Quadro 4, onde se percebe que a técnica de *Sentence-BERT* inglês apresentou uma melhoria significativa na precisão (7,4%), enquanto o seu equivalente *Sentence-BERT* multi-idioma manteve os resultados quase iguais (melhoria de 1,1%). Efeito similar foi observado entre *Word2vec* inglês (6% de aumento na precisão) e *Word2vec* retreinado (4,1%). A melhoria mais expressiva se deu na técnica de TF-IDF, com aumento de 8,3%.

Também segmentou-se apenas os chamados em língua portuguesa (51 chamados), e utilizou-se a mesma metodologia. Os resultados foram similares (Quadros 5 e 6), na medida em que os modelos específicos para língua inglesa pioraram (*Sentence-BERT* inglês perdeu 2,9% de precisão, e *Word2vec* inglês 4,0%), enquanto os modelos retreinados melhoraram ou mantiveram a performance. Entretanto, é importante salientar que o número de chamados é consideravelmente menor; portanto, há uma maior variabilidade estatística nos textos, diminuindo a validade das conclusões.

Quadro 3 – Segmentação por lingua inglesa, conjunto de controle

Nome	Acurácia pelo-menos-um	Precisão
BM25	65,6%	24,2%
BERT multi-idioma	61,3%	20,9%
Doc2vec	35,0%	7,7%
LDA	67,3%	21,5%
Seleção aleatória	39,7%	9,1%
Sentence-BERT inglês	78,4%	32,0%
Sentence-BERT multi-idioma	79,9%	38,2%
Sentence-BERT retreinado	81,3%	36,4%
Sistema especialista	51,4%	16,3%
TF-IDF	73,9%	30,7%
Word2vec inglês	64,7%	25,7%
Word2vec retreinado	77,6%	29,6%

Fonte: Elaboração própria (2022).

Quadro 4 – Segmentação por lingua inglesa, conjunto sob teste

Nome	Acurácia pelo-menos-um	Precisão
BM25	69,0%	29,7%
BERT multi-idioma	60,9%	22,9%
Doc2vec	40,4%	9,5%
LDA	72,8%	27,7%
Seleção aleatória	35,1%	8,3%
Sentence-BERT inglês	82,0%	39,4%
Sentence-BERT multi-idioma	80,6%	39,3%
Sentence-BERT retreinado	81,4%	36,2%
Sistema especialista	57,7%	23,7%
TF-IDF	80,5%	39,0%
Word2vec inglês	69,2%	31,7%
Word2vec retreinado	77,6%	33,7%

Fonte: Elaboração própria (2022).

4.4.2 Segmentação por categorias facilmente distinguíveis

Outro fator que possivelmente influencia a qualidade das recomendações são as categorias existentes no conjunto de documento: quando mais estereotípicos - ou bem diferenciáveis - forem os chamados, espera-se que seja mais fácil dar recomendações boas. Pode-se ilustrar tal raciocínio com a seguinte analogia: se uma pessoa vai a uma biblioteca procurando o livro mais parecido com o livro de receitas da sua vó, e todos os livros da biblioteca estão misturados, provavelmente será difícil encontrar os livros desejados. Porém, se todos os livros sobre culinária estiverem agrupados no lado direito da biblioteca, os de literatura no lado esquerdo, e os de química empilhados no meio da biblioteca, provavelmente a pessoa encontrará bons livros com muito mais facilidade. Para testar essa hipótese, segmentaram-se apenas os chamados de três categorias bem distintas entre si: *Identity Management*, *DataCenter* e *End-Of-Life*,

Quadro 5 – Segmentação por língua portuguesa, conjunto de controle

Nome	Acurácia pelo-menos-um	Precisão
BM25	98,2%	43,0%
BERT multi-idioma	96,5%	37,4%
Doc2vec	85,0%	30,8%
LDA	100,0%	42,5%
Seleção aleatória	93,3%	36,2%
Sentence-BERT inglês	94,9%	44,5%
Sentence-BERT multi-idioma	98,2%	50,7%
Sentence-BERT retreinado	98,2%	46,7%
Sistema especialista	91,6%	39,6%
TF-IDF	96,7%	45,5%
Word2vec inglês	96,7%	40,8%
Word2vec retreinado	93,0%	43,5%

Fonte: Elaboração própria (2022).

Quadro 6 – Segmentação por língua portuguesa, conjunto sob teste

Nome	Acurácia pelo-menos-um	Precisão
BM25	96,8%	41,4%
BERT multi-idioma	91,7%	35,2%
Doc2vec	83,5%	34,8%
LDA	90,3%	34,8%
Seleção aleatória	93,3%	37,2%
Sentence-BERT inglês	98,4%	41,6%
Sentence-BERT multi-idioma	98,4%	48,6%
Sentence-BERT retreinado	96,8%	48,0%
Sistema especialista	80,2%	37,2%
TF-IDF	96,8%	46,8%
Word2vec inglês	85,0%	36,8%
Word2vec retreinado	93,5%	43,2%

Fonte: Elaboração própria (2022).

totalizando 90 chamados. Avaliou-se com a mesma metodologia da seção anterior.

Nos Quadros 7 e 8 observa-se que todas as técnicas apresentaram melhorias e, portanto, a hipótese foi comprovada. Em especial, os modelos retreinados apresentaram melhoras maiores que os seus equivalentes não-retreinados: *Word2vec* retreinado melhorou 10,3 pontos percentuais de precisão (contra 3,6 pontos percentuais do *Word2vec* não retreinado), e *Sentence-BERT* retreinado melhorou 2,6 pontos percentuais de precisão (contra 1,2 pontos percentuais do *Sentence-BERT* multi-idioma). Destaca-se, também, que os modelos mais simples/tradicionais foram os que mais se beneficiaram com essa estereotipificação: o sistema especialista progrediu 15,2% (a maior melhoria percentual do experimento, na técnica que até agora havia sido a pior) e TF-IDF tornou-se a melhor técnica dentre as avaliadas.

Quadro 7 – Segmentação por categorias facilmente distinguíveis, conjunto de controle

Nome	Acurácia pelo-menos-um	Precisão
BM25	75,3%	27,9%
BERT multi-idioma	79,7%	25,0%
Doc2vec	54,1%	14,4%
LDA	81,0%	30,6%
Seleção aleatória	67,4%	17,1%
Sentence-BERT inglês	78,7%	36,9%
Sentence-BERT multi-idioma	86,6%	41,3%
Sentence-BERT retreinado	84,3%	41,1%
Sistema especialista	74,2%	24,2%
TF-IDF	79,8%	35,3%
Word2vec inglês	82,1%	33,3%
Word2vec retreinado	84,3%	36,9%

Fonte: Elaboração própria (2022).

Quadro 8 – Segmentação por categorias facilmente distinguíveis conjunto sob teste

Nome	Acurácia pelo-menos-um	Precisão
BM25	80,9%	37,8%
BERT multi-idioma	82,1%	35,8%
Doc2vec	69,6%	18,0%
LDA	93,2%	40,7%
Seleção aleatória	69,6%	18,4%
Sentence-BERT inglês	87,7%	42,3%
Sentence-BERT multi-idioma	87,6%	42,3%
Sentence-BERT retreinado	88,8%	43,7%
Sistema especialista	80,9%	39,4%
TF-IDF	92,1%	48,1%
Word2vec inglês	82,0%	36,9%
Word2vec retreinado	92,1%	47,2%

Fonte: Elaboração própria (2022).

4.4.3 Anotações de relevância por *clustering*

Outra hipótese a ser testada refere-se ao efeito da metodologia escolhida para definir os chamados mais relevantes a cada chamado (utilizou-se originalmente apenas a distância euclidiana). Uma metodologia alternativa é primeiro *clusterizar* os chamados em grupos de aproximadamente cinco chamados cada *cluster*, e considerar como "chamado relevante" de cada chamado todos os outros que pertencentes ao seu mesmo *cluster*.

Implementou-se essa metodologia alternativa por meio do algoritmo *Spectral Clustering*, utilizando a biblioteca *Sklearn* com parâmetros $\gamma = 1$ e $n_neighbors = 10$, e configurado para encontrar 20 *clusters*. O número de *clusters* foi adotado para que cada *cluster* possuísse aproximadamente 5 chamados, o mesmo número da

metodologia original de anotação. Escolheu-se esse algoritmo pois era o único que satisfazia a todos os seguintes critérios:

- a) já implementados na biblioteca *Sklearn*;
- b) é capaz de identificar *clusters* não convexos;
- c) não elimina amostras consideradas "anomalias" (isto é, não pertencem claramente a nenhum *cluster*). Tal eliminação é indesejada, pois espera-se utilizar todos os chamados disponíveis para a avaliação.

Os resultados obtidos são apresentados no Quadro 9. Se comparados aos resultados originais (Quadro 2), percebe-se que todas as técnicas obtiveram uma diminuição de aproximadamente 3% na precisão. Uma possível explicação para a diferença é que o número de documentos relevantes passa a ser variável (e não exatamente cinco), de forma que alguns chamados possuirão menos de cinco chamados similares, prejudicando a métrica de precisão igualmente para todos os modelos. Apesar disso, houve pouca alteração na ordenação de quais são as melhores técnicas, e ambas as métricas foram afetadas igualmente pela mudança metodológica. Dessa forma, é possível concluir que metodologia de anotações de relevância não é um fator crítico para a identificação da melhor técnica.

Quadro 9 – Comparação das técnicas, utilizando a anotação por *clustering*

Nome	Acurácia pelo-menos-um	Precisão
BM25	54,0%	20,6%
BERT multi-idioma	47,3%	15,7%
Doc2vec	16,3%	3,4%
LDA	58,3%	17,4%
Seleção aleatória	17,7%	3,7%
Sentence-BERT inglês	66,7%	25,3%
Sentence-BERT multi-idioma	71,3%	29,7%
Sentence-BERT retreinado	70,7%	27,9%
Sistema especialista	41,0%	15,3%
TF-IDF	62,7%	25,7%
Word2vec inglês	54,3%	20,9%
Word2vec retreinado	60,7%	22,8%

Fonte: Elaboração própria (2022).

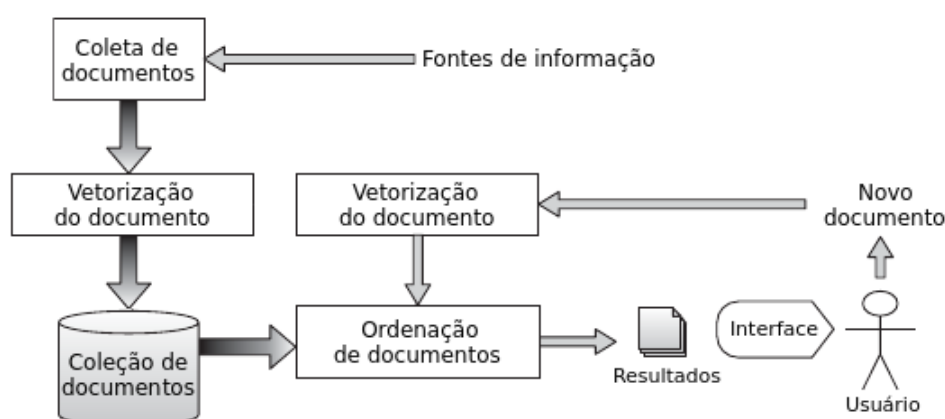
4.5 Desenvolvimento do protótipo

A fim de operacionalizar um protótipo do sistema proposto, desenvolveu-se um sistema web *backend* que fornece uma *Application Programming Interface* (API) para as atividades necessárias ao funcionamento do protótipo: autenticar usuários, cadastrar novos chamados de suporte e buscar chamados de suportes similares. Para tanto, utilizou-se a biblioteca *FastAPI*, para desenvolvimento de sistema usando a

linguagem *Python*, em conjunto com um banco de dados *Postgres* para armazenar os chamados e as informações referentes aos usuários.

A arquitetura do sistema seguiu proposta de Mitkov (2003) (Figura 5), e foi implementada conforme a Figura 7. Quando um novo chamado é registrado no sistema, sua representação vetorial é calculada (utilizando a técnica de *Sentence-BERT* multi-idioma); então, são buscados os 100 últimos chamados registrados, dentre os quais os 5 chamados mais similares são selecionados. Tal restrição pelos 100 últimos chamados se dá por motivos práticos, para evitar a sobrecarga do sistema, mas uma implementação futura do sistema pode permitir a busca em todo o banco de dados. Adicionalmente, o novo chamado também é armazenado no banco de dados do sistema, permitindo a sua utilização em recomendações futuras.

Figura 7 – Arquitetura do protótipo



Fonte: Adaptado de Mitkov (2003).

Para permitir um acesso gráfico à aplicação, desenvolveu-se uma interface web utilizando a biblioteca *VueJS* e a linguagem de programação *JavaScript*. Por meio dessa interface, o analista pode realizar o *login* (Figura 8), visualizar os chamados anteriores (Figura 9), cadastrar novos chamados manualmente (Figura 10) e, então, visualizar cinco recomendações de chamados anteriores similares (Figura 11). O protótipo completo foi operacionalizado na plataforma *Amazon Web Services*, usando as seguintes soluções nativas: *Amazon Relational Databases* (RDS), *Amazon Elastic Compute Cloud* (EC2) e *Amazon Elastic Block Store* (EBS).

Figura 8 – Tela de login do protótipo

Fonte: Elaboração própria (2022).

Figura 9 – Tela de visualização de chamados

Main menu

Dashboard

Profile

Edit Profile

Change Password

Your tickets

Admin

Manage Users

Create User

Logout

Collapse

Bea-DW

List of tickets

CREATE TICKET

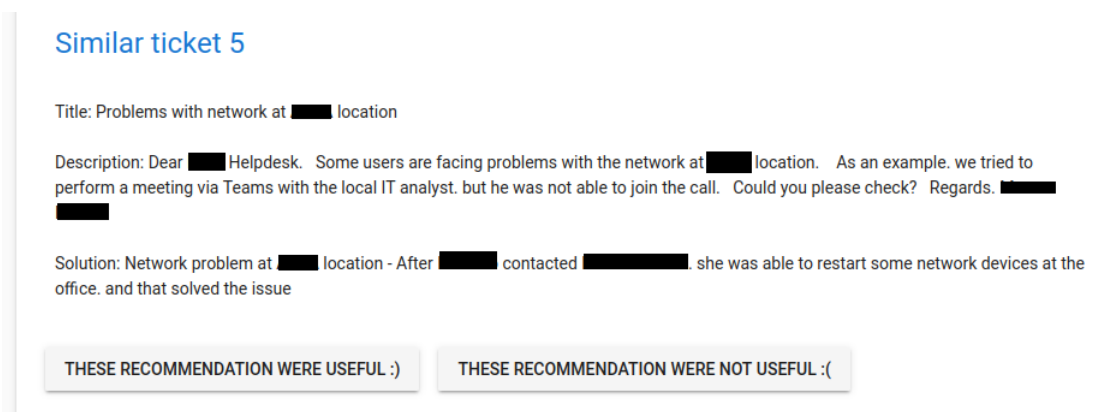
ID ↑	Título	Descrição	
94973	Email Doesnt work	This email inbox doesnt work correctly. I cannot receive emails but can send form this inbox. Im receiving no emails and have missed meetings.	
94974	File Share Access - [NAME] [NAME] ([NAME])	Service Request: File Share Access Grant access to file share for the following user: [NAME] [NAME] ([NAME]) (Username: [NAME]) Access Type: (RW) Read / Write File Share: J:/WST-Allg J:/WSTAZUBI/Allgemein J:/WSTAZUBI/Ausbildungsbeginn 2018 Der Azubi muss auf 14 interaktive Lernprogramme der Firma [NAME] zugreifen können. Die Softwer ist installiert auf deel-pc-10182/[NAME] programme Requestor: [NAME] ([NAME]) Comment:	
94975	Remove File Share Access - [NAME] ([NAME])	Service Request: Remove File Share Access Remove file share access for the following user: [NAME] ([NAME]) (Username: [USERNAME]) File Share: -Zugriff auf das E-Mail Postfach [EMAIL] „Senden als Treasury“ entfernen Ordnerzugriffe auf Abschluss. AZB Kaufleute. [NAME] INFO. [NAME] Marketing_Controlling entfernen - Aus dem [NAME] Kfm Email Verteller entfernen, sowie aus den selbigen Gruppen Requestor: [NAME] ([NAME]) Comment:	
94976	Certificate Signing request - [NAME]	Hi. Could you please sign the attached files for me? Please give the information to me as I can install it myself. No need to go through [NAME] it. Thank you in advance. [NAME]	
	Light User for	Service Request: Light User for external Create Light user for [USERNAME] [NAME] Firstname: [NAME] Lastname: [NAME] Mailaddress: [EMAIL] Phone: [PHONE] 0 Cost	

© Bea-DW

Fonte: Elaboração própria (2022).

Figura 10 – Tela de cadastro de chamados

Fonte: Elaboração própria (2022).

Figura 11 – Tela de chamados recuperados pelo protótipo

Similar ticket 5

Title: Problems with network at [REDACTED] location

Description: Dear [REDACTED] Helpdesk. Some users are facing problems with the network at [REDACTED] location. As an example, we tried to perform a meeting via Teams with the local IT analyst, but he was not able to join the call. Could you please check? Regards, [REDACTED]

Solution: Network problem at [REDACTED] location - After [REDACTED] contacted [REDACTED], she was able to restart some network devices at the office, and that solved the issue

THESE RECOMMENDATION WERE USEFUL :)

THESE RECOMMENDATION WERE NOT USEFUL :()

Fonte: Elaboração própria (2022).

A implementação do protótipo totalizou pouco mais de cinco mil linhas de código *Python* e 2400 linhas de código *JavaScript*, além de arquivos de configuração, definição de infraestrutura, *scripts* de automação, entre outros. Armazenou-se em um repositório git, que totaliza mais de 270 *commits*.

4.6 Avaliação do protótipo

Um grupo de seis analistas teve acesso ao sistema durante três semanas, nas quais puderam utilizá-lo no seu trabalho diário. Gravou-se um vídeo com explicações de como utilizar o sistema e explicou-se o propósito do projeto. Após registrar um novo chamado e receber as recomendações, o analista pôde dar o seu *feedback* (indicando se a recomendação foi útil ou não, conforme a Figura 11).

Entretanto, a adoção do sistema foi muito baixa, e apenas três *feedbacks* foram coletados durante o período. Durante a última semana, foi disponibilizado um formulário *online* (seguindo a estrutura do Apêndice A, e implementado com a ferramenta *Google Forms*) para a avaliação do protótipo, porém apenas uma resposta foi coletada.

Uma possível explicação para a baixa adoção do sistema é a inconveniência em seu uso: o analista de suporte já está acostumado a trabalhar com o sistema de gerenciamento, no qual consegue visualizar e responder a todos os chamados. Por sua vez, o protótipo desenvolvido constitui um sistema separado, no qual os chamados devem ser manualmente copiados e colados. Dessa forma, o seu uso torna-se inconveniente e anti-intuitivo. Idealmente, o sistema de recomendação integraria diretamente com o sistema de gerenciamento, de forma que o analista possa realizar todas as suas atividades em um mesmo ambiente.

Devido ao baixo número de *feedbacks* e respostas coletadas, tais resultados não serão analisados neste trabalho. Entretanto, a existência de trabalhos acadêmicos recentes na área, incluindo de grandes empresas implementando os seus próprios

sistemas de RI, indica a real perspectiva de uso prático dessas técnicas. Dessa forma, a hipótese mais provável é de que a adoção do sistema foi baixa devido a dificuldades de o analista interagir com o *software*, que deve ser aprimorado para além do protótipo desenvolvido para integrar-se de forma natural na rotina do analista de suporte.

5 CONSIDERAÇÕES FINAIS

Ao longo deste trabalho, realizou-se a comparação de onze técnicas de Recuperação de Informação, aplicadas sobre um conjunto de dados referentes a chamados de suporte de TI. Tais técnicas incluíram diversas abordagens de RI e abrangem desde técnicas clássicas/consolidadas até o estado da arte em pesquisa, de forma que foi possível identificar, de uma maneira clara, as possibilidades para implementar um sistema que, dado um novo chamado de suporte, seja capaz de buscar chamados de suporte similares.

O melhor resultado foi obtido com a técnica *Sentence-BERT*, na sua variação multi-idioma *distiluse-base-multilingual-cased-v1*, onde 78,7% das recomendações realizadas pelo modelo foram consideradas relevantes. As duas outras variações testadas da *Sentence-BERT* apresentaram o segundo e terceiro melhores resultados, seguidas pela técnica de TF-IDF.

Além disso, este trabalho procurou contribuir com a comunidade acadêmica na medida em que disponibilizou - de forma gratuita e irrestrita - o conjunto de dados utilizado, descreveu em detalhes a implementação de cada técnica, explorou as condições que afetam os resultados de cada modelo e demonstrou a viabilidade de um sistema de RI para chamados de suporte ao implementar um protótipo viável mínimo. Tais resultados atendem aos objetivos propostos, e norteiam o desenvolvimento de trabalhos futuros na área.

Entretanto, ao avaliar o quanto o protótipo auxiliou nas atividades diárias de uma equipe de suporte de TI, observou-se uma baixa adoção do sistema. Tais resultados apontam para a necessidade de melhorar a interação entre o analista e o *software*, de forma que a sua utilização seja simples e intuitiva, efetivamente facilitando o trabalho do analista.

Ao interpretar os resultados obtidos, pode parecer - à primeira vista - que as métricas de avaliação indicam resultados ruins. É importante salientar que a metodologia de avaliação foi definida de forma estrita e apenas 5 chamados dentre 99 foram considerados relevantes. Ademais, a própria natureza dos dados (textos curtos, pouco explicativos, e com muitos jargões técnicos) torna desafiadora a aplicação de técnicas de recuperação de informação. Apesar disso, todas as técnicas implementadas apresentaram resultados melhores do que a seleção aleatória, indicando que elas conseguiram capturar a semântica dos chamados de suporte.

Como resultado complementar, corroborou-se que a rede *Sentence-BERT* apresenta melhores resultados em RI do que a BERT original, conforme apresentado no trabalho de Reimers e Gurevych (2019). Tal superioridade manteve-se em todos os experimentos realizados, para todas as variações de *Sentence-BERT* testadas e para

todas as métricas de avaliação utilizadas.

Cabe aqui ressaltar o resultado positivo da técnica de TF-IDF, que é de simples implementação e computacionalmente rápida. A técnica também se mostrou robusta em todos os experimentos realizados, apresentando sempre resultados consistentes, enquanto outras técnicas não performaram igualmente bem em todas as condições. Dessa forma, é possível que o sistema final venha a ser implementado com TF-IDF, ao invés de *Sentence-BERT*.

Por fim, destaca-se que o sistema especialista obteve resultados consideravelmente ruins, em praticamente todos os cenários testados. Apesar de sua implementação ter sido simplista, isso demonstra a dificuldade de implementar "manualmente" sistemas de Recuperação de Informação, sendo justificada a utilização de técnicas mais avançadas de Aprendizado de Máquina.

5.1 Sugestões para trabalhos futuros

A partir dos resultados obtidos, é possível dar continuidade ao trabalho realizado com, por exemplo, os seguintes tópicos:

- a) para não necessitar restringir o protótipo a buscar entre apenas os 100 últimos chamados registrados, utilizar um banco de dados com suporte nativo a busca por similaridade vetorial, como por exemplo o *software ElasticSearch*;
- b) integrar o protótipo com o sistema de gerenciamento, facilitando a utilização do sistema;
- c) melhorar os resultados obtidos com um retreinamento mais longo do modelo *Sentece-BERT*.

REFERÊNCIAS

- AL-GHOSSEIN, M. *et al.* Adaptive collaborative topic modeling for online recommendation. In: *Proceedings of the 12th ACM Conference on Recommender Systems*. New York, NY, USA: Association for Computing Machinery, 2018. (RecSys '18), p. 338–346. ISBN 9781450359016. Disponível em: <https://doi.org/10.1145/3240323.3240363>. 24
- AL-HAWARI, F.; BARHAM, H. A machine learning based help desk system for it service management. *Journal of King Saud University - Computer and Information Sciences*, v. 33, n. 6, p. 702–718, 2021. ISSN 1319-1578. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1319157819300515>. 11, 26
- BAI, X. *et al.* Scientific paper recommendation: A survey. *IEEE Access*, PP, p. 1–1, 01 2019. 16
- CEZAR, N. L. *et al.* Applying a post-processing strategy to consider the multiple interests of users of a paper recommender system. In: *XVII Brazilian Symposium on Information Systems*. New York, NY, USA: Association for Computing Machinery, 2021. (SBSI 2021). ISBN 9781450384919. Disponível em: <https://doi.org/10.1145/3466933.3466985>. 24
- CHOLLET, F. *Deep Learning with Python*. 1st. ed. Greenwich, CT, USA: Manning Publications Co., 2017. ISBN 1617294438, 9781617294433. 17, 18
- DELUCIA, A.; MOORE, E. *Analyzing HPC Support Tickets: Experience and Recommendations*. arXiv, 2020. Disponível em: <https://arxiv.org/abs/2010.04321>. 23, 25
- DEVLIN, J. *et al.* BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186. Disponível em: <https://aclanthology.org/N19-1423>. 21, 23, 35
- DYRHOVDEN, F. S.; NORVANG, E.; SUND, M. *Word embeddings for recommending semantically similar support tickets*. Dissertação (Bachelor's Thesis) — Western Norway University of Applied Sciences, 2021. Disponível em: <https://hvlopen.brage.unit.no/hvlopen-xmlui/handle/11250/2983809>. 25
- FACELI, K. *et al.* *Inteligência artificial: uma abordagem de aprendizado de máquina*. [S.l.]: LTC, 2011. 17, 18, 19, 28
- FENG, L.; SENAPATI, J.; LIU, B. *TaDaa: real time Ticket Assignment Deep learning Auto Advisor for customer support, help desk, and issue ticketing systems*. arXiv, 2022. Disponível em: <https://arxiv.org/abs/2207.11187>. 25
- FIALHO, F. A. P. *Gestão do Conhecimento e aprendizagem*. [S.l.]: Visual Books, 2006. 11
- GIL, A. C. *Metodos e tecnicas de pesquisa social*. 6 ed. ed. Sao Paulo :: Atlas,, 2008. 27, 35

- GUO, W. *et al.* Detext: A deep text ranking framework with bert. *CoRR*, abs/2008.02460, 2020. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr2008.html#abs-2008-02460>. 15
- KAMATH, U.; GRAHAM, K.; EMARA, W. *Transformers for Machine Learning: A Deep Dive*. CRC Press, 2022. (Chapman & Hall/CRC Machine Learning & Pattern Recognition). ISBN 9780367771652. Disponível em: <https://books.google.com.br/books?id=9FC-zgEACAAJ>. 19, 20, 21, 23, 24, 34
- LIU, H.; GEGOV, A.; COCEA, M. *Rule based systems for big data: a machine learning approach*. 1. ed. [S.l.]: Springer, 2016. v. 13. (Studies in Big Data, v. 13). ISBN 9783319236957. 22
- MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008. ISBN 978-0-521-86571-5. Disponível em: <http://nlp.stanford.edu/IR-book/information-retrieval-book.html>. 11, 12, 14, 15, 16, 22, 23
- MIKOLOV, T. *et al.* Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781>. 15, 23
- MITKOV (Ed.). *The Oxford handbook of computational linguistics*. Oxford [u.a.]: Oxford Univ. Press, 2003. ISBN 0-19-823882-7. 15, 22, 29, 41
- MOLINO, P.; ZHENG, H.; WANG, Y.-C. Cota: Improving the speed and accuracy of customer support through ranking and deep networks. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2018. (KDD '18), p. 586–595. ISBN 9781450355520. Disponível em: <https://doi.org/10.1145/3219819.3219851>. 26
- MULSA, R. A. C.; SPANAKIS, G. Evaluating bias in Dutch word embeddings. In: *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*. Barcelona, Spain (Online): Association for Computational Linguistics, 2020. p. 56–71. Disponível em: <https://aclanthology.org/2020.gebnlp-1.6>. 15
- MUNI, D. P. *et al.* Recommending resolutions of itil services tickets using deep neural network. In: *Proceedings of the Fourth ACM IKDD Conferences on Data Sciences*. New York, NY, USA: Association for Computing Machinery, 2017. (CODS '17). ISBN 9781450348461. Disponível em: <https://doi.org/10.1145/3041823.3041831>. 11, 12, 25, 32
- PEREIRA, L. S. B. *Semantic Similarity of IT Support Tickets*. Zenodo, 2022. Dataset on Zenodo. Disponível em: <https://doi.org/10.5281/zenodo.7426225>. 31
- REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In: INUI, K. *et al.* (Ed.). *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 2019. p. 3980–3990. ISBN 978-1-950737-90-1. Disponível em: <http://dblp.uni-trier.de/db/conf/emnlp/emnlp2019-1.html#ReimersG19>. 14, 16, 24, 35, 45

- REIMERS, N.; GUREVYCH, I. Making monolingual sentence embeddings multilingual using knowledge distillation. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020. p. 4512–4525. 15
- RICCI, F. *et al. Recommender systems handbook*. New York; London: Springer, 2011. 14
- RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. New Jersey: Pearson Prentice Hall, 2010. 1152 p. 17, 18
- SEVERINO, A. J. *Metodologia do trabalho científico*. [S.l.]: Cortez, 2016. 27
- SILVA, C.; VASCONCELOS, A. Using the ideal model for the construction of a deployment framework of it service desks at the brazilian federal institutes of education. *Software Quality Journal*, v. 28, 09 2020. 11
- SINGHAL, A. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, v. 24, n. 4, p. 35–43, 2001. Disponível em: <http://dblp.uni-trier.de/db/journals/debu/debu24.html#Singhal01>. 14
- STAIR, R.; REYNOLDS, G. *Principles of Information Systems*. 9th. ed. Boston, MA, USA: Course Technology Press, 2009. ISBN 0324665288. 11
- VASWANI, A. *et al.* Attention is all you need. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 6000–6010. ISBN 9781510860964. 19, 20, 21, 24
- WAHBA, Y.; MADHAVJI, N. H.; STEINBACHER, J. Evaluating the effectiveness of static word embeddings on the classification of it support tickets. In: *Proceedings of the 30th Annual International Conference on Computer Science and Software Engineering*. USA: IBM Corp., 2020. (CASCON '20), p. 198–206. 26
- YIN, D. *et al.* Ranking relevance in yahoo search. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 323–332. ISBN 9781450342322. Disponível em: <https://doi.org/10.1145/2939672.2939677>. 24
- ZAIDI, S. *et al.* A multiapproach generalized framework for automated solution suggestion of support tickets. *International Journal of Intelligent Systems*, 09 2021. 12
- ZHOU, W. *et al.* Resolution recommendation for event tickets in service management. *IEEE Transactions on Network and Service Management*, v. 13, p. 1–1, 12 2016. 11

APÊNDICES

APÊNDICE A – QUESTIONÁRIO APLICADO

1. Em geral, as recomendações do protótipo lhe ajudaram no seu trabalho? ¹
2. De que forma o protótipo mais lhe ajudou? ²
3. Atualmente o sistema recomenda 5 chamados anteriores. Qual você considera que seria o número adequado? ³
4. De que forma você gostaria de interagir com o sistema? Diretamente integrado no sistema de gestão, um aplicativo no celular, ...? Por quê?²

¹ O campo aceitava respostas de 0 a 10

² O campo aceitava respostas da norma de texto livre

³ O campo aceitava uma resposta numérica