UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE QUIXADÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

MESTRADO ACADÊMICO EM COMPUTAÇÃO

VALMIR OLIVEIRA DOS SANTOS JÚNIOR

APRENDIZAGEM E ROTULAÇÃO DE INTENÇÕES SEMIAUTOMÁTICA PARA MODELOS DE CLASSIFICAÇÃO DE TEXTO EM LINGUAGEM NATURAL

QUIXADÁ

2024

VALMIR OLIVEIRA DOS SANTOS JÚNIOR

APRENDIZAGEM E ROTULAÇÃO DE INTENÇÕES SEMIAUTOMÁTICA PARA MODELOS DE CLASSIFICAÇÃO DE TEXTO EM LINGUAGEM NATURAL

Dissertação apresentada ao Curso de Mestrado Acadêmico em Computação do Programa de Pós-Graduação em Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em computação. Área de Concentração: Ciência da Computação

Orientador: Dr. Marcos Antônio de Oliveira

QUIXADÁ

2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S239a Santos Júnior, Valmir Oliveira dos.
Aprendizagem e rotulação de intenções semiautomática para modelos de classificação de Texto em Linguagem Natural / Valmir Oliveira dos Santos Júnior. – 2024.
68 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Campus de Quixadá, Programa de Pós-Graduação em Computação, Quixadá, 2024.
Orientação: Prof. Dr. Marcos Antônio de Oliveira.

1. Chatbots. 2. Natural Language Understanding. 3. Aprendizagem não Supervisionada. 4. Agrupamento. 5. COVID-19. I. Título.

CDD 005

VALMIR OLIVEIRA DOS SANTOS JÚNIOR

APRENDIZAGEM E ROTULAÇÃO DE INTENÇÕES SEMIAUTOMÁTICA PARA MODELOS DE CLASSIFICAÇÃO DE TEXTO EM LINGUAGEM NATURAL

Dissertação apresentada ao Curso de Mestrado Acadêmico em Computação do Programa de Pós-Graduação em Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em computação. Área de Concentração: Ciência da Computação

Aprovada em: 27 de Março de 2024

BANCA EXAMINADORA

Dr. Marcos Antônio de Oliveira (Orientador)
Universidade Federal do Ceará (UFC)

Dra. Ticiana Linhares Coelho da Silva
Universidade Federal do Ceará (UFC)

Dr. Vinícius Cezar Monteiro de Lira
Universidade Federal do Ceará (UFC)

Dedico este trabalho aos meus familiares e amigos, cujo apoio e incentivo foram fundamentais em todos os momentos desta jornada acadêmica. À minha orientação, que guiou com sabedoria e paciência, expresso minha profunda gratidão. .

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer a Deus por estar sempre comigo e me dado a benção de concluir esse trabalho com ótimos resultados. Em seguida quero expressar meus sinceros agradecimentos a todas as pessoas que contribuíram de alguma forma para a realização deste trabalho. À minha orientação, pelo suporte, direcionamento e valiosas contribuições que foram essenciais para o desenvolvimento deste estudo.

Este trabalho não seria possível sem o suporte do projeto da FUNCAP intitulado "Plataforma de Big Data para Acelerar a Transformação Digital do Estado do Ceará", que disponibilizou os recursos que tornaram a pesquisa realizável. Portanto, expresso meu agradecimento ao suporte financeiro e técnico.

Agradeço também aos meus familiares e amigos, cujo apoio constante e compreensão foram pilares fundamentais ao longo desta jornada. Agradeço aos colegas de estudo e colaboradores, pela troca de conhecimentos e experiências enriquecedoras.

Por fim, dedico meus agradecimentos a todos que, de alguma maneira, contribuíram para a concretização deste projeto. Seu apoio foi crucial e é verdadeiramente apreciado.

“Explorar as fronteiras do conhecimento é como navegar em um oceano vasto de possibilidades. Que esta jornada inspire descobertas inovadoras e ilumine os caminhos do saber.”

(Valmir Oliveira dos Santos Júnior)

RESUMO

É cada vez mais comum a utilização de chatbots como interface de serviços. Um dos principais componentes de um chatbot é o modulo de *Natural Language Understanding* (*NLU*), responsável por interpretar o texto, extrair a intenção e as entidades presentes. É possível focar apenas em uma dessas tarefas do *NLU*, como a classificação de intenções. Para treinar um modelo de classificação de intenção *NLU*, geralmente é necessário usar uma quantidade considerável de dados anotados, onde cada frase do conjunto de dados recebe um rótulo indicando uma intenção. A rotulagem manual dos dados é uma tarefa árdua que consome muito tempo, dependendo do volume de dados. Assim, uma técnica de aprendizado de máquina não supervisionada, como agrupamento de dados, poderia ser aplicada para encontrar e rotular padrões. Para esta tarefa, é essencial ter uma representação vetorial de textos eficaz que retrate as informações semânticas e ajude a máquina a compreender o contexto, a intenção e outras nuances de todo o texto. Este trabalho avalia extensivamente diferentes modelos de embeddings de texto para agrupamento e rotulagem. Também são aplicadas algumas operações para melhorar a qualidade do conjunto de dados, onde são descartadas as sentenças menos representativas de cada grupo gerado. Em seguida são treinados alguns Modelos de Classificação de Intenções com duas arquiteturas baseadas em Redes Neurais, utilizando o texto de atendimento da Plataforma do Plantão Coronavírus (PPC). Também foi anotado manualmente um conjunto de dados para ser usado como dados de validação. Foi realizado um estudo sobre a rotulagem semiautomática, implementada por meio de agrupamento de dados e inspeção visual, a qual introduziu alguns erros de rotulagem nos modelos de classificação de intenções. No entanto, seria inviável anotar todo o conjunto de dados manualmente. Contudo, ainda foram construídos modelos que obtiveram mais de 98% de acurácia com dados de testes e mais de 96% com dados de validação.

Palavras-chave: chatbots; natural language understanding; aprendizagem não supervisionada; agrupamento; COVID-19.

ABSTRACT

It is increasingly common to use chatbots as service interfaces. One of the main components of a chatbot is the *NLU* module, responsible for interpreting the text, extracting the intent, and identifying the entities present. It is possible to focus on just one of these *NLU* tasks, such as intent classification. To train an *NLU* intent classification model usually requires a considerable amount of annotated data, where each sentence in the dataset is labeled with an intent. Depending on the volume of data, manual data labeling can be laborious and time-consuming. Thus, an unsupervised machine learning technique, such as data clustering, could be applied to find and label patterns. For this task, an effective text vector representation that captures semantic information and helps the machine understand the context, intent, and other nuances of the entire text is essential.

This work extensively evaluates different text embedding models for clustering and labeling. Some operations are also applied to improve the dataset's quality, where the least representative sentences of each generated group are discarded. Then, some Intent Classification Models are trained using two architectures based on Neural Networks, using service text from PPC. A dataset was also manually annotated to be used as validation data. A study was conducted on semi-automatic labeling, implemented through data clustering and visual inspection, which introduced some labeling errors in the intent classification models. However, it would be unfeasible to manually annotate the entire dataset. Nonetheless, models were built that achieved over 98% accuracy with test data and over 96% with validation data.

Keywords: chatbots; natural language understanding; unsupervised learning; grouping; COVID-19.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

NLU	*Natural Language Understanding*
PPC	Plataforma do Plantão Coronavírus
AI	*Artificial intelligence*
ML	*Machine Learning*
NLP	*Natural Processing Language*
BOW	*Bag of Words*
WE	*Word Embeddings*
SE	*Sentence Embeddings*
USE	*Universal Sentence Encoder*
MUSE	*Multilingual Universal Sentence Encoder*
SBERT	*Sentence Bidirectional Encoder Representations from Transformers*
SVM	*Support Vector Machine*
FAQ	*Frequently Asked Questions*
DL	*Deep Learning*
OS	*Open Source*
NN	*Neural Network*
MTDNN	*Multi Task Deep Neural Network*
CNN	*Convolutional Neural Network*
RNN	*Recurrent Neural Network*
LSTM	*Long Short-Term Memory*
GRU	*Gated Recurrent Unit*
BERT	*Bidirectional Encoder Representations from Transformers*
NER	*Named Entity Recognition*
LaBSE	*Language-agnostic BERT Sentence Embedding*
DMN	*Deep Medium Network*
DNN	*Deep Neural Network*
UMAP	*Uniform Manifold Approximation and Projection*
CEP	Código de Endereçamento Postal
CPF	Cadastro de Pessoas Físicas
URL	*Uniform Resource Locator*

DBS	*Davies Bouldin Score*
SS	*Silhouette Score*
MCC	*Matthews Correlation Coefficient*

SUMÁRIO

1 INTRODUÇÃO

Atualmente é cada vez mais comum o uso de chatbots como interface de interação com os usuários. Chatbots são softwares que tentam se comportar como humanos em uma conversa. É notório o avanço desta linha de pesquisa (Xu *et al.*, 2017; Abdellatif *et al.*, 2020), permitindo que os chatbots realizem diversos tipos de serviços, desde a obtenção de informações até a realização de tarefas mais complexas como reservar um serviço de hotel, como também comprar e vender criptomoedas.

Na maioria das vezes essas interfaces apresentam um fluxo de conversação rígido a ser seguido, proporcionando uma experiência menos humanizada. Tornar as conversas com a máquina cada vez mais próximas das conversas entre humanos é um problema antigo na ciência da computação e na *Artificial intelligence* (*AI*), que remonta ao teste de Turing quando o objetivo era gerar a percepção de que a máquina seria o humano (Russell; Norvig, 2004).

Uma das maneiras de tornar o diálogo entre humanos e bots mais flexível, seria permitir ao humano um bom número de formas de se expressar a uma máquina, indicando um maior poder de generalização da máquina na interpretação de texto. Onde a máquina teria a capacidade de extrair intenções e entidades contidas nos textos recebidos como entrada. Para isso *Machine Learning* (*ML*) e *Natural Processing Language* (*NLP*) oferecem excelentes resultados, desde que se tenha muitas sentenças para treinamento de um modelo (Russell; Norvig, 2004). Ontologias também são empregadas para conferir generalidade à conversação, onde os bots buscam informações sobre domínios de conhecimento que podem surgir no diálogo em um modelo conceitual que relaciona os conceitos de um domínio.

Um componente essencial nesse tipo de chatbot é o modelo de *NLU*. Este componente é responsável por interpretar as informações fornecidas como entrada pelo usuário, gerando dados que o chatbot é capaz de entender e interpretar. Em geral, essas informações não são estruturadas, o modelo *NLU* identifica a intenção do usuário e extrai entidades específicas do domínio. Mais especificamente, uma intenção representa um mapeamento entre o que um usuário diz e qual ação deve ser executada pelo chatbot. As ações correspondem às etapas que o chatbot executará quando o usuário ativar intenções específicas. Uma entidade é o que ou quem é falado na entrada do usuário (Adamopoulou; Moussiades, 2020). Por exemplo, considere a frase "Quais são os sintomas da COVID-19?". O usuário pretende saber quais são os sintomas da doença COVID-19. A entidade dessa intenção seria COVID-19, que se refere a uma entidade do tipo doença.

No entanto, treinar o módulo *NLU* não é uma tarefa fácil. Uma grande quantidade de dados anotados é necessária para treinar o modelo e obter resultados satisfatórios. Frequentemente, os dados são anotados manualmente, o que leva muito tempo para ser realizado. Uma alternativa para ajudar o processo de anotação de dados é aplicar técnicas de aprendizagem não supervisionada para descobrir padrões ou relacionamentos entre os dados, é possível usar alguma técnica ou ferramenta que realize o agrupamento dos dados onde o objetivo é encontrar grupos mais informativos, que juntos podem representar uma classe (Monard; Baranauskas, 2003). Para realizar o agrupamento é possível aplicar um algoritmo de clusterização como K-means, onde o conjunto de dados é agrupado em *clusters*. Outra alternativa seria aplicar outra técnica de clusterização baseado em *topic modeling* como o é o caso do BERTopic[1], que atribui a probabilidade de cada item pertencer a um tópico, a partir do conjunto de dados. Vale ressaltar que os *clusters* gerados na clusterização e os tópicos gerados pelo BERTopic, representam um grupo dentro do conjunto de dados, sendo assim por generalização neste trabalho, quando for citado grupos, leve em consideração que pode estar se referindo a cluster ou tópico, ou os dois. Ambas as técnicas são exploradas neste estudo, onde o objetivo é aplicá-las em conjunto de diálogos existentes no intuito de identificar as intenções representadas por cada grupo gerado. Em seguida, essas intenções podem servir como entradas para treinar o modelo *NLU*.

O processo de agrupamento de textos tipicamente requer uma representação por meio de vetores numéricos de comprimento fixo, desses dados que serão usados como entrada. Uma forma comum de representação seria usar *Bag of Words* (*BOW*) (Harris, 1954) devido à simplicidade, no entanto, essa técnica apresenta algumas desvantagens, como esparsidade e alta dimensionalidade. Por outro lado, *Word Embeddings* (*WE*) pré-treinadas têm sido amplamente utilizados (Mikolov *et al.*, 2013; Pennington *et al.*, 2014), devido à capacidade de captar o contexto de uma palavra em um documento, semelhança semântica e sintática em relação as demais palavras. No entanto, caso ocorram pequenas mudanças entre uma sentença e outra (como adicionar uma negação, por exemplo), as *WE* podem não captar a mudança de significado com efetividade. Por exemplo, as sentenças *Pedro está sentindo dor de cabeça* e *Pedro não está sentindo dor de cabeça* têm significados opostos, apesar da diferença de apenas uma palavra. Uma alternativa para essa limitação é utilizar *Sentence Embeddings* (*SE*). Existem muitos modelos de *SE*, tais como *Universal Sentence Encoder* (*USE*) (Cer *et al.*, 2018), *Multilingual Universal Sentence Encoder* (*MUSE*) (Yang *et al.*, 2019), *Sentence Bidirectional Encoder Representations*

[1] https://maartengr.github.io/BERTopic

from Transformers (*SBERT*) (Reimers; Gurevych, 2019) e Doc2Vec (Le; Mikolov, 2014). De uma maneira geral, esses modelos recebem como entrada o texto (que pode ter tamanho variável), e produzem como saída um único vetor de tamanho fixo. *SE* representam frases inteiras e suas informações semânticas por meio de vetores numéricos. Isso ajuda a máquina a compreender o contexto, a intenção e outras nuances de todo o texto. Os dados disponibilizados para treinamento têm o impacto mais significativo nos vetores de *SE*. Para obter resultados ideais, as sentenças do conjunto de treinamento devem ser semanticamente relacionadas (Kiros *et al.*, 2015).

Este trabalho amplia os experimentos apresentados em (Junior *et al.*, 2021). Onde foram construídos vários modelos *NLU* usando o Rasa Framework com diferentes modelos de embeddings usando o mesmo conjunto de dados descrito neste estudo. Em (Junior *et al.*, 2021), o processo de agrupamento foi realizado apenas com a clusterização a qual ainda era mais simples, onde o valor de K era igual a 10. Porém, neste trabalho, esse limite foi aumentado, o melhor valor de K foi selecionado conforme as métricas de clusterização e foram aplicados diferentes abordagens de descarte das sentenças menos representativas nos clusters. Portanto, os experimentos neste novo estudo abordam a questão do treinamento de um modelo de classificação de intenção sem dados anotados. Dado um conjunto de dados de diálogos, como entrada não anotada, pretende-se propor um modelo de classificação de intenção para criar um chatbot. Semelhante ao que é apresentado em (Peikari *et al.*, 2018), onde eles primeiro aplicam um método de aprendizagem não supervisionado (clusterização), para encontrar um padrão nos dados, e então usam esses padrões (rótulos) para treinar um modelo *Support Vector Machine* (*SVM*), para apoiá-los na tomada de decisões. O caso de uso deste estudo é baseado em diálogos de atendimento da COVID-19, que compreendem os diálogos de aconselhamento de profissionais de saúde realizados na PPC[2] no estado do Ceará, Brasil. Além dessas diferenças, também foram estudados diferentes modelos de representação de embeddings, o erro de rotulagem adicionados a partir do processamento de anotações, foi melhorado a seção de trabalhos relacionados e realizados mais experimentos em comparação com o artigo anterior (Junior *et al.*, 2021).

As principais questões de pesquisa que norteiam este estudo são:

– **(RQ1)** A partir de um enorme conjunto de dados de conversas, como rotular intenções usando aprendizagem não supervisionada para diálogos com frases curtas, sem caracterizar perguntas e respostas ao longo da conversa?

– **(RQ2)** Como criar um modelo *NLU* para classificação de intenções usando os dados

[2] https://coronavirus.ceara.gov.br

rotulados semi automaticamente de **(RQ1)**?

– **(RQ3)** A representação de embedding de textos usada para a etapa de agrupamento e rotulagem poderia auxiliar no treinamento de um classificador de intenção?

– **(RQ4)** Como a fase de agrupamento é uma técnica não supervisionada, esta etapa adiciona erro de rotulagem para o classificador de intenção *NLU*?

As contribuições deste trabalho consistem nos seguintes itens:

– Avaliar diferentes estratégias de embedding de sentenças e de palavras para o problema de descoberta de intenções em diálogos sobre COVID-19. Foi utilizado uma variedade de métodos de embedding, incluindo o Glove, que oferece embeddings estáticos derivados de estatísticas de coocorrência de palavras. Além disso, foi aplicado modelos de embeddings contextualizadas, principalmente baseadas em BERT, para capturar nuances contextuais nos textos e lidar com desafios de escassez de dados.

– Propor uma abordagem não supervisionada (K-means), para lidar com a necessidade de anotar um conjunto de dados de conversação com rótulos de intenção.

– Propor diferentes abordagens para lidar com *outliers* adicionados devido ao processo não supervisionado adotado no processo de clusterização para rotulagem de intenções.

– Investigar o erro potencial incluído pela rotulagem semiautomática.

– Avaliar como diferentes representações de embeddings impactam nos modelos de classificação de intenções

– Ao longo do curso dessa pesquisa foi realizado a publicação do artigo *A natural language understanding model covid-19 based for chatbots.* em 2021 na *21st International Conference on Bioinformatics and Bioengineering do IEEE* (**BIBE**) com *Qualis* **B1**

– Foi realizado a submissão de outro artigo intitulado *Learning and Semiautomatic Intention Labeling for Classification Models: A COVID-19 Dialogue Attendance Study for Chatbots* para o *Journal Natural Language Processing* (**NLP**) com o Qualis **A1** O qual foi aprovado após 5 revisões e será publicado até o final do ano de 2024.

Em termos de classificação de intenções, foram consideradas duas arquiteturas: uma rede neural baseada em *feedforward* simples e o modulo *NLU* Rasa. O conjunto de dados experimental consiste em 1.237 diálogos do PCS, coletados entre 1 º de maio de 2020 e 6 de maio de 2020, com 26.754 frases de pacientes e 26.992 frases de profissionais de saúde, todas anotadas com seus respectivos atores. Para realizar a construção dos modelos de classificação de intenções foram usadas apenas as 26.754 sentenças ditas pelos pacientes.

O restante deste trabalho está organizado da seguinte forma: o Capítulo 2 explica os conceitos preliminares necessários para a compreensão deste trabalho, o Capítulo 3 apresenta alguns trabalhos relacionados. O Capítulo 4 descreve a arquitetura proposta para Classificação Automática de Intenções e discute os dados e métodos para atingir os objetivos principais. O Capítulo 5 apresenta os experimentos e suas análises. Finalmente, o Capítulo 6 fornece uma discussão sobre este trabalho, suas limitações, resume este trabalho e propõe desenvolvimentos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo serão abordados conceitos e fundamentos relacionados ao trabalho. Na Seção 2.1, é falado sobre a definição do que são os chatbots, e como eles podem ser classificados. Na Seção 2.2 será discutido sobre os modelos de Redes Neurais, sobre a estrutura, seus componentes a forma de construção e sua aplicabilidade. Na Seção 2.3 é abordado sobre o que são os modelos de *NLU*, suas funcionalidades, sua importância e como esse modelo pode ser construído. Na Seção 2.4 é abordado sobre os modelos de classificação de intenções, seus objetivos e as dificuldades encontradas na construção e sua importância no processo de encontrar as intenções do que é expresso por meio de fala ou digitação. Na Seção 2.5 é abordado sobre o que é a técnica de Clusterização, sobre algorítimos que podem ser usados para realizar esse processo, e como é possível realizar essa tarefa sobre um conjunto de dados textuais. Na seção 2.6 é discutido sobre diferentes tipos de embeddings os *WE* e os *SE*. E por último na Seção 2.7 é explicado sobre as ferramentas Rasa e BERTopic, que foram utilizadas nos experimentos desse trabalho.

2.1 Chatbots

Os chatbots são sistemas de *AI* (Adamopoulou; Moussiades, 2020), que possuem a capacidade de responder como sendo uma entidade inteligente quando participam de uma conversa por meio de voz ou texto, e conseguem interagir por mais de um idioma utilizando técnicas de *NLP* (Khanna *et al.*, 2015). Esses sistemas além de serem capazes de imitar a conversa humana e entreter os usuários, podem ser utilizados em aplicações como educação, recuperação de informações, negócios e comércio eletrônico (Shawar; Atwell, 2007).

Eles podem ser classificados levando em consideração diferentes aspectos, relacionados ao **domínio de conhecimento**, o **serviço oferecido,** os **objetivos**, a **forma de processamento das entradas e métodos de geração de respostas** , a **ajuda humana** e **métodos de construção**(Adamopoulou; Moussiades, 2020).

Quanto a classificação baseada no**domínio de conhecimento** podem ser de**domínio aberto** onde conseguem falar sobre tópicos gerais e responder adequadamente, enquanto os chatbots de **domínio fechado** estão focados em um domínio de conhecimento específico e podem não responder a outras perguntas (Nimavat; Champaneria, 2017).

Em relação ao **Serviço oferecido** pode ser classificado em **Interpessoal**, que estão

relacionados ao domínio da comunicação, serviços como reserva de restaurante, reserva de voos e chatbots de *Frequently Asked Questions* (*FAQ*). Em **Intrapessoais**, os quais geralmente realizam tarefas que são do domínio pessoal do usuário como gerenciamento de calendário, armazenamento de opinião de usuário, semelhante ao que um humano faz. E ainda pode ser classificado em **Interagente**, os quais são predominantes em áreas como Internet das coisas eles se comunicam entre si para realizar determinada tarefa (Nimavat; Champaneria, 2017).

Outra classificação é em relação aos objetivos do chatbot. Que são os **Informativos** cujo objetivo é prover informações que foram armazenadas anteriormente ou que estejam disponíveis em alguma fonte de dados fixa, como, por exemplo, chatbots de *FAQ*. Os de **Bate-Papo ou Conversação**, são responsáveis por se comunicarem com os usuários como se fossem seres humanos, objetivando responder corretamente as sentenças que lhe são dadas, são geralmente construídos usando técnicas como interrogatório cruzado, evasão e deferência. E os **Baseado em Tarefas**, atuam realizando uma tarefa bem definida como, por exemplo, reservar um voo, realizar a reserva em um hotel, são inteligentes, capazes de pedir informações pertinentes a tarefa a ser realizada (Nimavat; Champaneria, 2017).

Baseado na **forma de processamento das entradas e métodos de geração de respostas**, podem ser classificados em **Modelo baseado em regras**, os quais escolhem a resposta do sistema com base em um conjunto de regras pré-definidas fixas. **Modelo baseado em recuperação**, oferece um pouco mais de flexibilidade em relação ao anterior, por consultar e analisar recursos disponíveis usando APIs. **Modelo generativo** apresenta melhor geração de respostas em relação aos outros tipos de modelos, porque leva em consideração a mensagem atual e as anteriores, eles usam *ML* e técnicas de *Deep Learning* (*DL*), contudo apresenta dificuldades no processo de construção e treinamento (Adamopoulou; Moussiades, 2020; Hien *et al.*, 2018).

Os chatbots auxiliados por humanos usam computação humana em pelo menos um elemento do chatbot. Nessa abordagem humanos podem preencher as lacunas causadas pelas limitações dos chatbots totalmente automatizados. Embora a computação humana, em comparação com algoritmos baseados em regras e aprendizado de máquina, forneça mais flexibilidade e robustez, ainda assim, ela não pode processar uma determinada informação tão rápido quanto uma máquina, o que dificulta a escala para mais solicitações do usuário (Adamopoulou; Moussiades, 2020; Kucherbaev *et al.*, 2018).

Chatbots podem ainda ser classificados conforme as permissões providas pela plataforma na qual foram desenvolvidos. Podem ser plataformas de *Open Source* (*OS*) como Rasa,

que permite o design dos mais variados aspectos da implementação. E as Plataformas Fechadas, como Google ou IBM, esses geralmente agem como uma caixa preta, de forma que não se tenha muita autonomia de design em certos aspectos, o que pode ser uma desvantagem dependendo dos requisitos do projeto (Adamopoulou; Moussiades, 2020).

O Rasa é um framework de aprendizado de máquina de código aberto para conversas automatizadas de texto e voz. Responsável por entender mensagens, manter conversas e conectar-se a canais de mensagens e APIs. O módulo de *NLU* do Rasa trabalha com um pipeline de componentes para treinar um modelo capaz de extrair intenções e entidades de um texto. Para isso é necessário usar como entrada um conjunto de dados anotado com as respectivas entidades e intenções. O Rasa também fornece ferramentas para testar o desempenho do modelo *NLU*. O pipeline pode ser customizado conforme as necessidades do modelo e possibilita o ajuste do conjunto de dados. Embeddings de palavras pré-treinadas podem estar presentes no pipeline, adicionando versatilidade ao modelo treinado. Cada componente processa a entrada e/ou cria uma saída. A saída de um componente pode ser usada por qualquer outro componente que venha depois no pipeline. O Rasa fornece muitos modelos pré-treinados para diferentes idiomas, incluindo BERT e GPT[1] (Rasa, 2022).

Para a construção do modelo de classificação de intenção neste trabalho, utilizamos os dados de domínio fechado da Plataforma Internet do Serviço Coronavírus (PCS) no Ceará, Brasil. São pacientes suspeitos de estarem infectados pela doença COVID-19. Para isso foram utilizadas duas arquiteturas, uma com o framework Rasa e a outra utiliza redes neurais.

O modelo de classificação de intenções proposto pode ser utilizado em um chatbot interpessoal orientado a tarefas, permitindo a interação com os pacientes fazendo perguntas sobre como os pacientes estão se sentindo, pretendendo gerar respostas e propondo ações a serem tomadas por eles, por exemplo, procurar ajuda médica ao apresentar sintomas específicos relacionados à COVID-19.

A próxima seção discute outro conceito relevante para este trabalho: Modelos de redes neurais.

2.2 Modelos de Rede Neurais

Recentemente, as *Neural Network* (*NN*), especificamente as de aprendizagem profunda, ganharam enorme espaço na área de pesquisa de Reconhecimento de Fala, apresentando

[1] veja https://huggingface.co/models para obter uma lista completa de modelos disponíveis

resultados melhores que os métodos tradicionais (Nassif *et al.*, 2019). O aprendizado profundo consiste em um algoritmo de *ML* cuja entrada são modelos multicamadas. *NN* com diferentes níveis de operações não lineares formam algoritmos de *ML* que extraem características e informações específicas dos dados (Nassif *et al.*, 2019). Uma *NN* típica compreende camadas de neurônios que possuem funções de ativação e são conectadas por meio de pesos que são ajustados segundo a entrada de dados e um algoritmo de retro propagação (Liu *et al.*, 2017). Em algoritmos de aprendizado profundo, as camadas são pré-treinadas sem supervisão e, após serem conectadas para treinamento e refinamento supervisionados, (Liu *et al.*, 2017).

Em (Zhou *et al.*, 2020), encontramos o progresso mais recente na aplicação de *NN* em problemas de *NLP*. Eles classificam a estrutura neural da *NLP* como: “modelagem destinada a projetar estruturas de rede apropriadas para diferentes tarefas; aprendizado voltado para otimizar os parâmetros do modelo e raciocínio voltado para gerar respostas a questões invisíveis por meio da manipulação do conhecimento existente com técnicas de inferência"(Zhou *et al.*, 2020).

A seguir, discutimos sobre o que são os modelos de *NLU*.

2.3 Modelos de Entendimento de Linguagem Natural

Um modelo de *NLU* é responsável por interpretar as informações fornecidas como entrada pelo usuário, e gerar uma saída, que um chatbot consiga usar para entender e interpretar a entrada do usuário. Em geral, as entradas do usuário não são estruturadas.

Esse modelo identifica as intenções do usuário e extrai entidades específicas do domínio. Mais especificamente, a intenção resume o objetivo da frase de entrada do usuário e é usada como um mapeamento entre o que o usuário diz e qual ação deve ser executada pelo chatbot. As ações correspondem às etapas que o chatbot executará quando o usuário ativar intenções específicas. Uma entidade é o que ou quem é falado na entrada do usuário (Adamopoulou; Moussiades, 2020).

Uma das tarefas fundamentais em *NLU* é aprender representações de texto em espaço vetorial. Existem duas abordagens populares: aprendizagem multitarefa e pré-treinamento do modelo de linguagem. Essas técnicas são combinadas na proposta de uma *Multi Task Deep Neural Network* (MTDNN). A aprendizagem humana inspirou esta abordagem no sentido de que muitas vezes eles aplicam o conhecimento aprendido em tarefas anteriores para realizar uma nova tarefa (Liu *et al.*, 2019) e também, usar tarefas que simultaneamente podem se beneficiar de outras habilidades aprendidas.

(Gao *et al.*, 2018) apresenta uma pesquisa com métodos que demonstram como os pipelines de tarefas sequenciais são aplicados para alcançar o *NLU* usando o pré-treinamento do modelo de linguagem. Para aplicar um modelo pré-treinado a tarefas específicas de *NLU*, muitas vezes é necessário ajustá-lo, para cada tarefa, com camadas adicionais específicas de tarefa usando dados de treinamento específicos de tarefa. (Liu *et al.*, 2019) defende que a aprendizagem multitarefa e o pré-treinamento do modelo de linguagem são tecnologias complementares, possibilitando sua combinação para melhorar a aprendizagem das representações textuais, aumentando o desempenho das tarefas do *NLU*.

Normalmente, Acurácia e F1-score são as métricas usadas para avaliar a qualidade de predição de um modelo. *NLU* é uma etapa de pré-processamento para módulos posteriores em um sistema de chatbot, e seu desempenho interfere diretamente na qualidade geral do chatbot (Gao *et al.*, 2018). A classificação multi classe usando redes neurais é comum na literatura recente, e essa técnica é usada especialmente para as tarefas de classificação de domínio e intenção. Para frases curtas, onde o contexto é necessário para inferir informações, redes neurais recorrentes e convolucionais são aplicadas porque consideram o texto antes do enunciado atual. (Lee; Dernoncourt, 2016).

Quanto ao preenchimento de *slots* ou identificação de entidades, muitas vezes é usada a classificação de sequência (Gao *et al.*, 2018). Nesta abordagem, o classificador prevê rótulos de classe semântica para subsequências do enunciado de entrada (Wang *et al.*, 2005). Redes neurais recorrentes são aplicadas para esta tarefa, oferecendo bons resultados (Yao *et al.*, 2013).

2.4 Modelos de Classificação de intenções

Determinar a intenção de uma frase falada ou digitada durante uma conversa é crucial para a tarefa de *NLU*. Normalmente, um modelo de aprendizagem supervisionada é utilizado para classificar as intenções, que representa o objetivo da frase. A tarefa a seguir é preencher espaços que representam informações semânticas que ajudam a preencher a intenção da mensagem (Weld *et al.*, 2021). Essas duas tarefas podem ser executadas durante o processo de *NLU* sequencialmente em um pipeline ou simultaneamente em pesquisas recentes. O *NLU* é vital para uma interface linguística com os humanos. Tecnologias como agentes de conversação, chatbots, Internet das Coisas e assistentes virtuais, entre outras, precisam fazer um bom trabalho de reconhecimento de intenções e a ciência está procurando melhorar cada vez mais nesse sector.

As *WE* e o aprendizado profundo estão entre as tecnologias recentes empregadas no processo de *NLU*, alcançando resultados muito promissores e adaptando-se melhor à interface inteligente com humanos, não apenas em texto, mas em voz e em breve em vídeos e imagens (Weld *et al.*, 2021; Liu *et al.*, 2019).

Algumas dificuldades na detecção de intenções estão listadas em (Liu *et al.*, 2019): Falta de fontes de dados, irregularidade na expressão do usuário, detecção de intenções implícitas e detecção de múltiplas intenções. Os métodos para detecção de intenções podem ser tradicionais, como reconhecimento semântico de modelo baseado em regras ou algoritmos de classificação baseados em recursos estatísticos. Os métodos comuns incluem Naive Bayes, *SVM* e regressão logística. Os métodos atuais de última geração incluem representação de texto via embedding, *Convolutional Neural Network* (*CNN*), *Recurrent Neural Network* (RNN), *Long Short-Term Memory* (*LSTM*), *Gated Recurrent Unit* (*GRU*), mecanismo de atenção, e *Capsule Networks*. Esses modelos de aprendizado profundo apresentam grande melhoria no desempenho de detecção (Liu *et al.*, 2019).

2.5 Clusterização

A clusterização é uma classe de métodos de aprendizado de máquina não supervisionados em que o objetivo principal é particionar um conjunto de pontos de dados em grupos tão semelhantes quanto possível (Aggarwal; Reddy, 2014).

Na aplicação de clusterização em problemas de *NLP*, busca-se identificar padrões semelhantes em dados textuais. Este enfoque é fundamental para tarefas como agrupamento de documentos, segmentação de tópicos e análise de sentimentos.

Diversos algoritmos de clusterização estão disponíveis na literatura. Entre eles, destaca-se o K-means, um método iterativo que encontra k centróides para os k clusters, agrupando cada elemento do conjunto de dados com base na proximidade ao centróide mais próximo (Vassilvitskii; Arthur, 2006). É importante mencionar que o K-means é sensível a outliers, apesar de sua eficiência computacional. Alternativamente, outros métodos, como o DBSCAN (Ester *et al.*, 1996), baseado em densidade, ou o agrupamento hierárquico CLINK (Defays, 1977), oferecem abordagens distintas.

Neste trabalho, foi optado por utilizar o algoritmo K-means, em virtude de sua simplicidade. A implementação escolhida foi a do pacote *Fast Pytorch Kmeans*[2], que utiliza

[2] https://github.com/DeMoriarty/fast_pytorch_kmeans

PyTorch, permitindo um desempenho otimizado em ambientes com GPU. No entanto, ressalta-se que a abordagem proposta neste artigo é flexível e pode ser adaptada para qualquer algoritmo de clusterização.

Para realizar uma clusterização de dados textuais é necessário obter suas representações numéricas para cada elemento, do conjunto a ser clusterizado. Para isso é possível usar modelos de embeddings, na geração dos vetores numéricos que representam os dados textuais.

2.6 Embeddings

Os modelos de Embeddings transformam textos de tamanhos diferentes (como uma palavra, frase, parágrafo ou documento) em um vetor numérico de tamanho fixo a ser alimentado em aplicativos *downstream*, como detecção de similaridade ou modelos de *ML*. *BOW* é uma representação comum (Harris, 1954) devido à simplicidade. No entanto, essas técnicas apresentam algumas desvantagens, como vetores com alta dimensionalidade.

Nas Subseções a seguir, será abordado sobre as *WE* e *SE*, quais são as principais características de cada modelo, além de apresentar algumas propostas de embeddings amplamente usadas.

2.6.1 Embeddings de Palavras

WE pré-treinados têm sido amplamente usados (Mikolov *et al.*, 2013; Pennington *et al.*, 2014; Akbik *et al.*, 2019; Souza *et al.*, 2020), devido à sua capacidade de capturar o contexto de uma palavra em um documento, semelhança semântica e sintática com outras palavras.

(Mikolov *et al.*, 2013) propõe uma estrutura para aprender os vetores de palavras treinando um modelo de linguagem que prevê uma palavra dada as outras palavras em um contexto. Uma implementação particular de tal estrutura é word2vec. A principal desvantagem é que (Mikolov *et al.*, 2013) utiliza mal as estatísticas do corpus, uma vez que o modelo é treinado em uma janela de contexto local separada, em vez de contagens de coocorrência global. (Pennington *et al.*, 2014) contorna esse problema e propõe um modelo que produz um espaço vetorial de palavras. (Pennington *et al.*, 2014) treina o modelo em contagens globais de coocorrência palavra-palavra e faz uso eficiente de estatísticas. Outra proposta de *WE* é o FLAIR (Akbik *et al.*, 2019) que abstrai desafios específicos de engenharia que diferentes tipos de embedding de palavras acrescentam. FLAIR cria uma interface unificada para todos os *WE*,

SE e combinações arbitrárias de embeddings.

BERTimbau (Souza *et al.*, 2020) fornece modelos *Bidirectional Encoder Representations from Transformers* (BERT) para o português brasileiro. Os modelos foram avaliados em três tarefas de *NLP*: similaridade textual de sentenças, reconhecimento de vinculação textual e *Named Entity Recognition* (*NER*). BERTimbau melhora o estado da arte nessas tarefas sobre o BERT multilíngue e abordagens monolíngues anteriores, confirmando a eficácia de grandes modelos de linguagem pré-treinados para o português.

A partir de *WE*, pode-se obter vetores de sentenças calculando a média de todos os vetores gerados de cada palavra na sentença. No entanto, esse procedimento dá o mesmo peso a palavras importantes e não importantes. Outra limitação de representar texto usando *WE* é que cada palavra seria representada com o mesmo vetor, independentemente do contexto. Considere a frase "*Paris Hilton viajou dos Estados Unidos para Paris.*"Por exemplo, a palavra *Paris* dependendo do contexto, pode ser o primeiro nome de uma pessoa, como, (**Paris** Hilton) ou uma localização, quando representa, por exemplo, a capital da França, **Paris**. Uma extensão das *WE* são as *SE*, usadas para obter, representações das sentenças diretamente.

2.6.2 Embedding de Sentenças

As *SE* representam frases em um espaço vetorial n-dimensional de forma que palavras semanticamente semelhantes ou semanticamente relacionadas se reúnam no método de treinamento. Esse tipo de embedding consegue realizar a representação de uma frase, que pode ter diferentes representações de uma palavra com base em seu contexto.

Existem muitas propostas para *SE* como InferSent (Conneau *et al.*, 2017), *Language-agnostic BERT Sentence Embedding* (*LaBSE*) (Feng *et al.*, 2020), *USE* (Cer *et al.*, 2018), Doc2Vec (Le; Mikolov, 2014), entre outros. *USE* propõe dois encoders diferentes, um faz uso da arquitetura de transformers (Vaswani *et al.*, 2017) e atinge o melhor desempenho. O mecanismo de atenção calcula representações cientes do contexto de palavras em uma frase que leva em consideração tanto a ordem e a identidade de todas as outras palavras. As representações de palavras sensíveis ao contexto são convertidas em um vetor de codificação de sentenças de comprimento fixo calculando a soma dos elementos das representações em cada posição da palavra (Cer *et al.*, 2018). Para mais detalhes sobre o mecanismo de atenção, é recomendado a leitura de (Galassi *et al.*, 2020) O outro *encoder* proposto é baseado em *Deep Medium Network* (*DMN*) (Iyyer *et al.*, 2015) em que *WE* são usadas como entrada e bi-gramas são calculados

primeiro juntos e depois passados por uma *Deep Neural Network* (*DNN*) *Feedforward* para produzir *SE*. (Yang *et al.*, 2019) estende (Cer *et al.*, 2018) propondo o *MUSE*, um modelo de *SE* para dezesseis idiomas em um único espaço semântico usando um codificador duplo treinado para múltiplas tarefas.

Semelhante ao Word2Vec, o Doc2Vec treina as *SE* na tarefa de previsão da próxima palavra, com base no contexto presente nas sentenças. O vetor de sentença e os vetores de palavras são concatenados para prever a próxima palavra em um contexto.

O *LaBSE* é proposto em (Feng *et al.*, 2020), ele é treinado e otimizado para embeddings multilínguas ao nível de sentenças. Produz representações semelhantes exclusivamente para pares de frases bilíngues que são traduções um do outro. O *LaBSE* emprega um codificador duplo em que as sentenças de origem e de destino são codificadas separadamente usando um codificador compartilhado baseado em BERT e em seguida alimenta uma função de combinação. As representações da camada final são consideradas a *SE* para cada entrada. A similaridade entre as sentenças fonte e alvo é pontuada usando cosseno sobre a *SE* produzida pelos codificadores BERT.

2.7 Ferramentas

Nessa Seção será abordado sobre as ferramentas utilizadas na pesquisa. Na Subseção: 2.7.1, será explicado sobre o Rasa, qual seu objetivo o que proporciona, sobre sua arquitetura e seus componentes. Na Subseção 2.7.2, é discutido sobre a ferramenta BERTopic, seus aspectos e características relacionadas ao funcionamento, objetivos e aplicabilidade.

2.7.1 Rasa

O Rasa é uma de aprendizado de máquina de código aberto para conversas baseadas em voz ou texto. Entende mensagens, mantém conversas e conecta-se a canais de mensagens e APIs. O módulo de *NLU* do Rasa trabalha com um pipeline de componentes para treinar um modelo capaz de extrair intenções e entidades de texto bruto usando como entrada um conjunto de dados anotado. O Rasa também fornece ferramentas para testar o desempenho do modelo *NLU*. O pipeline pode ser personalizado de acordo com as necessidades do modelo e possibilita o ajuste fino do conjunto de dados. Embeddings de palavras pré-treinados podem estar presentes no pipeline, adicionando versatilidade ao modelo treinado. Cada componente processa a entrada

e/ou cria uma saída. A saída de um componente pode ser usada por qualquer outro componente que vier depois do pipeline. Rasa fornece um número significativo de modelos pré-treinados para diferentes idiomas, incluindo BERT e GPT[3] (Rasa, 2022).

2.7.2 BERTopic

O BERTopic é uma ferramenta que proporciona a extração de tópicos em textos (Gensim, 2022). Foi desenvolvido com base na arquitetura BERT (Devlin *et al.*, 2018), aproveita os benefícios do aprendizado profundo para prover uma compreensão mais refinada e contextualizada dos conjunto de dados textuais.

O processo de extração de tópicos com o BERTopic envolve a representação semântica das palavras e frases, permitindo uma análise mais aprofundada das relações semânticas e contextuais entre os elementos do texto (Devlin *et al.*, 2018). Diferentemente de abordagens tradicionais, que muitas vezes utilizam métodos estatísticos ou heurísticos, o BERTopic destaca-se ao capturar nuances semânticas e contextuais, proporcionando uma visão mais precisa e abrangente dos temas presentes no corpus analisado (Grootendorst, 2022).

A aplicação do BERTopic na análise de tópicos apresenta vantagens significativas em relação a técnicas convencionais. A capacidade do BERT de compreender o contexto das palavras em uma sentença, considerando tanto as palavras anteriores quanto as subsequentes, contribui para uma representação semântica mais rica (Devlin *et al.*, 2018). Isso se reflete na qualidade dos tópicos identificados, fornecendo uma compreensão mais refinada dos conteúdos presentes nos textos.

A implementação do BERTopic inicia-se com o pré-treinamento de um modelo BERT em extensas quantidades de dados textuais, permitindo que o modelo assimile padrões complexos e adquira representações semânticas aprimoradas das palavras (Devlin *et al.*, 2018). Posteriormente, o BERTopic realiza a extração de tópicos por meio da aplicação de técnicas de redução de dimensionalidade, como o *Uniform Manifold Approximation and Projection* (UMAP), que preserva as relações semânticas entre as palavras (McInnes *et al.*, 2018)

Além disso, o BERTopic permite lidar de maneira eficiente com grandes volumes de dados textuais, sendo uma ferramenta escalável para análise de conjuntos extensos de documentos (Gensim, 2022). Utilizando métodos de clusterização para agrupar palavras semanticamente semelhantes em tópicos distintos (Koren; Carmel, 2003). Essa abordagem revela-se crucial para

[3] consulte https://huggingface.co/models para uma lista completa dos modelos disponíveis

uma análise mais profunda dos temas presentes no corpus, proporcionando insights valiosos sobre a estrutura semântica subjacente ao texto.

Essa combinação de pré-treinamento, redução de dimensionalidade e clusterização confere ao BERTopic a capacidade de extrair de maneira eficaz tópicos significativos e relevantes em textos diversos, potencializando sua utilidade em aplicações, como modelos de classificação de intenções em conjuntos de dados específicos, como o utilizado neste estudo sobre a COVID-19.

3 TRABALHOS RELACIONADOS

Neste Capitulo serão apresentados os estudos relacionados a este trabalho relacionados ao desenvolvimento de métodos e ferramentas envolvendo *NLP* no contexto acerca da COVID-19, que fazem parte do contexto ao qual esse trabalho se localiza.

O trabalho (Lei *et al.*, 2021) treina um modelo *NER* usando artigos científicos extraídos do conjunto de dados de pesquisa aberta COVID-19, CORD-19 (Wang *et al.*, 2020). As propostas dos artigos, extraem entidades utilizáveis para identificar sintomas nas frases escritas dos pacientes. Eles usam nuvens de palavras para encontrar os sintomas mais frequentes citados nos artigos, e o modelo *NLU* do chatbot é usado para construir um gráfico de conhecimento que ajuda a realizar o acompanhamento dos pacientes que retornam.

(Fazzinga *et al.*, 2021) aplica linguagem natural e gráficos de argumentação para construir sistemas de diálogo que explicam por que um chatbot deu conselhos específicos sobre a vacinação contra a COVID-19. Em (Miner *et al.*, 2020), os autores levantam questões e problemas que um chatbot poderia resolver durante uma pandemia como a COVID-19. Iniciativas como Clara[1] do CDC nos Estados Unidos vêm para lidar com a disseminação de informações conflitantes causada pela falta de conhecimento e notícias falsas que, em última análise, podem tornar muito mais difícil lidar com a situação pandémica.

Devido à pandemia da COVID-19, muitas escolas não conseguiram se preparar o suficiente para exigir atividades em ambiente virtual, (Gaglo *et al.*, 2021) construiu um chatbot para auxiliar na tutoria de alunos de uma escola secundária no Senegal. O chatbot foi construído utilizando o framework Rasa e integrado como plugin do ambiente virtual de aprendizagem Moodle. O chatbot faz algumas perguntas aos alunos traçando um perfil para eles, e com isso, poderá propor conteúdos para seus estudos. Vale ressaltar que os professores puderam consultar todo o diálogo entre o bot e os alunos. Isso permite a análise dos professores sobre a tutoria do bot, para poderem medir a qualidade do processo de aprendizagem e intervir com ações para melhorá-lo.

(Klein *et al.*, 2021) constrói um classificador usando redes neurais profundas baseado em um modelo BERT pré-treinado com tweets relacionados ao COVID-19. Eles coletaram tweets da interface de programação do aplicativo Twitter Streaming que mencionam palavras-chave relacionadas ao COVID-19. Eles aplicaram expressões regulares para identificar os tweets indicando se o usuário teria sido exposto ao COVID-19. O modelo treinado pode detectar tweets

[1] https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/testing.html

que relatam casos potenciais de COVID-19.

(Li *et al.*, 2020a) apresenta o EmoCT (Emotion-Covid19-Tweet); para isso, selecionaram aleatoriamente 1.000 tweets e classificaram cada um nas seguintes emoções: raiva, aceitação, nojo, medo, felicidade, tristeza, surpresa e confiança. Eles usaram esse conjunto de dados para treinar modelos de classificação usando *NLP* e aplicaram o modelo de incorporação BERT para representar os tweets.

O trabalho (Aguiar *et al.*, 2022) aplica aumento de dados para aumentar dados de treinamento para seleção de respostas em chatbots com base na recuperação multiturno. Eles aplicam a tradução automática de um conjunto de dados massivo para chatbots multiturno do inglês para o português do Brasil, treinam uma rede neural profunda com o conjunto de dados traduzido e ajustam a rede neural usando um conjunto de dados de diálogos relacionados ao COVID-19.

Dados de argumentos rotulados manualmente são uma tarefa árdua, Para minimizar o esforço neste problema, em (Peikari *et al.*, 2018) primeiro é aplicado um método de aprendizagem não supervisionado (clusterização) para encontrar um padrão em um conjunto de dados de Imagens Patológicas e então usam esses padrões (rótulos) para treinar o Modelo *SVM* para fazer a classificação. Suas observações e comparadas com outras abordagens de última geração mostraram bons resultados, mostrando que é possível usar métodos de aprendizado não supervisionado para obter rótulos para um conjunto de dados não classificado e usar esse conjunto de dados para treinar um modelo de aprendizado de máquina.

A Tabela 1 apresenta uma análise comparativa entre este trabalho e os estudos relacionados apresentados nesta seção, destacando elementos-chave que divergem e fazem adjacência. Esses trabalhos relacionados abordam diversos aspectos ligados ao uso de chatbots, *NLP* e aprendizado de máquina para lidar com desafios impostos pela pandemia da COVID-19. Essa pesquisa visa ampliar e aprimorar essas abordagens, introduzindo inovações específicas no contexto da descoberta de intenções em diálogos sobre a COVID-19, partir de um conjunto de dados não rotulados. Para melhor entendimento da Tabela 1, será explicado o que cada coluna representa.

Na Coluna **Trabalho**: são listados os trabalhos relacionados analisados para estabelecer um contexto e compreender as abordagens existentes na área de chatbots e *NLP* no contexto da COVID-19.

A Coluna **Modelo de Embedding**: indica os métodos específicos de embedding

utilizados por cada trabalho para representar palavras e sentenças. Cada método é escolhido com base em sua adequação ao contexto da pesquisa. Por exemplo, o trabalho (Klein *et al.*, 2021) utiliza BERT pré-treinado com tweets, enquanto (Lei *et al.*, 2021) emprega *NER* com base em artigos CORD-19.

A Coluna **Abordagem de Rotulação**: descreve o processo de anotação de dados adotado por cada trabalho. A anotação refere-se à marcação de dados com rótulos ou categorias relevantes para treinamento e avaliação de modelos. Por exemplo, (Li *et al.*, 2020a) realiza a classificação manual de tweets para atribuir emoções, enquanto (Klein *et al.*, 2021) utiliza expressões regulares para identificar tweets relacionados à COVID-19.

A Coluna **Arquitetura do Modelo NLU**: especifica as arquiteturas de modelos ou abordagens técnicas adotadas em cada trabalho. Isso inclui as escolhas relacionadas à estrutura dos modelos de chatbots, como redes neurais profundas, ou até mesmo o uso de abordagens específicas, como o módulo NLU Rasa.

Contudo, vale ressaltar que esse trabalho foi o único que utilizou mais de um modelo de embedding em seu pipeline, bem como aplicou uma abordagem de rotulação Semi-automática utilizada na maioria dos outros trabalhos, e teve como saída mais de uma arquitetura de modelo *NLU*.

Tabela 1 – Comparação com Trabalhos Relacionados

Trabalho	Modelo de Embeddings Usados	Abordagem de Rotulação	Arquitetura do Modelo NLU
(Lei *et al.*, 2021)	Não especificado	Manual	Gráfico de Conhecimento
(Fazzinga *et al.*, 2021)	Não especificado	Manual	Gráficos de Argumentação
(Gaglo *et al.*, 2021)	Não especificado	Semi-automática	Rasa
(Klein *et al.*, 2021)	BERT	Semi-automática	Redes neurais Profundas
(Li *et al.*, 2020a)	BERT	Manual	Não especificado
(Aguiar *et al.*, 2022)	Não especificado	Semi-automática	Redes Neurais Profundas
(Peikari *et al.*, 2018)	Não especificado	Semi-automática	*SVM*
Esse Trabalho	BERT, Glove, FLAIR, MUSE e LaBSE	Semi-automática	Feed-forward e Rasa NLU

Fonte: elaborado pelo autor.

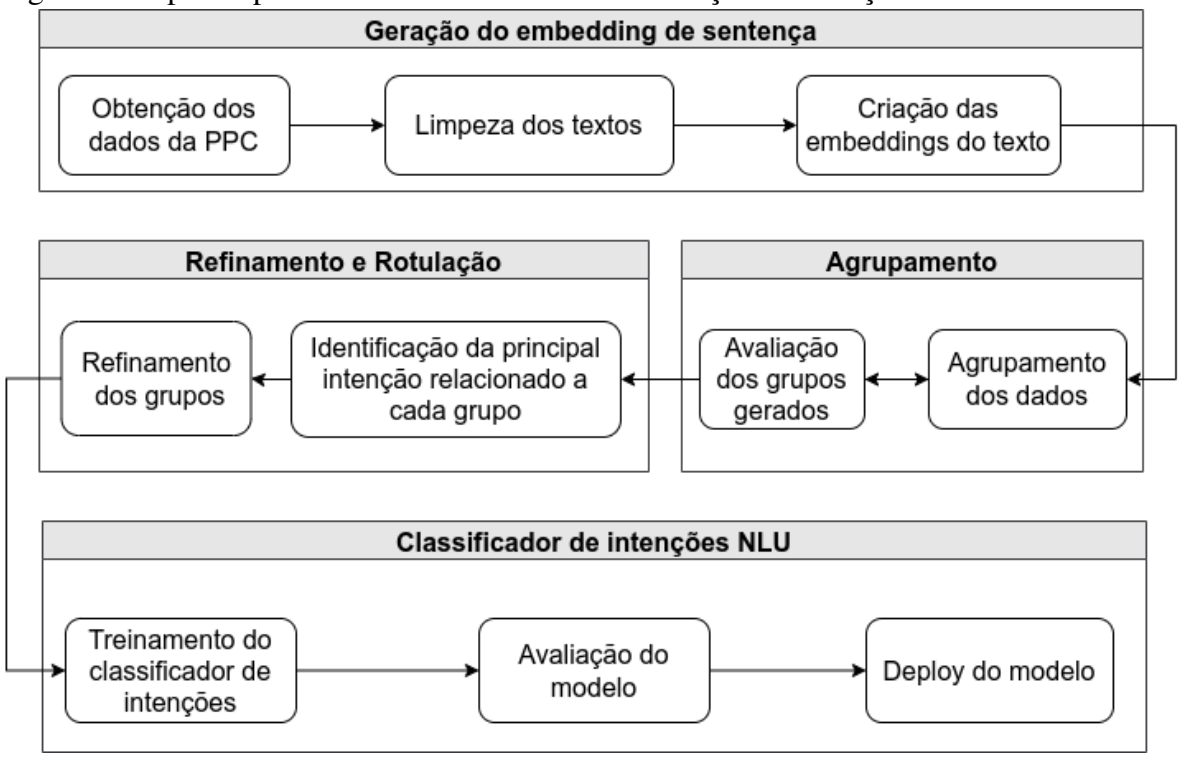
4 METODOLOGIA

Esse Capitulo trata-rá sobre a metodologia empregada nesse trabalho, explicando as técnicas e métodos que foram empregadas ao longo da condução dos experimentos. Na Seção 4.1 será discutido sobre o Pipeline aplicado desde a obtenção de dados até a construção e avaliação dos modelos de classificação de intenção *NLU*. As Seções seguintes irão tratar sobre cada passo do Pipeline de maneira mais especifica.

4.1 Pipeline para a classificação semi automática de intenção

O pipeline utilizado neste trabalho, representado na Figura 1 consiste na Geração de Embedding de Sentenças, Agrupamento dos dados, Refinamento dos dados, Rotulagem das Intenções e Treinamento do Modelo Classificador de Intenções.

Figura 1 – Pipeline para construir modelos de classificação de intenção NLU



Fonte: elaborado pelo autor (2024).

Uma das representações mais comuns do vocabulário de documentos são os embeddings de palavras pré-treinados. Outra alternativa é codificar a frase inteira em vetores de embeddings. Este método é conhecido como embedding de frases e existem muitos embeddings de frases pré-treinadas, conforme mencionado anteriormente. A etapa Geração do embedding de sentença tem como objetivo gerar representações vetoriais para as sentenças do paciente. Os

vetores resultantes desta etapa devem capturar informações sintáticas e semânticas das sentenças, de modo que as sentenças que expressam a mesma intenção tenham vetores próximos uns dos outros no novo espaço vetorial.

O objetivo da fase de agrupamento é agrupar os vetores de embedding de sentenças gerados na etapa anterior para compor grupos de sentenças referentes à mesma intenção. Nesta fase foram adotadas duas ferramentas de clusterização, a primeira foi o K-means a qual fez a plicação das métricas de separação e compacidade para definir os hiper parâmetros, que obtiveram o melhor resultado. Em geral, uma métrica de separação avalia o quão bem separado um cluster está em comparação com outros, e uma métrica de compacidade avalia o quão próximos estão os objetos pertencentes ao mesmo cluster. A outra ferramenta foi o BERTopic, que por sua vez agrupou as sentenças em tópicos. Na camada de embeddings do BERTopic, foram usados os mesmos modelos de embeddings, utilizados com o K-means. Tanto os clusters gerados pelo K-means, quanto os tópicos gerados pelo BERTopic, representam grupos.

Após o agrupamento, os grupos gerados são inspecionados visualmente para determinar a intenção associada a cada um. As sentenças do mesmo grupo são rotuladas com a intenção correspondente aquele grupo. A mesma intenção pode ser expressa de diferentes maneiras. Assim é possível reconhecer grupos distintos com a mesma intenção.

Ao realizar a rotulagem das sentenças, é assumido que todas as sentenças pertencentes a um grupo têm a mesma intenção. No entanto, esta hipótese pode ser inválida. Por esse motivo, antes de treinar o classificador, a fase de refinamento descarta as sentenças menos representativas de cada grupo, reduzindo assim a quantidade de exemplos de intenções errados para o aprendizado do classificador de intenção. Esta etapa é essencial para remover alguns *outliers* que não são reconhecidos pelo algoritmo K-means. O BERTopic, por sua vez, já realiza o descarte das sentenças que ele considera como *outiliers*, contudo foi mantido o mesmo processo de refinamento, onde para cada tópico gerado são realizados os mesmos processos de refinamento realizado na aplicação do K-means.

Por fim, as sentenças fornecidas pela etapa de refinamento são utilizadas para treinar e validar o classificador de intenção.

4.2 Obtenção dos dados da Plataforma do Plantão Coronavírus

O conjunto de dados da PPC tem mecanismos para triagem de pacientes por meio de interação via chatbot. O serviço é um chat online onde um chatbot realiza a primeira interação.

Com base na resposta do paciente a algumas perguntas predeterminadas, o chatbot classifica a condição do paciente conforme a criticidade, que pode ser leve, moderada ou grave. Após essa primeira interação, dependendo da classificação de criticidade, o paciente é encaminhado para um Tele atendimento com um profissional de saúde. A interação entre paciente e profissional de saúde fornece mais detalhes sobre as condições do paciente, incluindo sintomas mais específicos, que podem ser físicos ou psicológicos, na Seção 5.1 é discutido mais detalhes sobre a PPC. A Tabela 2 mostra exemplos de frases entre o Paciente e o Atendente.

Tabela 2 – Exemplo de Dialogo entre o Paciente e o profisionall de sáude

Ator	Sentença	Intenção
Atendente	Olá boa noite. Meu nome é ...	Greeting
Paciente	Estou a 03 dias com febre e ...	Inform Symptoms
Atendente	Você conseguiu pegar o ...	Request Inform
Paciente	Sim.	Others
Atendente	Você sente falta de ar?	Request Inform
Paciente	... tomei dipirona	Inform Medicine
Paciente	... Acho que por causa da tosse	Inform Symptoms
Atendente	Certo	Others
Atendente	Sente a falta de ar quando faz o quê exatamente?	Request Inform
Paciente	Só quando tusso	
...	...	...

Fonte: elaborado pelo autor.

No final, o paciente avalia o serviço. Para esta avaliação, a PPC exige que o paciente responda à pergunta "Você está satisfeito com o serviço?", a resposta deve ser "Sim"ou "Não"; e uma pontuação de avaliação variando de 0 a 10, onde 0 é o índice de satisfação mais baixo e 10 o índice de satisfação mais alto. O objetivo é construir um modelo de classificação para reconhecer automaticamente as intenções usando um conjunto de dados de diálogo não rotulado para identificar as intenções dos pacientes enquanto eles relatam suas condições de saúde. As intenções que nos interessam aqui referem-se ao diagnóstico do paciente.

A Tabela 3 apresenta um fragmento de um diálogo mostrando esta parte onde o PPC obtém a avaliação e o nível de satisfação do Paciente.

Portanto, o conjunto de dados PPC utilizado nesta abordagem é o conjunto de diálogos entre pacientes e profissionais de saúde com uma avaliação positiva. Na prática, os diálogos que o paciente informou estar satisfeito com o atendimento, e atribuiu nota 10. É composto por 1.237 diálogos coletados de 1º de maio de 2020 a 6 de maio de 2020, com um total de 53.633 sentenças. As frases dos diálogos são categorizadas com seu ator (pacientes ou profissionais de saúde), contudo não possui uma intenção associada a frase. Do total de

Tabela 3 – Fragmento de um diálogo mostrando o trecho da conversa onde é realizada a da avaliação do atendimento realizado pelo Paciente

Actor	Sentence
Bot	Obrigado pelo seu contato, Antonio! Seja um(a) aliado(a) essa luta, compartilhe esse serviço com pessoas e grupos que podem precisar da nossa ajuda. Juntos venceremos essa luta contra o novo Coronavírus! #FiqueEmCasa Até logo!
Bot	Suas dúvidas foram respondidas?
Patient	Sim
Bot	Por favor, antes de desconectar, avalie esse serviço. Dê uma nota de 1 a 10, sendo: 1.Ruim >>> 10. Muito bom!
Patient	10

Fonte: elaborado pelo autor.

sentenças, 26.647 sentenças são dos pacientes e 26.986 sentenças são dos profissionais de saúde. Uma vez que neste trabalho, o objetivo é identificar e conhecer as intenções dos pacientes. Foram usadas apenas as sentenças do paciente para construir o classificador de intenção *NLU*.

4.3 Limpeza do Texto

Na limpeza do texto foram removidas as frases duplicadas dos diálogos. Em seguida, são selecionados os diálogos melhor avaliados pelos pacientes, ou seja, conversas com pontuação igual a 10. É importante ressaltar que ao final de cada conversa, os pacientes devem avaliar o serviço de conversação com pontuação entre 0 e 10. O objetivo da filtragem é melhorar a qualidade dos diálogos no processo de aprendizagem. Também são retiradas as frases representadas pelas seguintes entidades: Código de Endereçamento Postal (CEP), Cadastro de Pessoas Físicas (CPF), números de telefone, *Uniform Resource Locator* (*URL*), emoticons, nomes de pacientes e locais. A remoção de tais entidades evita a criação de clusters não diretamente relacionados a uma intenção relevante. Além disso, evita o uso de qualquer informação pessoal dos usuários no processo de treinamento do modelo de intenção *NLU*.

4.4 Geração dos vetores de Embeddings

A primeira etapa visa gerar uma representação vetorial para as sentenças ditas pelos pacientes nas conversas. Os vetores resultantes desta etapa devem capturar informações sintáticas e semânticas de sentenças de forma que sentenças relacionadas à mesma intenção do usuário fiquem próximas umas das outras no espaço vetorial. Foram usados embeddings de sentenças

usando modelos pré-treinados do estado da arte. Nessa proposta, foram avaliados os modelos de embeddings de sentença, o MUSE (Yang *et al.*, 2019) e *LaBSE*, (Feng *et al.*, 2020); e modelos embedding de palavras, o FLAIR (Akbik *et al.*, 2019), BERTimbau (Souza *et al.*, 2020) e Glove (Pennington *et al.*, 2014).

Em relação aos modelos de embeddings de palavras, foi usado o vetor médio de todos os vetores de palavras como representação de sentença para os modelos de embeddings de palavras, de forma que o vetor resultante da operação consiga representar as sentenças, com vetores de mesmo tamanho.

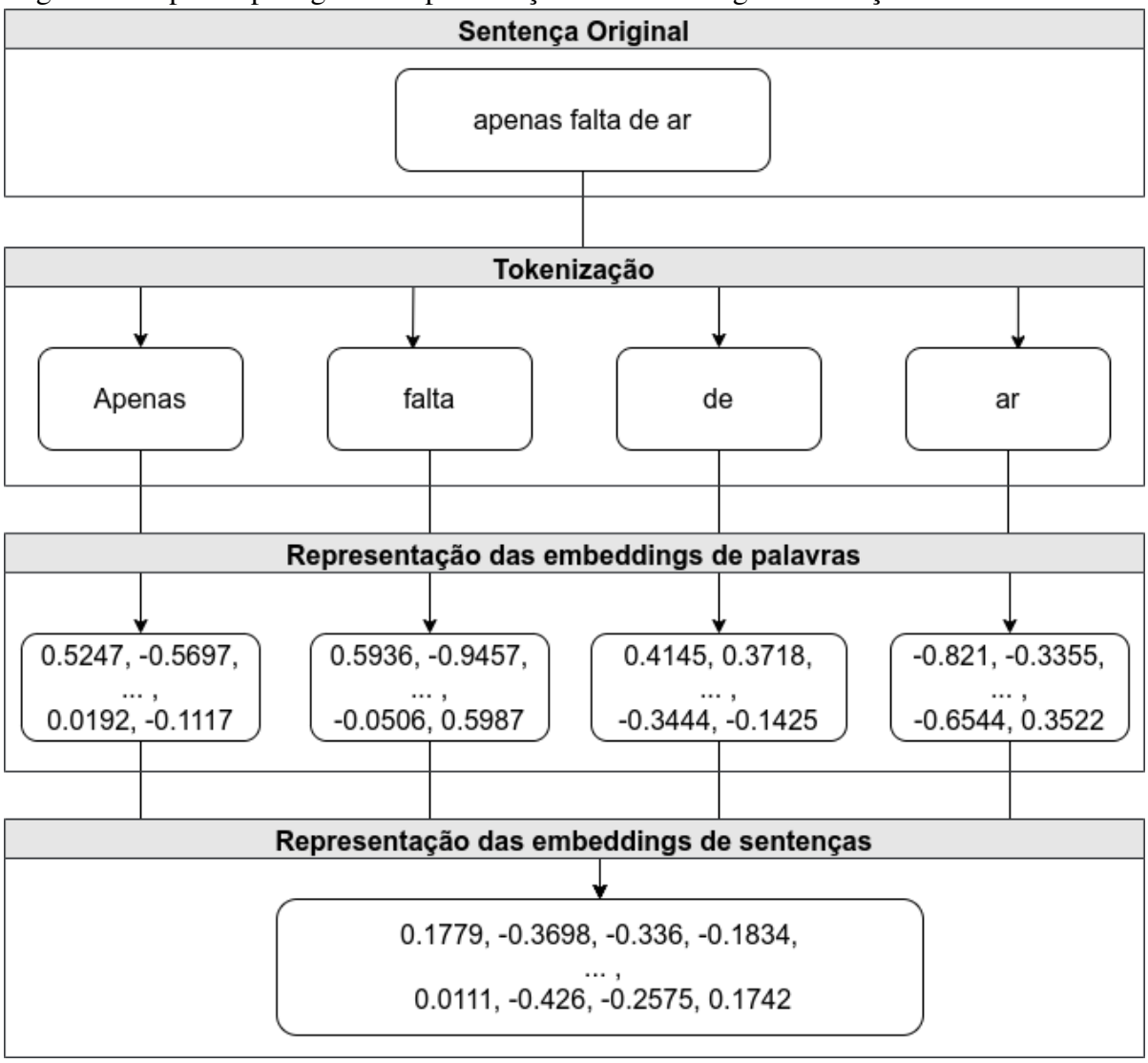
Por exemplo, usando o modelo de embedding Glove para representar a frase *"apenas falta de ar"*, temos o seguinte processo mostrado na Figura 2. O primeiro passo é para tokenizar a frase. Cada token encontrado na frase recebe um vetor com uma dimensão de 300, onde cada item é um número (a palavra representação de embedding); para simplificar a Figura 2 e torná-lo legível, mostramos apenas os dois primeiros e os dois últimos itens de cada vetor gerado. Na última etapa, aplicamos um cálculo médio de todos esses vetores para obter a representação vetorial da frase: um vetor com a mesma dimensionalidade como as embeddings palavra . Novamente, é mostrado apenas o primeiro e o último item do vetor de sentença gerado nesta etapa. Outros artigos realizaram a mesma abordagem que (Junior *et al.*, 2021; Wieting *et al.*, 2016).

Nos estágios iniciais dos modelos de linguagem neural, variantes fundamentais como Glove foram a base da *NLP* moderna, fornecendo recursos de palavras pré-treinadas. A principal vantagem que reside nesses modelos estáticos, é que geralmente é empregado uma arquitetura de rede neural superficial para realizar cálculos entre vetores de palavras, facilitando um treinamento eficiente. Mais recentemente, surgiram esforços para adquirir representações de palavras contextualizadas através de redes neurais profundas, exemplificadas por BERT e abordagens baseadas em BERT. Esses modelos de embeddings estão na vanguarda da área, superando os modelos estáticos em termos de recursos de última geração (Naseem *et al.*, 2021).

4.5 Agrupamento

O objetivo principal dessa etapa, é formar grupos a partir do conjunto de dados, de forma que os os elementos de cada grupo formado, seja composto por elementos que tenham o maior grau de semelhança em relação a elementos de outros grupos. Para ser possível identificar o conjunto de intenções do paciente e rotular o texto da conversa usando os vetores de embeddings

Figura 2 – Pipeline para gerar a representação de embedding de sentenças usando Glove



Fonte: elaborado pelo autor (2024).

gerados na 4.4. Foram utilizadas duas ferramentas de clusterização para realizar esse processo de agrupamento. A primeiro consistiu em realizar a aplicação do algorítimo K-means e a segunda foi a utilização do BERTopic, os tópicos gerados por esta ferramenta, podem ser comparados aos clusters gerados pelo clusterização com o K-means.

4.5.1 K-means

Para esta tarefa de agrupamento, uma das ferramentas usadas foi o conhecido algoritmo de clusterização K-means com similaridade de cosseno. O valor de k foi variado e escolhido o melhor valor com base na *Davies Bouldin Score* (*DBS*) (Davies; Bouldin, 1979) e *Silhouette Score* (*SS*) (Rousseeuw, 1987). A *DBS* é uma métrica de separação dada pela similaridade média entre um cluster e seu cluster mais semelhante. Conforme a *DBS*, a melhor

clusterização minimiza a similaridade média e o menor valor de similaridade é0. Assim, valores mais baixos da *DBS* indicam melhor clusterização. A ideia por trás do uso de uma métrica de separação é reduzir a sobreposição de intenções entre os clusters. A *SS* indica a relação entre coesão e separação. Ele estima a semelhança entre o objeto e seu cluster em comparação com a semelhança entre o objeto e os outros clusters. Como foram experimentados vários modelos de embeddings para representar as sentenças das conversas, cada modelo de embedding pode resultar em diferentes valores ótimos de *k*.

4.5.2 BERTopic

Assim como foi feito em 4.5.1 para identificar o conjunto de intenções do paciente usando o K-means, a aplicação da ferramenta BERTopic, usou como entrada os vetores de embeddings gerados na 4.4. A quantidade de Tópicos foi encontrada pela própria ferramenta. Como foram experimentados vários modelos de embeddings para representar as sentenças das conversas, cada modelo de embedding pode resultar em diferentes valores para a quantidade de tópicos encontrados.

4.6 Rotulação das Intenções

Foi utilizado inspeção visual para rotular as sentenças em intenções que correspondem a cada grupo gerado. Em geral, a inspeção visual acaba sendo uma ferramenta útil sempre que (1) diferentes abordagens produzem grupos com semânticas diferentes, (2) diferentes conjuntos de parâmetros produzem grupos que apresentam bom desempenho em termos de métricas de qualidade, mas mostram claramente características diferentes, ou (3) quando a verdade básica não estiver disponível. Pode-se confirmar que a inspeção visual também foi adotada em vários trabalhos anteriores (Ester *et al.*, 1996; Han *et al.*, 2012; Silva *et al.*, 2020a).

As técnicas de visualização utilizadas foram t-Distributed Stochastic Neighbor Embedding (t-SNE) e nuvens de palavras. A abordagem t-SNE (Maaten; Hinton, 2008) permite a visualização de dados de alta dimensão, convertendo cada ponto de dados em um espaço bidimensional ou tridimensional que exibe a estrutura de dados em múltiplas escalas. O t-SNE permite visualizar a distribuição das intenções e a sobreposição dos grupos. As nuvens de palavras foram aplicadas para visualizar as palavras mais frequentes nos grupos. Em seguida, as sentenças foram rotuladas com suas respectivas intenções conforme a intenção atribuída ao

grupo para criar os conjuntos de treinamento e teste para o modelo de classificação de intenção *NLU*. Resumindo, os dados de treinamento consistem em exemplos de sentenças ditas pelos pacientes categorizadas por uma intenção.

Nesta etapa, são identificadas as intenções correspondentes a cada grupo. Dado o grande número de sentenças, foi aplicado uma avaliação empírica por meio de análise visual. Foi usado o t-SNE (Maaten; Hinton, 2008). Esta ferramenta permite visualizar a distribuição das intenções e a sobreposição dos grupos. Em conjunto com t-SNE, foram usadas nuvens de palavras para a visualização do conteúdo dos grupos. Em seguida, foram rotuladas as sentenças com suas respectivas intenções conforme a intenção atribuída ao grupo, foram avaliadas as 10 sentenças mais próximas do centroide, com o intuído de comprovar se elas realmente pertenciam à intenção identificada, para criar o conjunto de treinamento para o modelo *NLU*.

4.7 Refinamento dos dados

Foram filtradas as sentenças representativas antes de treinar o classificador de intenção e foram descartadas os *outliers* dos grupos. Foi definido um limite superior para a distância permitida entre as sentenças no cluster e seu centroide clusterizado pelo K-means, então as sentenças com distância maior que o valor do limite superior foram removidas. Foi usado a distância do cosseno entre duas sentenças para calcular as distâncias entre seus vetores de embedding. O BERTopic por sua vez provê uma métrica chamada *Probability*, a qual indica o grau de certeza de determinado elemento pertencer ao tópico, com isso foram aplicados os mesmos critérios de refinamento do k-means, porém ao invés de usar a distância do elemento ao centroide, foi usado a métrica *Probability*.

Duas estratégias para os limites superiores foram propostas. Seja CT um cluster ou tópico e CT_{Q_i} o *i*ésimo quartil das distâncias das sentenças em CT ao centroide de C ou o *i*ésimo quartil da *Probability* de um elemento pertencer a um tópico. A primeira estratégia aplica a remoção dos outliers que pode ser calculado pelas Equações 4.1 e 4.2. O valor *1.5* usado na Equação 4.1 é um valor comum usado para encontrar os *outliers* do conjunto de dados conforme explicado em (Hoaglin; Iglewicz, 1987).

$$outliers(CT) = CT_{Q_3} + (1.5 * iqr) \tag{4.1}$$

$$iqr(CT) = CT_{Q_3} - CT_{Q_1} \tag{4.2}$$

Na segunda estratégia, o limite superior é a mediana da distância (Equação 4.3).

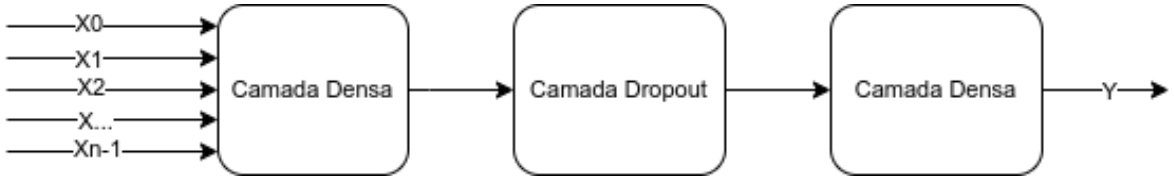
$$mediana(CT) = CT_{Q_2} \tag{4.3}$$

Além do refinamento dos dados, também foram removidos do conjunto de dados todas as sentenças pertencentes aos grupos onde não é possível identificar uma intenção precisa, as quais foram rotuladas com a intenção *others*. Foram avaliadas as diferentes estratégias de refinamento medindo a qualidade dos resultados do K-means e do BERTopic, com e sem a intenção *others*.

4.8 Classificação de intenção

Para classificação de intenções, foram empregadas duas arquiteturas. A primeira, é um Classificador de Intenção baseado em *NN*, ele é construído usando a representação de embedding usada para o processo de agrupamento. A arquitetura do classificador *NN* é apresentada na Figura3, ela é baseada em uma rede neural *feed-forward* simples. O vetor de embedding de sentença é usado como entrada para o modelo e então processado por uma camada totalmente conectada. Para reduzir o *over-fitting*, a camada de entrada é vinculada a uma camada *dropout*. A taxa de frequência usada para a camada *dropout* foi de 0,1. Uma camada totalmente conectada com *so f tmax* como função de ativação produz a saída. Finalmente, o número de intenções definidas na etapa de rotulagem de intenções determina a dimensionalidade da saída. Esses modelos são treinados usando a *cross entropy* como função de perda e o algoritmo Adam como otimizador.

Figura 3 – Arquitetura do modelo de classificação de intenção



Fonte: elaborado pelo autor (2024).

O Rasa *NLU* também é aplicado como classificador de intenções. O Rasa *NLU* é treinado usando a estrutura do Rasa, que requer a especificação de um pipeline Rasa compre-

endendo vários componentes, como tokenizer, extrator de recursos e arquitetura classificadora. A sequência desses componentes converte a frase de entrada em dados estruturados, que são então passados para um modelo de aprendizado de máquina. Para o pipeline Rasa, foi utilizado o módulo spaCy com o componente *SpacyNLP* que recebe um template pré-treinado do spaCy no idioma desejado (português para esse trabalho). No pipeline aplicado o *SpacyTokenizer* é o tokenizer; e *SpacyFeaturizer*, *RegexFeaturizer* e *CountVectorsFeaturizer* são os extratores de recursos que criam a representação vetorial de mensagens a serem fornecidas como entrada para o classificador. Observe que ao usar esses componentes para extração de recursos, o classificador de intenção Rasa NLU é independente da representação de embedding aplicada para agrupar e rotular sentenças no pipeline desse trabalho. Finalmente, é aplicado o Classificador Dual Intent e Entity Transformer (*DIETClassifier*) como arquitetura do modelo de classificação. DIET é uma arquitetura multitarefa baseada em transformadores que podem prever intenções e entidades.

Foi necessário ainda anotar as entidades do texto, para treinar o modulo *NLU* do Rasa. Para isso foi usada a ferramenta chamada SINTOMATIC (Silva *et al.*, 2020b) para detectar e identificar sintomas (entidades) presentes nos diálogos. O SINTOMATIC ajuda a identificar padrões de sinais de doenças e novos sintomas ou sintomas raros, que os profissionais de saúde ainda não mapearam. Assim, seguindo a evolução dos sintomas do COVID-19 nos diálogos do usuário. Portanto, o SINTOMATIC foi usado para extrair sintomas das conversas do conjunto de dados PPC. Neste trabalho, os sintomas são interpretados como entidades adequadas para o reconhecimento de dados estruturados em modelos *NLU*. Vale ressaltar que é possível usar diferentes modelos de *NER* (Li *et al.*, 2020b), como os disponíveis em spaCy *NER*[1], para extrair as entidades. Contudo foi focado apenas na extração dos sintomas como entidades, uma vez que eles estão relacionados às intenções de diagnóstico do paciente nas quais se tem interesse.

Para avaliar o modelo de intenção NLU treinado, foram calculadas as métricas de precisão, recall, F1, Acurácia e *Matthews Correlation Coefficient* (*MCC*). Elas podem ser definidas em termos de uma matriz de confusão M para N classes, onde m_{ki} é o número de mensagens da intenção I_k classificadas como pertencentes à intenção I_i. Algumas considerações podem ser feitas para simplificar o entendimento das definições, para isso considere as seguintes variáveis intermediárias:

– $t_k = \sum_i^N m_{ki}$ o número de vezes que a intenção I_k realmente ocorreu.

– $p_k = \sum_i^N m_{ik}$ o número de vezes que a intenção I_k foi prevista.

[1] https://v2.spacy.io/api/entityrecognizer

– $c = \sum_k^N m_{kk}$ o número total de amostras previstas corretamente.

– $s = \sum_i^N \sum_j^N m_{ij}$ o número total de amostras.

Após essas considerações, as métricas são descritas nas Equações 4.4, 4.5, 4.6, 4.7 e 4.8.

Em resumo, a precisão da intenção I_k é a porcentagem de instâncias da intenção I_k entre todas as instâncias classificadas como pertencentes à intenção I_k.

$$\text{Precisão}(I_k) = \frac{m_{kk}}{p_k} \tag{4.4}$$

O recall da intenção I_k é a porcentagem de instâncias da intenção I_k entre todas as instâncias que pertencem à intenção I_k.

$$\text{Recall}(I_k) = \frac{m_{kk}}{t_k} \tag{4.5}$$

A F1 Score da intenção I_k é a média harmônica entre sua precisão e recuperação.

$$\text{F1}(I_k) = 2\frac{\text{Precision}(I_k)\text{Recall}(I_k)}{\text{Precision}(I_k) + \text{Recall}(I_k)} \tag{4.6}$$

A acurácia é a fração das previsões corretas feitas pelo modelo pelo total de sentenças.

$$\text{Acurácia} = \frac{c}{s} \tag{4.7}$$

O *MCC* considera verdadeiros e falsos positivos e negativos e é geralmente considerado uma medida equilibrada que pode ser usada mesmo que as classes sejam de tamanhos muito diferentes. Nesse trabalho h´á rótulos multi classe, que possuem um valor mínimo variando entre -1 e 0 dependendo do número e distribuição dos rótulos verdadeiros básicos, e o valor máximo é sempre +1. O *MCC* pode ser definido como mostrado na Equação 4.8 (Scikit-Learn, 2023).

$$\text{MCC} = \frac{c \cdot s - \sum_k^N p_k \cdot t_k}{\sqrt{(s^2 - \sum_k^N p_k^2) \cdot (s^2 - \sum_k^N t_k^2)}} \tag{4.8}$$

Na avaliação experimental, são relatados a precisão média, o recall médio, a média f1 e a acurácia média ao longo de todas as intenções e *MCC*. 0

5 RESULTADOS

Neste Capítulo serão descritos os resultados obtidos com este trabalho. Na Seção 5.1 é discutido sobre o que é a PPC e como ela funciona. Na Seção 5.2 é discutido sobre a Pergunta de pesquisa (RQ1), onde é explicado sobre o processo que foi aplicado para realizar a rotulação dos diálogos, atribuindo uma intenção para cada uma. Na Seção 5.3 é discorrido sobre a Pergunta de pesquisa (RQ2), onde é explicado o processo utilizado na construção dos modelos de classificação de intenções. Na Seção 5.4 é apresentado a Pergunta de pesquisa (RQ3), onde é abordado sobre a qualidade da presentação dos modelos de embeddings. E por fim na Seção 5.5, é analisado a Questão de pesquisa (RQ4), na qual é discutido osbre o erro de rotulagem adicionado pelo processo de agrupamento.

5.1 Plataforma do Plantão Coronavírus

A PPC, possui mecanismos que permitem a triagem dos pacientes por meio de interações via chatbot. O serviço é composto por um chat online onde a primeira interação é realizada por chatbot, o objetivo dele é classificar o estado do paciente de acordo com a criticidade, que pode ser leve, moderada ou grave. Após esse primeiro atendimento, dependendo dessa triagem que é realizada, o paciente pode ser direcionado para um tele atendimento com um profissional de saúde (Silva *et al.*, 2020b).

As interações entre os paciente e os profissionais de saúde, possuem muito valor visto que representam formas como os pacientes e profissionais de saúde se comunicam, é possível através dessa interação encontrar padrões da doença, sintomas mais comuns entre outras possibilidades que ajudem a avaliar o quadro de desenvolvimento da doença de uma maneira geral.

É possível ainda usar esses dados em conjunto com alguma técnica de aprendizagem de máquina não supervisionada como a clusterização para classificar as sentenças dos diálogos em intenções, essas intenções podem servir para realizar o treinamento de algum modelo de de Classificação de Intenções *NLU*, o qual pode ser usado na construção de uma ferramenta de chatbot.

5.2 Análise do processo da rotulação

Os experimentos realizados e explicados nesta Seção permanecem com a questão de pesquisa **(RQ1)**: *A partir de um enorme conjunto de dados de conversas, como rotular intenções usando aprendizagem não supervisionada para diálogos com frases curtas, sem caracterizar perguntas e respostas ao longo da conversa?*.

Como foi mencionado anteriormente, o conjunto de dados de diálogo não é anotado em relação aos rótulos de intenção. Uma alternativa é realizar uma estratégia de agrupamento no conjunto de dados de acordo com a semelhança entre os diálogos dos pacientes. Portanto, cada grupo pode representar um rótulo de intenção associado aos textos atribuídos naquele grupo.

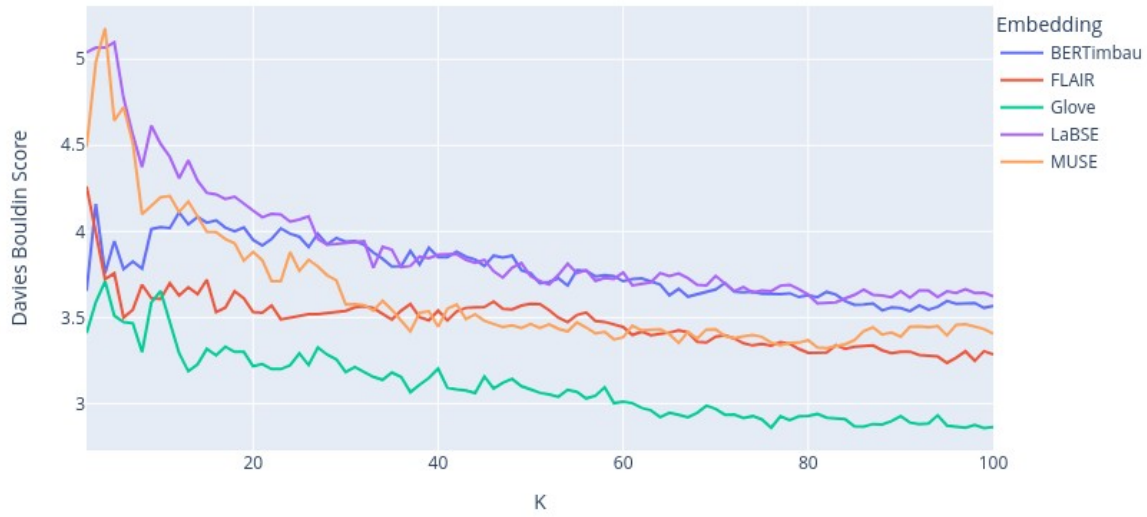
Antes de realizar o agrupamento, as sentenças foram representadas usando modelos de embedding. O agrupamento foi realizado utilizando duas abordagens, a primeira consistiu em aplicar K-means, um algoritimo de clusterização, e a segunda foi a aplicação do BERTopic, para agrupar os dados em tópicos.

Na clusterização foi analisado o número ideal de clusters para cada modelo de embedding de texto variando o número de clusters k entre 2 e 100. Além disso, foi utilizado o *DBS* e *SS*, com distância do cosseno, como métricas para escolher o melhor valor para k dentro deste intervalo.

A figura 4 mostra os resultados da métrica *DBS* para todos os modelos. O objetivo é minimizar o valor *DBS* (o melhor valor é zero (0)). Embora todos os modelos de embeddings tenham um desempenho ligeiramente igual, de acordo com a Figura 4, é possível notar que o modelo de embedding com melhor valor de *DBS* foi o Glove. A figura 5 mostra os resultados da métrica *SS* para todos os modelos. É importante lembrar que o melhor valor de *SS* é um (1).

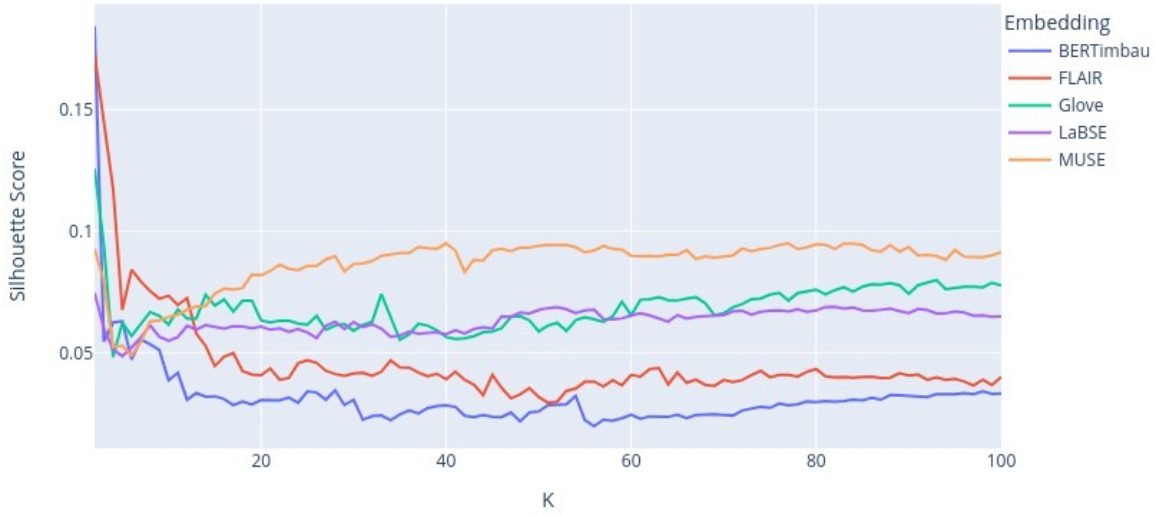
A tabela 4 mostra as métricas, os valores de k escolhidos de acordo com as métricas e o tipo de codificador (a abordagem de embedding). Pode-se facilmente perceber que, infelizmente, os valores encontrados para o *SS* são menores que 1 e muito próximos de 0, indicando uma sobreposição entre clusters encontrados por todos os modelos de embedding utilizados nestes experimentos. Isso significa que a distância entre uma sentença S e as outras sentenças S^0 no mesmo cluster é quase a mesma que a distância entre S e outra sentença atribuída a um cluster diferente. Um motivo pode ser que as representações do texto pelos vetores numéricos com alta dimensionalidade, ou seja, de cerca de 500 a mais de 4.000. Quando é aumentado a dimensionalidade dos vetores, sua distância aos pares tende a aumentar. Então por esse motivo, foi optado por selecionar o valor de k com base no melhor valor do *DBS*. Neste caso, quanto

Figura 4 – Davies Bouldin Score de cada Modeloe de Embedding.



Fonte: elaborado pelo autor (2024).

Figura 5 – Silhouette Score de cada Modelo de Embedding.



Fonte: elaborado pelo autor (2024).

menor o *DBS*, melhores serão os clusters.

Na aplicação do BERTopic, foi alterado a camada de embedding para utilizar os mesmos modelos usados no processo de clusterização. Contudo o BERTopic tem a capacidade de encontrar o número ideal de tópicos de maneira automática. Então o número de tópicos encontrados pelo BERTopic, não foram variados.

Para complementar ainda mais a avaliação da qualidade do agrupamento, *foi inspeci-*

Tabela 4 – O valor de K escolhido para cada modelo de embedding

Embedding	Codificador	Dimensionalidade	K	DBS	SS
BERTimbau	Palavra	512	91	3.5375	0.0323
FLAIR	Palavra	768	95	3.2366	0.0393
Glove	Palavra	768	99	**2.8577**	0.0788
LaBSE	Sentença	300	81	3.5825	0.0688
MUSE	Sentença	4096	82	3.3227	**0.0928**

Fonte: elaborado pelo autor.

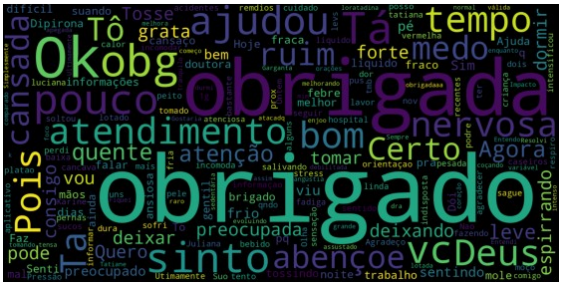
onado visualmente alguns dos resultados produzidos a partir das embeddings de sentenças em cada modelo de embedding. Para cada modelo de embedding, o melhor número de clusters (k) é quase 100 para a clusterização, já no BERTopic, o modelo que produziu menos tópicos, gerou um pouco mais de 300 tópicos, o que significa que inspecionar visualmente tantos grupos é inviável. Portanto, foi inspecionado visualmente alguns dos grupos gerados usando t-SNE e nuvens de palavras. São apresentadas essas visualizações para os clusters a partir da representação do modelo de embedding, que obteve o melhor desempenho. Na tabela 4, o Glove supera os demais na métrica *DBS*.

A Figura 6 mostra exemplos de nuvens de palavras construídas a partir de alguns clusters de sentenças obtidas por meio da clusterização, realizado utilizando o modelo de embedding Glove. É importante ressaltar que cada nuvem de palavras é formada apenas pelas sentenças de um determinado cluster, que foi rotulado com uma intenção específica. A figura 7(a) mostra uma nuvem de palavras para um cluster representando uma intenção de saudação. 7(b) mostra um cluster, representando a intenção Informar sintomas. 7(c) mostra um cluster, representando a intenção Informar O Medicamento. 7(d) mostra um cluster representando a intenção Request Inform. 7(e) mostra um cluster que representa a intenção others.

A técnica de visualização t-SNE foi utilizada para apresentar a distribuição das sentenças no espaço vetorial. A figura 7 mostra a distribuição dos vetores de sentença para os **99** clusters gerados usando o modelo de embedding Glove. As sentenças com a mesma cor pertencem a mesma Intenção.

A Tabela 5, mostra algumas frases que fazem parte de clusters formados com o modelo de embedding Glove em que os pacientes relatam sintomas. Vale ressaltar que a mesma intenção pode ser encontrada em mais de um cluster. Isto já é esperado uma vez que o valor da métrica *SS* é próximo de 1, conforme mostrado na Tabela 4. Por inspeção visual, foi decidido mesclar os grupos relacionados à mesma intenção ao invés de ter vários clusters representando o mesmo rótulo.

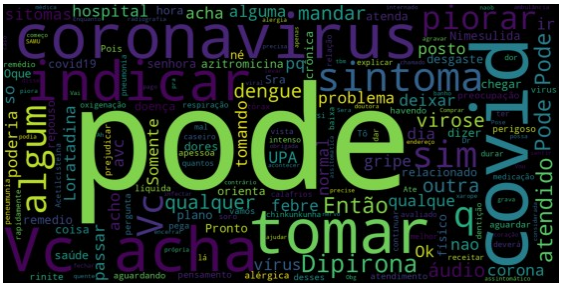
Figura 6 – A word cloud generated by Glove embedding model representing sentences of one cluster intention



(a) Greeting

(b) Inform Symptoms

(c) Inform Medicine



(d) Request Inform

(e) Others

Fonte: elaborado pelo autor (2024).

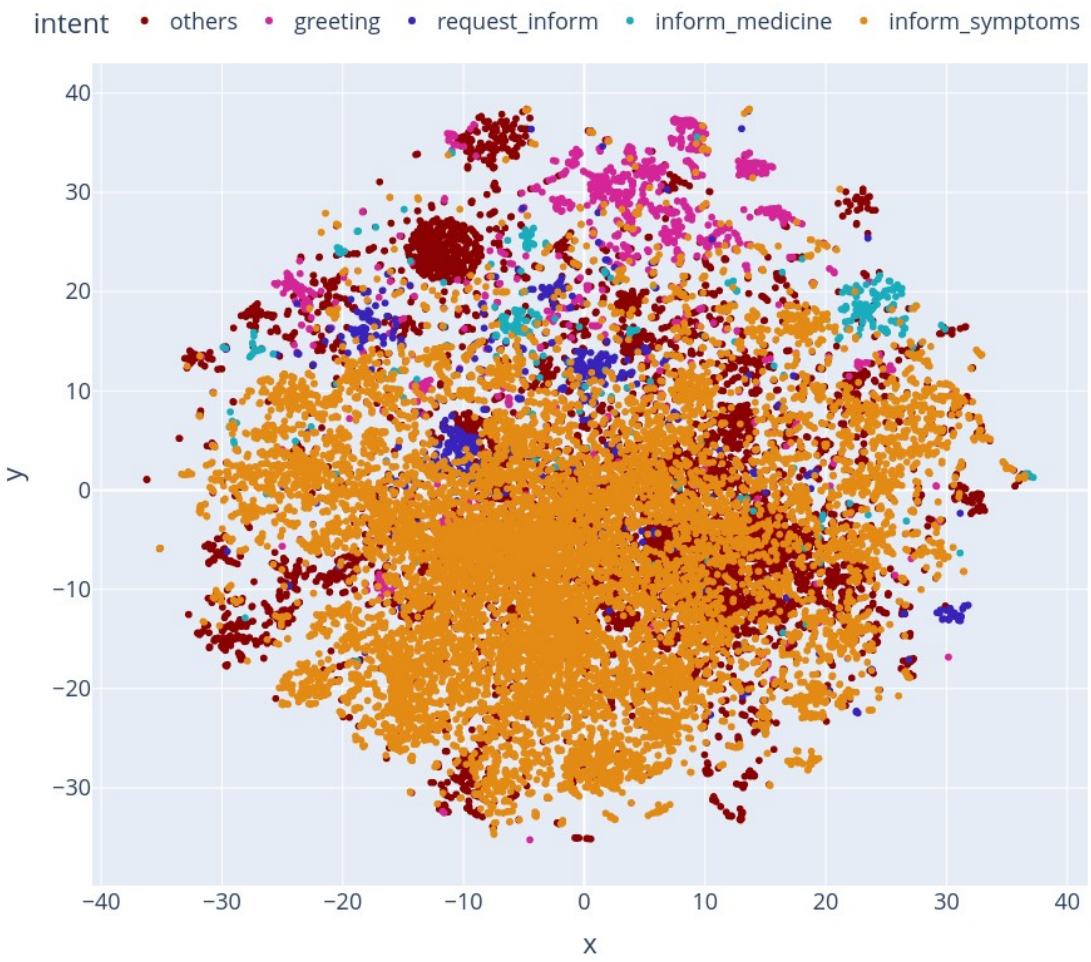
Tabela 5 – Frases de um cluster Glove representando uma intenção de inform_symptoms

Sentença
Tenho cansado ao falar, e perdi totalmente o apetite, não sinto gosto da comida e o cheiro eu sinto, mas é pouco.
Não sei se outros sintomas aparecerão. Mas, por enquanto, só sinto dores no corpo e nas costas.
Às vezes sinto falta de ar e dor no peito, mas nunca tentei descobrir o que era foi que eu sempre tomava aerossol e melhorava
Sentido mal, com vontade de vomitar. Aqui em casa está eu e minha mãe assim

Fonte: elaborado pelo autor.

Com base nas inspeções realizadas através do t-SNE, foi atribuído a cada grupo um rotulo que melhor representasse a intenção presente nas sentenças daquele grupo. Vale a

Figura 7 – t-SNE visualização para os 99 clusters gerados com o modelo de Embedding Glove



Fonte: elaborado pelo autor (2024).

pena relembrar que durante toda a experimentação são utilizadas apenas frases do paciente, eliminando a necessidade de diferenciar o ator nas categorias de intenção. No geral, as intenções identificadas nos clusters foram as seguintes:

1. **Greeting**: Frases relacionadas a cumprimentos ou saudações, por exemplo, quando o paciente agradece, inicia uma conversa ou até mesmo se despede, em frases como *"Olá", "Bom dia", "Muito obrigado"*. Observe que a Figura 7(a) apresenta algumas palavras relacionadas a saudações.
2. **Inform Symptoms**: Frases onde são relatados sintomas do paciente, exemplos são: *"Amanheci com dor de cabeça", "Uma leve dor de cabeça"*. Observe que a Figura 7(b) apresenta os sintomas frequentes e como os usuários descreveram como se sentiram.
3. **Inform Medicine**: Frases onde o paciente informa algum medicamento que está tomando, por exemplo, *"Tomei paracetamol ontem à noite", "Só tomo dipirona"*. Observe a Figura 7(c) mostra as palavras frequentes relacionadas aos medicamentos.

4. **Request Inform**: Quando o paciente solicita alguma informação ao profissional de saúde, como nestes exemplos: *"A Dipirona é mais eficaz?", "Onde posso fazer o exame?"*. A figura 7(d) apresenta as palavras frequentes utilizadas nos diálogos para solicitar informações sobre o COVID-19.
5. **Others**: Cluster com dificuldade em identificar a intenção primária ou frases que representem outros tipos de intenções além das apresentadas acima. A figura 7(e) ilustra algumas palavras frequentes do cluster rotuladas como outras.

A Tabela 6 mostra a distribuição entre o número de clusters rotulados por tipo de intenção encontrados em cada modelo de embedding.

Tabela 6 – Número de clusters por Intenção

Intention	BERTimbau	FLAIR	Glove	LaBSE	MUSE
Greeting	14	14	14	7	4
Inform Symptoms	31	35	35	25	23
Inform Medicine	4	3	5	7	6
Request Inform	8	6	6	8	8
Others	34	37	39	34	41
Total	91	95	99	81	82

Fonte: elaborado pelo autor.

A Tabela 7 mostra a distribuição entre o número de tópicos rotulados por tipo de intenção encontrados em cada modelo de embedding.

Tabela 7 – Número de tópicos por Intenção

Intention	BERTimbau	FLAIR	Glove	LaBSE	MUSE
Greeting	33	30	58	36	48
Inform Symptoms	113	114	141	89	88
Inform Medicine	20	15	25	14	20
Request Inform	35	12	35	46	28
Others	116	142	224	160	137
Total	317	313	483	345	321

Fonte: elaborado pelo autor.

No geral, a abordagem adotada nesse trabalho mostrou que, mesmo sem caracterizar perguntas e respostas ao longo da conversa, foram rotuladas com sucesso as intenções nos diálogos utilizando métodos não supervisionados. Isto foi conseguido representando sentenças com um modelo de embedding e posteriormente agrupando os vetores resultantes. É fundamental ressaltar que esse método de rotulagem é automatizado. No entanto, na Seção 5.5, foi conduzido um exame aprofundado do erro potencial introduzido durante a fase de agrupamento.

5.3 Análise do modelo NLU para classificação de intenções

Os experimentos discutidos nesta seção estão relacionados à seguinte questão de pesquisa: **(RQ2)** Como criar um modelo NLU para classificação de intenções usando os dados rotulados semi automaticamente de **(RQ1)**?

Já é esperado que o K-means possa não identificar outliers. Portanto, para melhorar a qualidade dos dados rotulados antes de usá-los para treinar o modelo de intenção *NLU*, é necessário eliminar potenciais outliers dentro dos clusters. Para isso, foi calculado a distância do cosseno entre cada sentença e o centróide do cluster ao qual ela pertence. A remoção de valores discrepantes é baseada neste cálculo de distância.

As sentenças que apresentavam uma distância significativa dos centróides dos clusters foram excluídas do conjunto de dados de treinamento. Este refinamento, que visa eliminar outliers e frases não representativas dos clusters, foi implementado através da aplicação de filtros, nomeadamente *upper_bound_outiliers* e *upper_bound_median*, conforme descrito na Secção 4.7. Notavelmente, certos clusters foram identificados como contendo sentenças de diversos contextos, todos rotulados com a intenção *others*. Consequentemente, juntamente com o processo de refinamento inicial, foram filtrados os clusters associados à intenção *others*, reconhecendo o ruído potencial introduzido no classificador de intenções. Ao todo, essa avaliação da qualidade da clusterização passa por 6 cenários detalhados abaixo: consideradas todas as sentenças (**A**), incorporando diversas estratégias de refinamento e a presença (**B e C**) ou ausência (**D , E e F**) da intenção *others*. Os resultados de *DBS* e *SS* são ilustrados nas Figuras 8 e 9.

A legenda nestas Figuras indica o conjunto de dados utilizado para fins de cálculo e avaliação:

– **A**: Conjunto de dados com todas as frases.

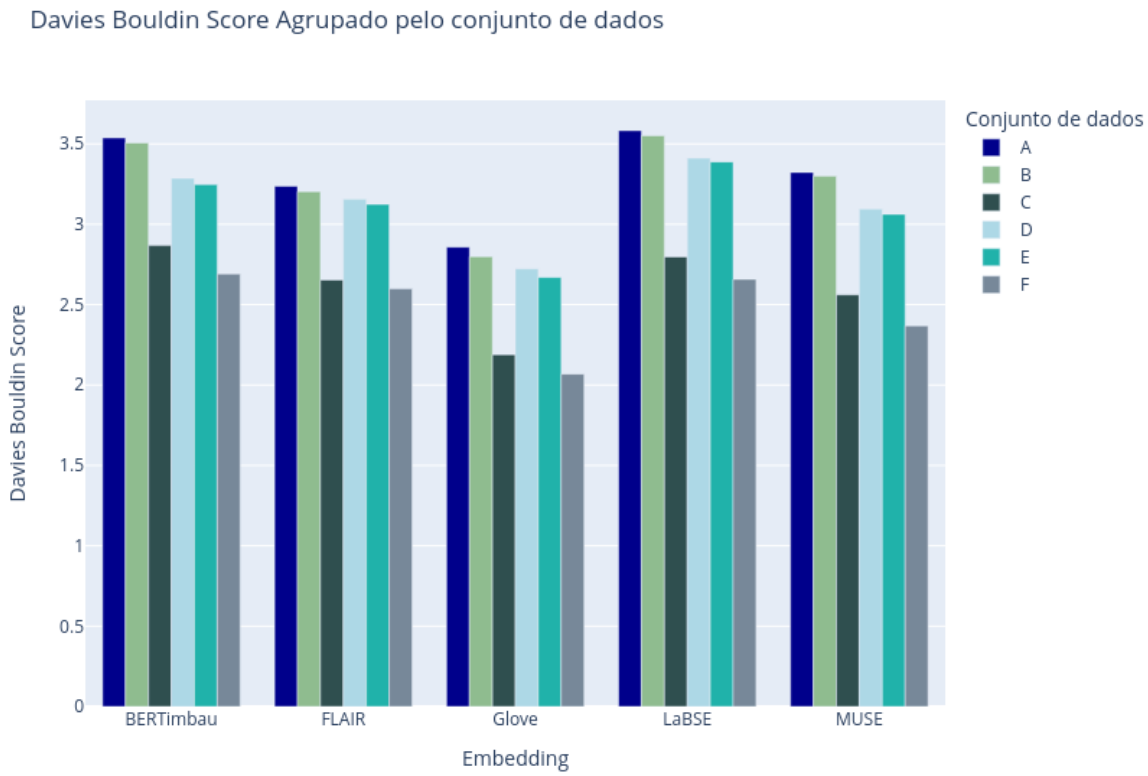
– **B** : Conjunto de dados refinado excluindo sentenças consideradas discrepantes, ou seja, com uma distância de cosseno ao centróide de seu cluster excedendo o valor do limite superior.

– **C** : Conjunto de dados refinado excluindo sentenças com uma distância de cosseno ao centróide de seu cluster excedendo o valor da mediana (*upper_bound_median*).

– **D** : Conjunto de dados refinado excluindo as sentenças associadas a clusters rotulados com a intenção *others* .

– **E** : Conjunto de dados sujeito ao mesmo processo de exclusão que em**B e D**, onde tanto sentenças discrepantes quanto sentenças pertencentes a clusters rotulados como *others* são

Figura 8 – Davies Bouldin scores for datasets variations.



Fonte: elaborado pelo autor (2024).

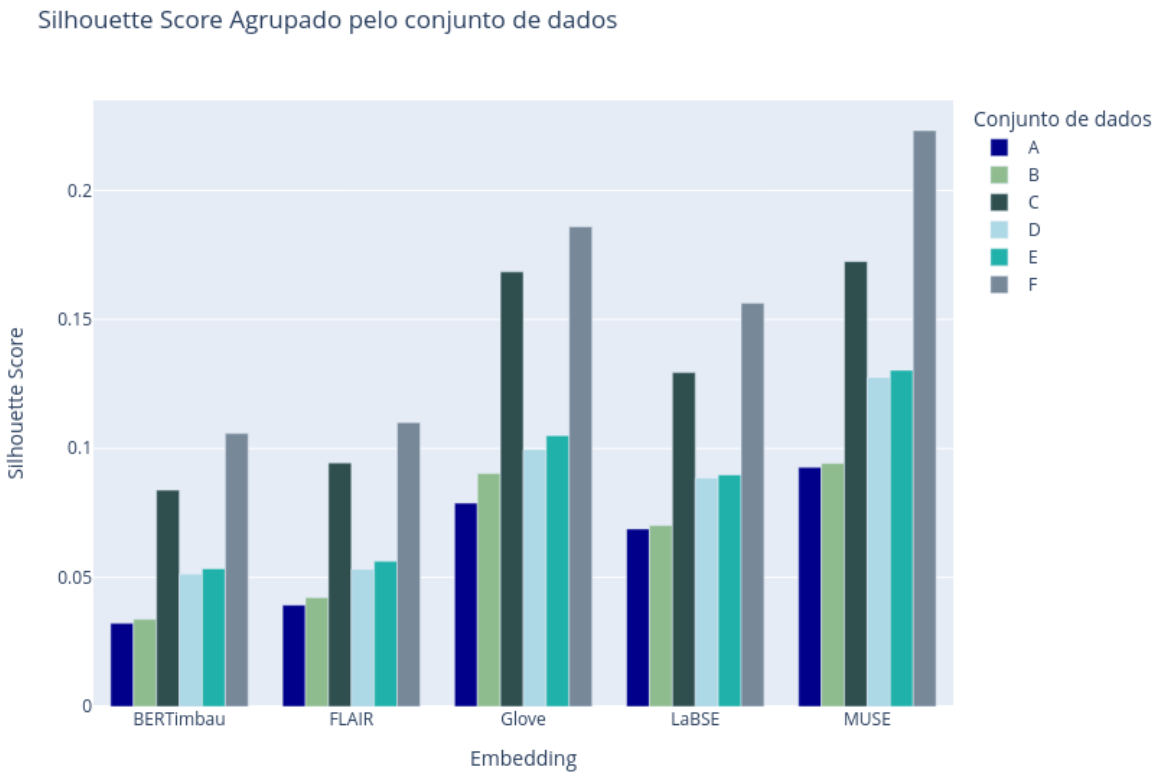
descartadas.

– **F** : Conjunto de dados sujeito ao mesmo processo de exclusão que em **C e D**, onde tanto as sentenças discrepantes que ultrapassam o valor da distância mediana quanto as sentenças de clusters rotulados coma intenção *others* são eliminadas.

Ao aplicar essas estratégias de filtragem, a qualidade dos dados melhorou, como pode ser observado, de acordo com as métricas *DBS* e *SS*. A Figura 8 mostra a variação da métrica *DBS*, e a Figura 9 mostra a variação da métrica *SS* para cada abordagem de embedding, à medida que os dados se aproximam do centróide, o valor *DBS* melhora (diminui o valor, compare os cenários **A, B e C**), bem como quando a intenção *outros* é removida (compare os cenários**D, E e F**), os valores também melhoram. Da mesma forma, para SS, quando os valores aumentam, há uma melhoria na qualidade dos clusters após a remoção dos outliers.

O BERTopic, já agrupa todos os dados que ele considera *outiliers*, em um tópico especifico, porém como é explicado na Seção: 4.7, foi aplciado o mesmo processo de refinamento de dados nos tópicos gerados, para isso foi considerado a métrica *Probability*, gerada pelo BERTopic.

Figura 9 – Silhouette scores for datasets variations.



Fonte: elaborado pelo autor (2024).

Foi decidido construir o modelo de intenção *NLU* usando o conjunto de dados **F** como conjunto de treinamento, pois demonstrou os melhores valores para *DBS* e *SS*. Lembre-se de que cada modelo de embeddings produziu clusters com elementos distintos. Portanto, já é esperado que o número de sentenças restantes dentro dos clusters após a remoção dos outliers seja diferente para cada modelo de embedding, conforme mostrado na Tabela 8.

Tabela 8 – Número de sentenças após remoção de outliers na clusterização

Embedding	Número de sentenças
BERTimbau	8837
FLAIR	8840
Glove	8900
LaBSE	8007
MUSE	6928

Fonte: elaborado pelo autor.

A Tabela 9, apresenta a distribuição das sentenças para o BERTopic.

Em outras palavras, o número de sentenças no conjunto **treinamento/teste** varia para cada modelo de representação de embedding. A partir de todo o conjunto de sentenças resultantes da clusterização usando cada modelo de embedding, conforme ilustrado na Tabela 8,

Tabela 9 – Número de sentenças após remoção de outliers com o BERTopic

Embedding	Número de sentenças
BERTimbau	5417
FLAIR	5532
Glove	5518
LaBSE	5211
MUSE	6639

Fonte: elaborado pelo autor.

foi conduzida uma análise das interseções entre eles. Observou-se que apenas **1068** sentenças eram comuns entre esses modelos. Consequentemente, foi selecionado aleatoriamente **300** sentenças dessa interseção para rotulagem manual. Estas sentenças rotuladas constituem o conjunto de validação e serão excluídas dos conjuntos de **treinamento** e **teste**. Os experimentos realizados com O BERTopic, foram realizados apos a clusterização, como os dados de validação já haviam sido rotulados manualmente e por ser um trabalho bem oneroso. As sentenças presentes nesse conjunto de dados foram excluídas do conjuntos gerado pelo BERTopic, enttão o conjunto de dados usados para treinamento e teste no BERTopic, ficou com a distribuição apresentada na Tabela: 10

Tabela 10 – Número de sentenças após remoção de outliers com o BERTopic excluindo os dados de validação

Embedding	Número de sentenças
BERTimbau	5288
FLAIR	5394
Glove	5413
LaBSE	5083
MUSE	6490

Fonte: elaborado pelo autor.

A Tabela 11 mostra os valores de Acurácia alcançados pelos modelos de classificação de intenções construídos usando a arquitetura de rede neural apresentada na seção 4.8 implementada com Keras, a partir do conjunto de dados gerados pela clusterização.

Tabela 11 – Métricas de Resultado (Macro) para os Modelos de Classificação de Intenção baseados em rede neural feed-forward treinados com os dados provenientes da clusterização

Embedding	Precisão	Recall	F1-Score	Accurácia	MCC
BERTimbau	0.8912	0.9001	0.8954	0.9251	0.8677
FLAIR	0.9285	0.9150	0.9216	0.9450	0.8824
Glove	0.9307	0.9189	0.9244	0.9698	0.9116
LaBSE	0.9722	0.9676	0.9697	0.9749	0.9582
MUSE	**0.9846**	**0.9843**	**0.9844**	**0.9874**	**0.9792**

Fonte: elaborado pelo autor.

As métricas apresentadas na Tabela 11 são relativas ao desempenho dos modelos de classificação de intenção usando a arquitetura de Rede Neural, performada sobre os dados de teste que representam 30% do conjunto de dados, que foram anotados aplicando o processo de clusterização. Todos os modelos obtiveram bom desempenho, principalmente aqueles gerados com a representação de sentenças através de embeddings como Glove, LaBSE e MUSE, que também superam ligeiramente os demais embeddings nas métricas de *DBS* e *SS*. Já na Tabela 12, é apresentado os valores de Acurácia alcançados pelos modelos de classificação de intenções construídos usando a arquitetura de rede neural apresentada na seção 4.8 implementada com Keras, a partir do conjunto de dados gerados pela aplicação da ferramenta BERTopic.

Tabela 12 – Métricas de Resultado (Macro) para os Modelos de Classificação de Intenção baseados em rede neural feed-forward treinados com os dados provenientes da aplicação da ferramenta BERTopic

Embedding	Precisão	Recall	F1-Score	Accurácia	MCC
BERTimbau	0.9370	0.9401	0.9385	0.9471	0.9186
FLAIR	0.9642	0.9342	0.9472	0.**9741**	0.**9606**
Glove	0.9320	0.8832	0.9052	0.9329	0.8948
LaBSE	0.**9713**	0.**9572**	0.**9638**	0.9718	0.9526
MUSE	0.8538	0.8045	0.8270	0.8680	0.7734

Fonte: elaborado pelo autor.

As métricas apresentadas na Tabela 12 são relativas ao desempenho dos modelos de classificação de intenção usando a arquitetura de Rede Neural, performada sobre os dados de teste que representam 30% do conjunto de dados, que foram anotados aplicando a ferramenta BERTopic. Assim como na clusterização, os modelos obtiveram bom desempenho, houve uma melhora nas métricas para os modelos treinados usando o BERTimbau e FLAIR, porém para o Glove, LaBSE e MUSE tiveram uma leve piora em relação aos modelos gerados usando a clusterização, contudo ainda obtiveram bons resultados.

Com base na análise realizada, foi constatado que os modelos que receberam avaliação obtiveram pontuações mais altas quando avaliados a métrica *MCC*. É importante notar que uma pontuação 1 no *MCC* representa uma previsão perfeita, portanto, os modelos que foram avaliados chegaram muito perto de atingir esse ideal.

Também foram avaliados os modelos de classificação obtidos do Rasa conforme explicado na Seção 4.8. A Tabela 13 mostra os valores de Precisão, Recall, F1-score, Acurácia e *MCC* referentes à previsão de intenções sobre os dados de teste. Vale a pena ressaltar que não foram construídos modelos com a ferramenta Rasa, usando os dados rotulados com o BERTopic,

devido a questões de recursos disponiveis para montar o ambiente necessario para realizar o treinamento.

Tabela 13 – Métricas de resultado (macro) para os modelos de classificação de intenção Rasa, com os dados rotulado aplicando a clusterização.

Embedding	Precisão	Recall	F1-Score	Accurácia	MCC
BERTimbau	0.8842	0.8733	0.8785	0.9157	0.8454
FLAIR	0.9015	0.9014	0.9013	0.9372	0.8663
Glove	0.9163	0.9070	0.9116	0.9671	0.9028
LaBSE	0.9474	0.9370	0.9418	0.9546	0.9228
MUSE	**0.9575**	**0.9493**	**0.9530**	**0.9643**	**0.9408**

Fonte: elaborado pelo autor.

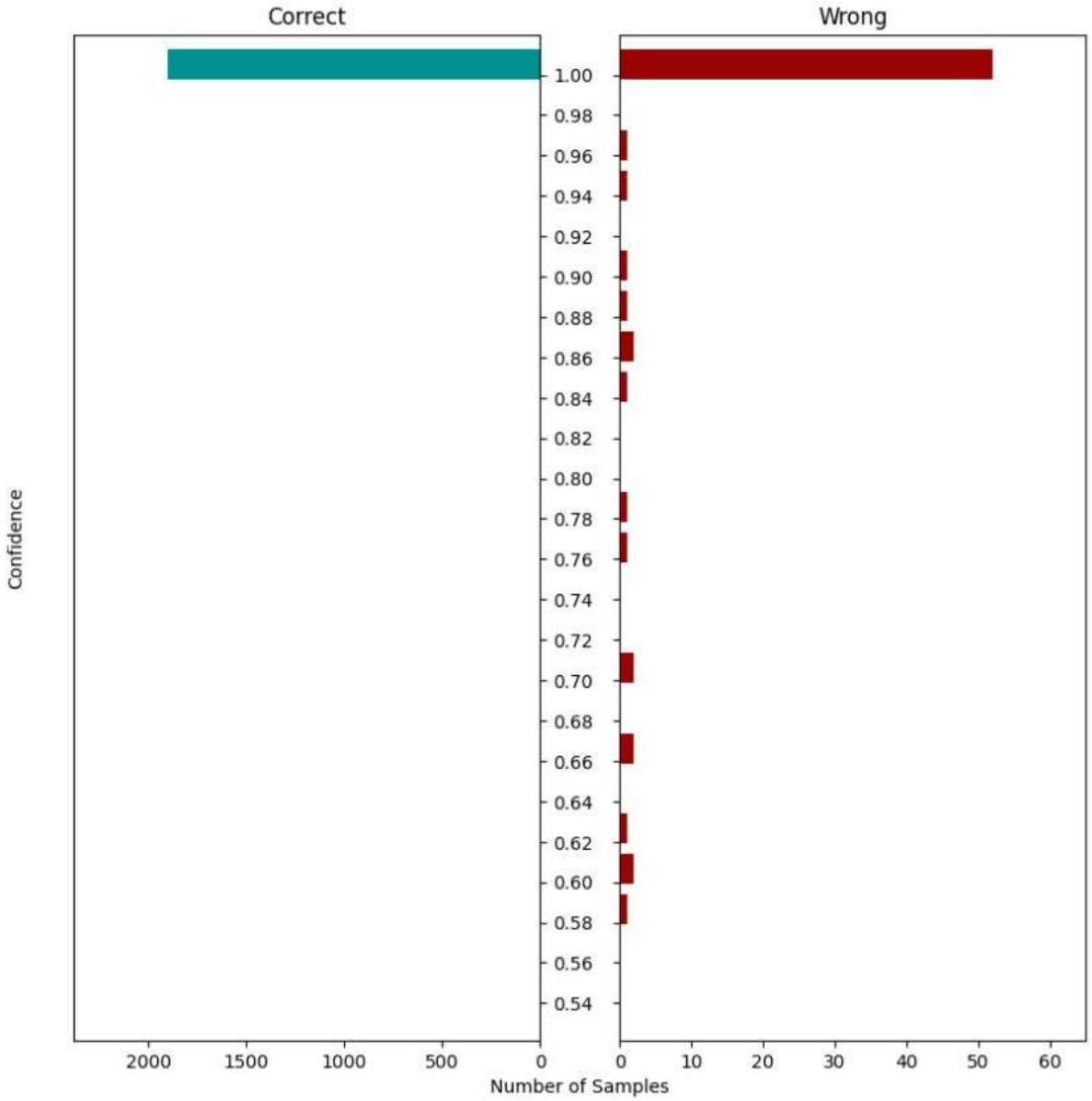
Foi percebido novamente quee os melhores modelos de classificação de intenções foram gerados usando Glove, LaBSE e MUSE como modelos de representação de embedding. Como pode se visto nas Tabelas 11 e 13, nas métricas de classificação de intenções, a representação de embedding utilizada no modelo de classificação que obteve o os melhores valores foram MUSE, que é um modelo de embedding de frases.

A Figura 10 contém o histograma que mostra a distribuição da classificação de intenções para o modelo treinado com MUSE.

No lado esquerdo está a distribuição das classificação feitas corretamente e no lado direito, as feitas incorretamente. As classificação são distribuídas de acordo com o grau de confidência. Pode-se observar que quase todas as classificações realizadas corretamente tiveram nível de confidência igual a 100%. Ainda assim, algumas classificações incorretas tiveram o mesmo nível de confidência, isso pode ser devido à sobreposição significativa de clusters e à distância das amostras de diferentes clusters mostradas durante a aplicação da *SS*. Porém, este modelo possui todas as intenções definidas e apresenta o melhor resultado comparado aos demais embeddings.

Com base nos resultados apresentados nesta seção, abordar a segunda questão de pesquisa (**RQ2**) envolve explorar várias estratégias de remoção de valores discrepantes e implementar duas redes neurais para construir o modelo de classificação de intenções. Com essas abordagens, foram alcaçados resultados promissores, com o *MCC* próximo de 1 para o modelo *NLU*.

Figura 10 – Histograma de previsão de intenções utilizando o modelo NLU treinado com o embedding MUSE na plataforma Rasa.



Fonte: elaborado pelo autor (2024).

5.4 Análise da representação de embeddings

Nesta seção, será discutido sobre os resultados orientados pela questão de pesquisa: **(RQ3)** *A representação de embedding de textos usada para a etapa de agrupamento e rotulagem poderia auxiliar no treinamento de um classificador de intenções?* pretende-se descobrir se os embeddings empregados para criar os grupos ainda podem treinar efetivamente o classificador de intenção, ou seja, usando o embedding como uma camada pré-treinada através da rede de treinamento.

Uma linha de pesquisa em *NLP* oferece resultados experimentais comparativos

dos métodos para que os pesquisadores possam determinar a emebedding mais adequada para seu problema com base na análise comparativa. Os artigos (Boggust *et al.*, 2022; Toshevska *et al.*, 2020) fornecem diferentes comparações entre vetores de embedding de palavras para garantir a qualidade da representação de palavras antes do uso em uma tarefa de aprendizado de máquina. Os métodos de avaliação são classificados em duas categorias principais: intrínsecos e extrínsecos (Zhai *et al.*, 2016; Qiu *et al.*, 2018). A avaliação intrínseca é independente de uma tarefa específica de *NLP*, avaliando diretamente as relações de sintaxe ou semântica entre palavras. Por exemplo, avaliando a distância entre palavras e frases. O método extrínseco de vetores de palavras é a avaliação integrada em uma tarefa de *NLP*, como inferência de linguagem natural ou análise de sentimento, escolhida como método de avaliação. Normalmente, as avaliações de embedding de palavras coletam Precisão e F1 Score entre outras métricas.

Considerando a qualidade do grupo, todos os modelos de representação de embedding têm um desempenho ligeiramente igual (consulte a tabela 4); no entanto, as métricas MUSE para *DBS* e *SS* superaram as outras (veja as figuras 8 e 9) quando foram removidos os outliers. O mesmo se aplica à precisão dos modelos de classificação de intenções (Tabela 11 e 13) usando MUSE para representar as sentenças. Assim, os modelos de embeddings empregados nesses experimentos demonstram resultados eficazes durante a fase de agrupamento. Essas descobertas estão bem alinhadas porque esses embeddings continuam a ser efetivamente utilizados como uma camada de embedding pré-treinada na rede do modelo de classificação de intenção, especialmente o MUSE.

5.5 Análise do potencial erro de rotulagem introduzido pela clusterização

Nesta seção, é abordado os experimentos realizados para analisar a questão de pesquisa: **(RQ4)**. Dado que a fase de agrupamento é uma técnica não supervisionada, existe a possibilidade de introdução de erros de rotulagem durante esta etapa no classificador de intenção *NLU*. O modelo de classificação de intenções desse trabalho é treinado com dados rotulados usando uma abordagem não supervisionada. É crucial observar que no aprendizado não supervisionado, onde os dados usados para o aprendizado carecem de informações sobre a saída "correta", existe o risco de possíveis erros de rotulagem serem incorporados ao conjunto de treinamento, o que pode enganar o modelo de classificação de intenção.

Foi utilizado o conjunto de dados de validação composto por **300** sentenças, conforme descrito na seção 5.3, para analisar os erros potenciais introduzidos pela fase de agrupamento

nessa abordagem. É importante observar que esse conjunto de validação foi anotado manualmente e os rótulos de intenção dos textos não vêm da fase de agrupamento. As tabelas 14, 15 e 16 apresentam os resultados de qualidade alcançados pelos modelos de classificação de intenções quando foi aplicado ao conjunto de dados de validação, tanto para o modelo baseado na rede neural *feed-forward* treinado com os dados rotulados tanto pela clusterização como pela aplicação do BERTopic e aquele treinado com o Rasa usando os dados rotulado por meio da clusterização.

Tabela 14 – Métricas de Resultado (Macro) para o **conjunto de validação rotulado manualmente** para os Modelos de Classificação de Intenção baseados em rede neural *feed-forward* Treinados com os dados anotados pelo processo de clusterização.

Embedding	Precisão	Recall	F1-Score	Accurácia	MCC
BERTimbau	0.9427	0.8793	0.9073	0.9529	0.9158
FLAIR	**0.9800**	0.9164	0.9449	**0.9630**	0.9341
Glove	0.8769	0.7115	0.7564	0.8653	0.7575
LaBSE	0.9454	0.9360	0.9338	**0.9630**	**0.9350**
MUSE	0.9703	**0.9376**	**0.9496**	**0.9630**	0.9348

Fonte: elaborado pelo autor.

Tabela 15 – Métricas de Resultado (Macro) para o **conjunto de validação rotulado manualmente** para os Modelos de Classificação de Intenção baseados em rede neural *feed-forward* Treinados com os dados rotulados pela aplicação do BERTopic

Embedding	Precisão	Recall	F1-Score	Accurácia	MCC
BERTimbau	0.9596	0.8793	0.9073	0.9596	0.9293
FLAIR	**0.9882**	0.9115	**0.9452**	**0.9697**	0.9463
Glove	0.8383	0.8364	0.8312	0.9057	0.8322
LaBSE	0.9479	**0.9477**	0.9425	**0.9697**	**0.9467**
MUSE	0.8055	0.8240	0.7883	0.8788	0.7941

Fonte: elaborado pelo autor.

Tabela 16 – Métricas de Resultado (Macro) para o **conjunto de validação rotulado manualmente** para os modelos de classificação de intenção Rasa.

Embedding	Precisão	Recall	F1-Score	Accurácia	MCC
BERTimbau	0.9325	0.7900	0.8347	0.9360	0.8854
FLAIR	0.9488	0.8908	0.9119	0.9360	0.8862
Glove	0.8796	0.6892	0.7352	0.8620	0.7521
LaBSE	**0.9582**	**0.9419**	**0.9451**	**0.9663**	**0.9408**
MUSE	0.9536	0.9182	0.9308	0.9596	0.9288

Fonte: elaborado pelo autor.

Como pode-se observar, os modelos treinados com os embedding LaBSE e MUSE obteveram os melhores resultados. Semelhante aos resultados obtidos com os dados de teste rotulados por meio do agrupamento. Porém, é importante destacar que o GLOVE obteve

uma diminuição nos valores das métricas utilizando este conjunto de dados de validação, em comparação com os dados de teste rotulados através do agrupamento.

Comparando as Tabelas 11 com a 14, as Tabelas 12 com a 15 e as tabelas 13 com a 16, pode-se ver facilmente que a fase de agrupamento adiciona alguns erros de rotulagem que enganaram os classificadores de intenções. Porém, ainda assim foram alcançados resultados competitivos em termos de precisão dos modelos treinados. Portanto, ainda é possível se apropria do conjunto de dados anotado pela abordagem não supervisionada de agrupamento para treinar um classificador de intenção de chatbot baseado em COVID-19. O método é genérico e pode ser aplicado a qualquer outro domínio ou doença, como gripe e dengue. Vale ressaltar que anotar um grande conjunto de dados é uma tarefa desafiadora e demorada. Foram tomadas medidas para mitigar o erro de rotulagem que poderia ser introduzido através do processo de agrupamento na clusterização com o K-means e geração dos topicos com o BERTopic. Foram usadas embeddings de palavras e de sentenças para comparação. Foram feitas análises da *SS* e o *DBS* para ajustar o número e tamanho dos clusters. Foi realizado a Inspeção manualmente dos grupos por meio da visualização em nuvem de palavras. Além disso, foram removidos os valores discrepantes do conjunto de treinamento.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, foi abordado o problema de como rotular intenções em diálogos de um chatbot voltado para o atendimento de pacientes da COVID-19 e como criar e treinar um modelo de classificação de intenções *NLU* para chatbots. Foram experimentados diferentes modelos de embeddings para representar os textos nos diálogos para aplicar neste problema. Os modelos de embeddings analisados foram BERTimbau, FLAIR, Glove, LaBSE e MUSE. As representações vetoriais das sentenças foram passadas por uma fase de agrupamento, na qual foi aplicado duas variações uma com um algoritmo de clusterização (K-means) e outra com a ferramenta BERTopic, para agrupar sentenças com significados semelhantes em grupos. Os rótulos foram atribuídos por meio de inspeção visual (nuvens de palavras e t-SNE), referentes às intenções de cada grupo, realizando assim a rotulagem semiautomática.

Após rotular os dados com os grupos (cada um representa uma intenção), um processo de refinamento de dados foi aplicado para melhorar a qualidade dos conjuntos de dados rotulados por cada modelo de embedding. Este processo de refinamento consistiu em descartar as sentenças distantes (em termos de distância do cosseno) do centróide de cada cluster para o K-means e as sentenças com menor probabilidade de pertencer ao tópico com o BERTopic. Em seguida, foram aplicadas abordagens baseadas em limites. Em um deles foram retiradas apenas sentenças classificadas como outliers; no outro, foram retiradas as sentenças com distância ao centróide de seu cluster maior que a mediana (considerando a distribuição de distância dentro do cluster), analogamente foi realizado a distribuição das probabilidades de cada sentença pertencer a determinado tópico do BERTopic e foram descartadas as que tinham uma probabilidade menor do que a mediana encontrada para esta distribuição. As métricas referentes a SS e DBS foram avaliadas neste processo. Ao final desta fase, comprovou-se que os grupos com apenas as sentenças mais próximas ou com maior probabilidade obtiveram melhores resultados para essas métricas.

Após o refinamento, foram construídos modelos de classificação de intenção para cada conjunto de dados rotulado (de acordo com os grupos). Além disso, é essencial observar que cada modelo de embedding gera um conjunto de grupos diferente. Cada conjunto de grupos é usado para treinar um modelo de classificação de intenção. Foi experimentado duas arquiteturas diferentes para treinar os modelos de classificação de intenções. Um é baseado na estrutura de código aberto Rasa e outro é baseado em redes neurais profundas. Após a construção dos modelos, também foram validados com dados não vistos pelos modelos, ou seja, dados rotulados

manualmente. Percebeu-se que a rotulagem semiautomática (o agrupamento) adicionou erros de rotulagem nos modelos de classificação de intenção. Porque alguns deles obtiveram excelentes resultados com os dados dos testes; No entanto, com os dados rotulados manualmente, obtiveram valores de precisão muito diferentes. Porém, de forma geral, os resultados obtidos com todos os modelos construídos nos testes e na validação em termos de precisão ficaram acima de 86%. Além disso, é essencial ressaltar que rotular uma grande quantidade de dados para treinar essas redes profundas consumiria muito tempo.

Este trabalho mostra uma visão geral de um chatbot, suas classificações e como ele pode ser utilizado. Foi explicado sobre o componente *NLU* do chatbot, a importância e o processo de construção. Um conjunto de dados de diálogos entre profissionais de saúde e pacientes foi utilizado para aplicar aprendizagem não supervisionada (agrupamento) para encontrar as classes por meio da inspeção visual dos grupos encontrados. Essas classes foram usadas para treinar modelos de classificação de intenções. Os resultados desses modelos foram avaliados e, de modo geral, obtiveram bons resultados. Ainda assim, foi utilizado dados rotulados manualmente para validar os modelos, houve queda nos resultados, mas ainda assim foram satisfatórios.

Em trabalhos futuros, pretende-se explorar com mais profundidade sobre a aplicação da ferramenta BERTopic para rotular os diálogos. Identificar o motivo pelo qual, foram gerados tantos grupos, tanto usando o K-means quanto como o BERTopic, e como melhorar a qualidade do agrupamento, obtendo grupos com elementos mais homogêneos dentro do grupo e mais distintos dos outros grupos. Pretende-se ainda aplicar o mesmo pipeline usado nesse trabalho para conjuntos de dados pertencentes a outros domínios.

REFERÊNCIAS

ABDELLATIF, A.; COSTA, D.; BADRAN, K.; ABDALKAREEM, R.; SHIHAB, E. Challenges in chatbot development: A study of stack overflow posts. In: **Proceedings of the 17th international conference on mining software repositories**. [*S. l.*: *s. n.*], 2020. p. 174–185.

ADAMOPOULOU, E.; MOUSSIADES, L. An overview of chatbot technology. In: MAGLOGIANNIS, I.; ILIADIS, L.; PIMENIDIS, E. (Ed.). **Artificial Intelligence Applications and Innovations**. Cham: Springer International Publishing, 2020. p. 373–383. ISBN 978-3-030-49186-4.

AGGARWAL, C. C.; REDDY, C. K. Data clustering. **Algorithms and applications. Chapman&Hall/CRC Data mining and Knowledge Discovery series, Londra** , Citeseer, 2014.

AGUIAR, L. A.; CRUZ, L. A.; SILVA, T. L. C. da; CARMO, R. A. F. do; PAIXAO, M. H. E. Large-scale translation to enable response selection in low resource languages: A covid-19 chatbot experiment. In: SBC. **Anais do XXXVII Simpósio Brasileiro de Bancos de Dados**. [*S. l.*], 2022. p. 203–215.

AKBIK, A.; BERGMANN, T.; BLYTHE, D.; RASUL, K.; SCHWETER, S.; VOLLGRAF, R. Flair: An easy-to-use framework for state-of-the-art nlp. In: **Proceedings of NAACL (Demonstrations)**. [*S. l.*: *s. n.*], 2019. p. 54–59.

BOGGUST, A.; CARTER, B.; SATYANARAYAN, A. Embedding comparator: Visualizing differences in global structure and local neighborhoods via small multiples. In: **27th International Conference on Intelligent User Interfaces**. [*S. l.*: *s. n.*], 2022. p. 746–766.

CER, D.; YANG, Y.; KONG, S.-y.; HUA, N.; LIMTIACO, N.; JOHN, R. S.; CONSTANT, N.; GUAJARDO-CÉSPEDES, M.; YUAN, S.; TAR, C. *et al.* Universal sentence encoder**arXiv preprint arXiv:1803.11175**, 2018.

CONNEAU, A.; KIELA, D.; SCHWENK, H.; BARRAULT, L.; BORDES, A. Supervised learning of universal sentence representations from natural language inference data. In: **Proceedings of EMNLP**. [*S. l.*: *s. n.*], 2017. p. 670–680.

DAVIES, D. L.; BOULDIN, D. W. A cluster separation measure. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, PAMI-1, n. 2, p. 224–227, 1979.

DEFAYS, D. An efficient algorithm for a complete link method. **The Computer Journal**, Oxford University Press, v. 20, n. 4, p. 364–366, 1977.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

ESTER, M.; KRIEGEL, H.-P.; S, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: **KDD**. [*S. l.*: *s. n.*], 1996. v. 96, p. 226–231.

FAZZINGA, B.; GALASSI, A.; TORRONI, P. **An Argumentative Dialogue System for COVID-19 Vaccine Information**. 2021.

FENG, F.; YANG, Y.; CER, D.; ARIVAZHAGAN, N.; WANG, W. Language-agnostic bert sentence embedding. **arXiv preprint arXiv:2007.01852**, 2020.

GAGLO, K.; DEGBOE, B. M.; KOSSINGOU, G. M.; OUYA, S. Proposal of conversational chatbots for educational remediation in the context of covid-19. In: **2021 23rd International Conference on Advanced Communication Technology (ICACT)**. [*S. l.*: *s. n.*], 2021. p. 354–358.

GALASSI, A.; LIPPI, M.; TORRONI, P. Attention in natural language processing. **IEEE TNNLS**, IEEE, 2020.

GAO, J.; GALLEY, M.; LI, L. Neural approaches to conversational ai. In: **The 41st International ACM SIGIR**. [*S. l.*: *s. n.*], 2018. p. 1371–1374.

GENSIM. **Gensim BERTopic Documentation**. 2022. Disponível em: https://bertopic.readthedocs.io/en/latest/.

GROOTENDORST, M. Bertopic: Neural topic modeling with a class-based tf-idf procedure. **arXiv preprint arXiv:2203.05794**, 2022.

HAN, B.; LIU, L.; OMIECINSKI, E. Neat: Road network aware trajectory clustering. In: IEEE. **2012 IEEE 32nd International Conference on Distributed Computing Systems**. [*S. l.*], 2012. p. 142–151.

HARRIS, Z. S. Distributional structure. **Word**, Taylor & Francis, v. 10, n. 2-3, p. 146–162, 1954.

HIEN, H. T.; CUONG, P.-N.; NAM, L. N. H.; NHUNG, H. L. T. K.; THANG, L. D. Intelligent assistants in higher-education environments: the fit-ebot, a chatbot for administrative and learning support. In: **Proceedings of the ninth international symposium on information and communication technology**. [*S. l.*: *s. n.*], 2018. p. 69–76.

HOAGLIN, D. C.; IGLEWICZ, B. Fine-tuning some resistant rules for outlier labeling. **Journal of the American Statistical Association** , [American Statistical Association, Taylor Francis, Ltd.], v. 82, n. 400, p. 1147–1149, 1987. ISSN 01621459. Disponível em: http://www.jstor.org/stable/2289392.

IYYER, M.; MANJUNATHA, V.; BOYD-GRABER, J.; III, H. D. Deep unordered composition rivals syntactic methods for text classification. In: **Proceedings of the 53rd ACL-IJCNLP (volume 1: Long papers)**. [*S. l.*: *s. n.*], 2015. p. 1681–1691.

JUNIOR, V. O. D. S.; BRANCO, J. A. C.; OLIVEIRA, M. A. D.; SILVA, T. L. C. D.; CRUZ, L. A.; MAGALHãES, R. P. A natural language understanding model covid-19 based for chatbots. In: **2021 IEEE 21st International Conference on Bioinformatics and Bioengineering (BIBE)**. [*S. l.*: *s. n.*], 2021. p. 1–7.

KHANNA, A.; PANDEY, B.; VASHISHTA, K.; KALIA, K.; PRADEEPKUMAR, B.; DAS, T. A study of today's ai through chatbots and rediscovery of machine intelligence.**International Journal of u-and e-Service, Science and Technology**, v. 8, n. 7, p. 277–284, 2015.

KIROS, R.; ZHU, Y.; SALAKHUTDINOV, R. R.; ZEMEL, R.; URTASUN, R.; TORRALBA, A.; FIDLER, S. Skip-thought vectors. In: **NIPS**. [*S. l.*: *s. n.*], 2015. p. 3294–3302.

KLEIN, A. Z.; MAGGE, A.; O'CONNOR, K.; AMARO, J. I. F.; WEISSENBACHER, D.; HERNANDEZ, G. G. Toward using twitter for tracking covid-19: a natural language processing pipeline and exploratory data set. **Journal of medical Internet research**, JMIR Publications Inc., Toronto, Canada, v. 23, n. 1, p. e25314, 2021.

KOREN, Y.; CARMEL, D. Efficient algorithms for large-scale universal manifold approximation and projection. **Proceedings of the 20th international conference on World Wide Web**, p. 574–583, 2003.

KUCHERBAEV, P.; BOZZON, A.; HOUBEN, G.-J. Human-aided bots. **IEEE Internet Computing**, v. 22, n. 6, p. 36–43, 2018.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. In: PMLR. **ICML**. [*S. l.*], 2014. p. 1188–1196.

LEE, J. Y.; DERNONCOURT, F**Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks**. 2016.

LEI, H.; LU, W.; JI, A.; BERTRAM, E.; GAO, P.; JIANG, X.; BARMAN, A. **COVID-19 Smart Chatbot Prototype for Patient Monitoring**. 2021.

LI, I.; LI, Y.; LI, T.; ALVAREZ-NAPAGAO, S.; GARCIA-GASULLA, D.; SUZUMURA, T. What are we depressed about when we talk about covid-19: Mental health analysis on tweets using natural language processing. In: SPRINGER. **International Conference on Innovative Techniques and Applications of Artificial Intelligence**. [*S. l.*], 2020. p. 358–370.

LI, J.; SUN, A.; HAN, J.; LI, C. A survey on deep learning for named entity recognition.**IEEE TKDE**, IEEE, 2020.

LIU, J.; LI, Y.; LIN, M. Review of intent detection methods in the human-machine dialogue system. In: IOP PUBLISHING. **Journal of Physics: Conference Series**. [*S. l.*], 2019. v. 1267, n. 1, p. 012059.

LIU, W.; WANG, Z.; LIU, X.; ZENG, N.; LIU, Y.; ALSAADI, F. E. A survey of deep neural network architectures and their applications. **Neurocomputing**, Elsevier, v. 234, p. 11–26, 2017.

LIU, X.; HE, P.; CHEN, W.; GAO, J. Multi-task deep neural networks for natural language understanding. **arXiv preprint arXiv:1901.11504**, 2019.

MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne**Journal of machine learning research**, v. 9, n. 11, 2008.

MCINNES, L.; HEALY, J.; MELVILLE, J. Umap: Uniform manifold approximation and projection. **arXiv preprint arXiv:1802.03426**, 2018.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed representations of words and phrases and their compositionality. In:**NIPS**. [*S. l.*: *s. n.*], 2013. p. 3111–3119.

MINER, A. S.; LARANJO, L.; KOCABALLI, A. B. Chatbots in the fight against the covid-19 pandemic. **NPJ digital medicine**, Nature Publishing Group, v. 3, n. 1, p. 1–4, 2020.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina**Sistemas inteligentes-Fundamentos e aplicações**, Manole, v. 1, n. 1, p. 32, 2003.

NASEEM, U.; RAZZAK, I.; KHAN, S. K.; PRASAD, M. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. **Transactions on Asian and Low-Resource Language Information Processing**, ACM New York, NY, v. 20, n. 5, p. 1–35, 2021.

NASSIF, A. B.; SHAHIN, I.; ATTILI, I.; AZZEH, M.; SHAALAN, K. Speech recognition using deep neural networks: A systematic review. **IEEE access**, IEEE, v. 7, p. 19143–19165, 2019.

NIMAVAT, K.; CHAMPANERIA, T. Chatbots: An overview. types, architecture, tools and future possibilities. **International Journal for Scientific Research & Development**, v. 5, n. 7, p. 1019–1024, 2017.

PEIKARI, M.; SALAMA, S.; NOFECH-MOZES, S.; MARTEL, A. L. A cluster-then-label semi-supervised learning approach for pathology image classification. **Scientific reports**, Springer, v. 8, n. 1, p. 1–13, 2018.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **Proceedings of EMNLP**. [*S. l.*: *s. n.*], 2014. p. 1532–1543.

QIU, Y.; LI, H.; LI, S.; JIANG, Y.; HU, R.; YANG, L. Revisiting correlations between intrinsic and extrinsic evaluations of word embeddings. In: **Chinese computational linguistics and natural language processing based on naturally annotated big data**. [*S. l.*]: Springer, 2018. p. 209–221.

RASA. **Introduction to Rasa Open Source** . 2022. https://rasa.com/docs. Accessed: 2022-02-03.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. **arXiv preprint arXiv:1908.10084**, 2019.

ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. **JCAM**, v. 20, p. 53–65, 1987. ISSN 0377-0427. Disponível em: https://www.sciencedirect.com/science/article/pii/0377042787901257.

RUSSELL, S. J.; NORVIG, P. **Inteligência artificial**. [*S. l.*]: Elsevier, 2004.

SCIKIT-LEARN. **Metrics and scoring: quantifying the quality of predictions** . 2023. https://scikit-learn.org/stable/modules/model_evaluation.html. Accessed: 2023-06-10.

SHAWAR, B. A.; ATWELL, E. Chatbots: are they really useful? In:**Ldv forum**. [*S. l.*: *s. n.*], 2007. v. 22, n. 1, p. 29–49.

SILVA, T. L. C. da; LETTICH, F.; MACÊDO, J. A. F. de; ZEITOUNI, K.; CASANOVA, M. A. Online clustering of trajectories in road networks. In: IEEE. **2020 21st IEEE International Conference on Mobile Data Management (MDM)**. [*S. l.*], 2020. p. 99–108.

SILVA, T. L. Coelho da; FERREIRA, M. G. F.; MAGALHAES, R. P.; MACÊDO, J. A. F. de; ARAÚJO, N. da S. Rastreador de sintomas da covid19. **SBBD**, 2020.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. BERTimbau: pretrained BERT models for Brazilian Portuguese. In: **9th BRACIS**. [*S. l.*: *s. n.*], 2020.

TOSHEVSKA, M.; STOJANOVSKA, F.; KALAJDJIESKI, J. Comparative analysis of word embeddings for capturing word similarities. **arXiv preprint arXiv:2005.03812**, 2020.

VASSILVITSKII, S.; ARTHUR, D. k-means++: The advantages of careful seeding. In: **Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms**. [*S. l.*: *s. n.*], 2006. p. 1027–1035.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. In: **NIPS**. [*S. l.*: *s. n.*], 2017. p. 5998–6008.

WANG, L. L.; LO, K.; CHANDRASEKHAR, Y.; REAS, R.; YANG, J.; EIDE, D.; FUNK, K.; KINNEY, R.; LIU, Z.; MERRILL, W. *et al.* Cord-19: The covid-19 open research dataset. **ArXiv**, ArXiv, 2020.

WANG, Y.-Y.; DENG, L.; ACERO, A. Spoken language understanding. **IEEE Signal Processing Magazine**, IEEE, v. 22, n. 5, p. 16–31, 2005.

WELD, H.; HUANG, X.; LONG, S.; POON, J.; HAN, S. C. A survey of joint intent detection and slot filling models in natural language understanding. **ACM Computing Surveys (CSUR)**, ACM New York, NY, 2021.

WIETING, J.; BANSAL, M.; GIMPEL, K.; LIVESCU, K. Towards universal paraphrastic sentence embeddings. 2016.

XU, B.; XING, Z.; XIA, X.; LO, D. Answerbot: Automated generation of answer summary to developers' technical questions. In: IEEE. **2017 32nd IEEE/ACM international conference on automated software engineering (ASE)**. [*S. l.*], 2017. p. 706–716.

YANG, Y.; CER, D.; AHMAD, A.; GUO, M.; LAW, J.; CONSTANT, N.; ABREGO, G. H.; YUAN, S.; TAR, C.; SUNG, Y.-H. *et al.* Multilingual universal sentence encoder for semantic retrieval. **arXiv preprint arXiv:1907.04307**, 2019.

YAO, K.; ZWEIG, G.; HWANG, M.-Y.; SHI, Y.; YU, D. Recurrent neural networks for language understanding. In: **Interspeech**. [*S. l.*: *s. n.*], 2013. p. 2524–2528.

ZHAI, M.; TAN, J.; CHOI, J. Intrinsic and extrinsic evaluations of word embeddings. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [*S. l.*: *s. n.*], 2016. v. 30, n. 1.

ZHOU, M.; DUAN, N.; LIU, S.; SHUM, H.-Y. Progress in neural nlp: modeling, learning, and reasoning. **Engineering**, Elsevier, v. 6, n. 3, p. 275–290, 2020.