

UNIVERSIDADE PAULISTA

FABIO LUCAS SOUSA COLAÇO – N336EJ9

VITOR BRANDÃO SANTOS – D76JAB9

**USO DO PROCESSAMENTO DE LINGUAGEM NATURAL NO
DESENVOLVIMENTO DE UM ASSISTENTE VIRTUAL**

JUNDIAÍ

2022

FABIO LUCAS SOUSA COLAÇO – N336EJ9

VITOR BRANDÃO SANTOS – D76JAB9

**USO DO PROCESSAMENTO DE LINGUAGEM NATURAL NO
DESENVOLVIMENTO DE UM ASSISTENTE VIRTUAL**

Trabalho de conclusão de curso para
obtenção do título de graduação em
Ciência da Computação apresentado à
Universidade Paulista – UNIP.

Orientador: Prof. Peter Jandl Junior

JUNDIAÍ

2022

Biblioteca da Universidade Paulista - UNIP.

Guia de normalização para apresentação de trabalhos acadêmicos da Universidade Paulista: ABNT / Biblioteca da Universidade Paulista - UNIP; revisado e atualizado pelos Bibliotecários Alice Horiuchi e Rodrigo da C. Aglinskas. – 2019. 52 p.: il. color.

1. Normalização. 2. Trabalhos acadêmicos. 3. ABNT. I.
Biblioteca da Universidade Paulista - UNIP.

FABIO LUCAS SOUSA COLAÇO – N336EJ9

VITOR BRANDÃO SANTOS – D76JAB9

**USO DO PROCESSAMENTO DE LINGUAGEM NATURAL NO
DESENVOLVIMENTO DE UM ASSISTENTE VIRTUAL**

Trabalho de conclusão de curso para
obtenção do título de graduação em
Ciência da Computação apresentado à
Universidade Paulista – UNIP.

Aprovado em:

BANCA EXAMINADORA

_____/____/____

Prof. Clayton Valdo

Universidade Paulista – UNIP

_____/____/____

Prof. Peter Jandl Junior

Universidade Paulista – UNIP

_____/____/____

Prof.

Universidade Paulista – UNIP

RESUMO

Neste trabalho será realizado o estudo das tecnologias referentes a inteligência artificial, focado na área de assistente virtual, descrevendo sobre os estudos do processamento de linguagem natural, relatando como a máquina interpreta a linguagem humana e coleta informações processando-a, estudos sobre o aprendizado de máquina em uma inteligência artificial, juntamente com o ramo de aprendizagem profunda, utilizando a ferramenta TensorFlow, uma ferramenta de código aberto para aprendizagem de máquina. O TensorFlow também será utilizado para a criação de redes neurais, que são modelos computacionais interconectados simulando os neurônios do cérebro humano, sendo capaz de decifrar padrões, agrupar e classificar dados. Para demonstrar todas essas tecnologias, será desenvolvido uma assistente virtual. Tendo como princípio explorar a tecnologia para criar meios de facilitar e agilizar a interação homem e máquina, disponibilizando formas de realizar pesquisas, executar aplicativos, buscar algumas informações, realizando essas tratativas através do uso da voz, onde será utilizado o processamento de linguagem natural, com isso também será abrangido o reconhecimento de voz e a síntese de voz. Para tornar está comunicação possível será utilizado um kit de ferramentas, com um modelo de vocabulário em língua portuguesa, sendo utilizadas essas ferramentas para a criação do assistente, possibilitando a interação e também tornando o uso da tecnologia mais produtiva por meio das diferentes tarefas executadas e respondida com a sintetização de voz.

Palavras-chave: Assistente virtual, processamento de linguagem natural, PLN, reconhecimento de voz, síntese de voz, comandos de voz, aprendizado profundo, aprendizado de máquina, redes neurais.

ABSTRACT

In this work, the study of technologies related to artificial intelligence will be carried out, focused on the virtual assistant area, describing the studies of natural language processing, reporting how the machine interprets human language and collects information processing it, studies on the learning of machine in an artificial intelligence, together with the branch of deep learning, using the tool TensorFlow, an open-source tool for machine learning. TensorFlow will also be used to create neural networks, which are interconnected computational models simulating neurons in the human brain, being able to decipher patterns, group and classify data. To demonstrate all of these technologies, a virtual assistant will be developed. With the principle of exploring technology to create ways to facilitate and streamline human and machine interaction, providing ways to conduct research, run applications, search for some information, carrying out these negotiations through the use of voice, where natural language processing will be used, this will also include speech recognition and speech synthesis. To make communication possible use a toolkit, with a vocabulary model in Portuguese, using these tools to create the assistant, enabling interaction and also making the use of technology more productive through different tasks. performed and answered with speech synthesis.

Keywords: Virtual assistant, natural language processing, PLN, speech recognition, speech synthesis, voice commands, deep learning, machine learning, neural networks.

LISTA DE ILUSTRAÇÕES

Figura 1 – Figura 1: Representação do jogo	16
Figura 2 – Representação do perceptron.	24
Figura 3 – Termos algébricos threshold	25
Figura 4 – Diferentes pesos de evidências do perceptron.	25
Figura 5 - Árvore de Decisão para jogar Tênis	29
Figura 6 - Hipótese da árvore de decisão	29
Figura 7 – Hierarquia de Aprendizado de Máquina	30
Figura 8 – Rede Neural Simples e Rede Neural Profunda (Deep Learning)	33
Figura 9 – Marcos no desenvolvimento das redes neurais	34
Figura 10 – Resumo do projeto desenvolvido	36
Figura 11 – Representação de funcionamento biblioteca Google	38
Figura 12 – Implementação do Keras ao projeto	39
Figura 13 – Modelos de redes neurais utilizadas para treinamento do projeto	40
Figura 14 – Arquivo conjunto de dados do modelo criado	41
Figura 15 – Modelo depreciado	41
Figura 16 – Modelo de PLN implementado	42
Figura 17 – Etapas do projeto até a classificação	43
Figura 18 – Classificação em execução	43
Figura 19 – Exemplo de função categorizada por grupo	44
Figura 20 – Estágios para sintetização de voz	45
Figura 21 – Etapa inicial da implantação do processamento de linguagem natural	46
Figura 22 – Complementação da implantação do processamento de linguagem natural	47
Figura 23 – Comandos para treinamento	48
Figura 24 – Dados sem One Hot Enconding	51
Figura 25 – One Hot Encoding aplicado	52
Figura 26 – Tratamento dos dados para a implementação do One Hot Encoding	52
Figura 27 – Quantidade de épocas treinadas	53
Figura 28 – Demonstração perca e precisão	54
Figura 29 – Treinamento e classificação dos grupos	55
Figura 30 – Comando de horário	56

Figura 31 – Comando de data	56
Figura 32 – Comando de cotação de dólar	56
Figura 33 – Comando de cotação de euro	57
Figura 34 – Comando de abrir navegador	57
Figura 35 – Comando de abrir bloco de notas	58
Figura 36 – Comando de abrir calculadora	58
Figura 37 – Comando de pesquisas web	59
Figura 38 – Comando de navegação	59
Figura 39 – Comando de navegação 2	60
Figura 40 – Comando de abertura	60

SUMÁRIO

1	INTRODUÇÃO	11
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	INTELIGÊNCIA ARTIFICIAL	14
2.2	PROCESSAMENTO DE LINGUAGEM NATURAL	16
2.3	REDES NEURIAS ARTIFICIAIS	23
2.3.1	REDES NEURAIIS COM MULTICAMADAS	24
2.3.2	REPRESENTAÇÃO DO CONHECIMENTO	25
2.3.3	REDES NEURAIIS CONVOLUCIONAIS	28
2.4	APRENDIZADO DE MÁQUINA (MACHINE LEARNING)	28
2.5	APRENDIZADO PROFUNDO (DEEP LEARNING)	34
3	PROJETO	38
3.1	SPEECH RECOGNITION	39
3.2	GOOGLE SPEECH-TO-TEXT	39
3.3	TENSORFLOW	41
3.3.1	KERAS	41
3.4	CLASSIFICAÇÃO E COMANDOS	44
3.5	SÍNTESE DE VOZ	46
4	RESULTADOS E TESTES	48
4.1	PROCESSAMENTO DE LINGUAGEM NATURAL	48
4.2	CRIAÇÃO DE UM DATASET	50
4.3	TREINAMENTO	51
4.3.1	CATEGORIZAÇÃO POR GRUPO	54
4.4	EXECUÇÃO DE COMANDOS	55
4.4.1	HORÁRIO	55
4.4.2	DATA	56
4.4.3	COTAÇÃO ATUAL	56
4.4.4	SOFTWARES E SITES	57
4.4.5	PESQUISA	58
4.4.6	NAVEGAÇÃO	59
5	CONCLUSÃO	61

6 BIBLIOGRAFIA

62

1. INTRODUÇÃO

A inteligência virtual (ou IA) é um ramo de pesquisa da ciência da computação que busca construir algoritmos para desenvolver máquinas capazes de simular as capacidades do ser humano. Através da inteligência virtual é possível criar máquinas que possam ser inteligentes, ou seja, possuem a capacidade de pensar e aprender informações para tomarem decisões autônomas.

Segundo Araribóia (1988), a Inteligência Artificial é um campo que usa técnicas de programação que procuram, por intermédio de máquinas, resolver problemas do mesmo modo que um ser humano os resolveria. As técnicas de Inteligência Artificial procuram imitar mecanismos da natureza por intermédio de mecanismos tecnológicos cujo desenvolvimento foi baseado em mecanismos naturais.

Segundo McCarthy (2002), a Inteligência Artificial é a ciência e a tecnologia de construção de máquinas inteligentes, especialmente programas de computador. Relaciona-se com o uso de computadores para o entendimento e a exploração da inteligência humana, não se limitando, porém, a métodos biologicamente observáveis.

A utilização de sistemas inteligentes com o uso da inteligência artificial está cada vez mais frequente atualmente. Os estudos da IA estão em constante evolução, gerando conflito de pensamentos entre pessoas que apoiam a ideia de dar a capacidade de uma máquina tomar decisões por conta própria e pessoas que acreditam ser um ramo perigoso devido a possibilidade de erros que a máquina pode tomar em decorrência da capacidade de pensar e talvez futuramente superar a capacidade humana.

Uma grande utilização da IA atualmente está nas assistentes virtuais disponíveis em diversas plataformas. Uma assistente virtual é uma aplicação computacional que auxilia na resolução de problemas cotidianos, atendendo diversos públicos diferentes e suprimindo diversas dificuldades que exijam uma grande distribuição de tarefas e, aumentando a produtividade, substituindo tarefas que comumente eram executadas por seres humanos e, consequentemente, diminuindo os custos.

Segundo Maes (1994), uma assistente virtual trata-se do emprego de uma metáfora em que um assistente pessoal colabora com o usuário no mesmo ambiente de trabalho.

Uma aplicação clássica da assistente virtual atualmente, é realizar diversas funções domésticas, automatizando tarefas que antes necessitavam de uma atenção completa por parte do ser humano e que podem ser substituídas por ações de inteligência artificial.

Atualmente existem uma diversidade de assistentes virtuais disponíveis para uso. Como por exemplo, a Siri, da Apple é uma delas, assim como a Alexa, da Amazon, Cortana, da Microsoft, dentre outras.

Para que fosse necessário a criação de uma assistente virtual, foram considerados conjuntos de tecnologias de diversas áreas da Ciência da Computação. Alguns exemplos são:

Redes de comunicação de dados: Responsável pela segurança do transporte de dados entre as máquinas computacionais;

Banco de dados: Responsável por armazenar os dados, guardando informações para que possam ser buscadas e consultadas;

Processamento de linguagem natural: Responsável pela ligação entre “homem e máquina”. Tendo como objetivo permitir a comunicação do humano com a máquina, como se o humano tivesse se comunicando com outro humano;

Inteligência artificial: Responsável pela “inteligência da máquina”, sendo elaborada para espelhar-se na inteligência humana, tentando fazer a máquina pensar o mais próximo possível como um humano.

Com a combinação de áreas da computação, foi possível reunir diversas informações de fontes diferentes e unificar diversas tarefas distintas em uma única aplicação computacional, denominada como assistente virtual. Através do processamento de linguagem natural, é possível fazer com que a aplicação computacional possa reconhecer e interpretar a linguagem humana e executar uma determinada tarefa de acordo com o que foi interpretado, promovendo assim uma comunicação entre a máquina e o homem.

O objetivo principal deste trabalho de conclusão de curso é demonstrar os estudos das tecnologias referentes a assistente virtual, como o processamento de linguagem natural, a aprendizagem de máquina presente nesta área juntamente com o aprendizado profundo utilizando redes neurais e então promover o desenvolvimento de uma assistente virtual, com o intuito de otimizar o tempo em tarefas rotineiras e demonstrar como o uso de uma inteligência artificial pode auxiliar qualquer pessoa no dia a dia, e auxiliar ainda mais pessoas com algum tipo de

deficiência, como por exemplo deficientes visuais, que podem se comunicar com uma assistente para realizar tarefas que possuem maiores dificuldades de executarem.

Nos próximos capítulos deste trabalho será contemplado: A Fundamentação Teórica, onde irá conter os estudos das tecnologias necessárias para o desenvolvimento deste projeto, juntamente com seus fundamentos; Projeto, onde será demonstrado todo o desenvolvimento do projeto elaborado utilizando as tecnologias demonstradas na fundamentação teórica e os objetivos do trabalho; Resultados e Testes, com o desenvolvimento da solução, demonstrando os testes referentes ao projeto, validando e verificando a viabilidade dos objetivos do trabalho; e Conclusão, onde será apresentado todo o resultado final envolvendo todo o projeto, com os objetivos alcançados e a conclusão do grupo acerca do tema do trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será abordada a fundamentação teórica, sendo o principal embasamento de pesquisa deste projeto em relação a inteligência artificial, nomeadamente as principais ferramentas e tecnologias que cercam a área de Processamento de Linguagem Natural, como por exemplo, Redes Neurais, Machine Learning e Deep Learning, sendo apresentado também outros subtópicos particulares de suas tecnologias e bibliotecas.

2.1 INTELIGÊNCIA ARTIFICIAL

A inteligência artificial, ou IA, é um dos estudos dos campos de ciência mais recentes. Os estudos foram iniciados logo após a Segunda Guerra Mundial, durante a década de 1950.

Para Russel e Norvig (2013), os filósofos já muito antes dos computadores procuravam a resposta para o funcionamento da mente humana, o mesmo objetivo da inteligência artificial. Os filósofos denominaram de hipótese de IA fraca a possibilidade de as máquinas agirem de forma inteligente e definiram como hipótese de IA forte a ascensão de que as máquinas fazem o que estão pensando. A maior parte dos pesquisadores de IA levam como base a hipótese de IA fraca para realizar os estudos.

Segundo McCulloch e Pitts (1943), o primeiro trabalho de inteligência artificial foi baseado em três fontes: o conhecimento de filosofia básica e da função dos neurônios no cérebro, uma análise formal da lógica proposicional, e a teoria da computação. Foi proposto um modelo de neurônios artificiais onde cada neurônio caracteriza por estar “ligado” ou “desligado”, onde o estado “ligado” é ativado quando há um estímulo por um número suficiente de neurônios vizinhos.

O estado de um neurônio era caracterizado como equivalente em termos concretos a uma proposição que definia seu estímulo adequado. Através desse modelo, os pesquisadores mostraram que qualquer função computável podia ser calculada por uma determinada quantidade de neurônios conectados formando uma rede e que todos os conectivos lógicos podem ser implementados por estruturas de redes simples. Também foi sugerido que se as redes fossem adequadas e definidas corretamente acerca dos neurônios elas seriam capazes de aprender. As intensidades de conexão entre os neurônios sofreram uma regra de atualização

para que fosse mais intensa e sua regra, conhecida como aprendizado de Hebb, continua como um modelo influente atualmente. Essa regra consiste em se os neurônios em cada lado de uma conexão (ou sinapse) são ativados simultaneamente, então a “força” daquela conexão deve ser aumentada e se os dois neurônios em cada lado de uma conexão são ativados assincronamente, então a conexão deve ser enfraquecida (HEBB, 2002).

A IA pode ser definida pelos seus termos. “Artificial” é tudo que é feito pelo homem. “Inteligência” é uma definição mais abrangente, podendo ser generalizada como a capacidade de extrair informações, aprender com elas e adaptar-se de acordo com a situação, utilizando o pensamento e a razão. A inteligência artificial então é o estudo de como fazer com que as máquinas criadas pelo homem possam pensar como o homem. A IA surgiu ao longo do tempo com base em quatro linhas de raciocínio:

Segundo Haugeland (1985), Sistemas que pensam como seres humanos é o novo e excitante esforço para fazer computadores pensarem, máquinas com mentes, no sentido literal e completo.

Segundo Charniak e McDermott (1985), sistemas que atuam como seres humanos é o estudo das faculdades mentais pelo uso de modelos computacionais.

De acordo com Kurzweil (1990), sistemas que pensam racionalmente é a arte de criar máquinas que realizam funções que requerem inteligência quando realizadas por pessoas.

Sistemas que atuam racionalmente é o ramo da ciência da computação que se preocupa com a automação do comportamento inteligente (KURZWEIL E STUBBLEFIELD, 1993).

Segundo Russel e Norvig (2013) para realizar o estudo da inteligência artificial, as quatro bases de linha de raciocínio são seguidas, envolvendo hipóteses e confirmação experimental centradas em torno da racionalidade. Uma abordagem racionalista envolve uma combinação de matemática e engenharia.

Se a inteligência artificial busca criar uma inteligência que pense como um ser humano, é preciso definir como um ser humano pensa. Para conseguir buscar a fundo os componentes reais da mente humana, é possível utilizar três maneiras (RUSSEL e NORVIG, 2013):

- I. A introspecção, procurando captar novos pensamentos à medida que eles se desenvolvem;
- II. Experimentos psicológicos, observando a ação de uma pessoa;

III. Imagens cerebrais, observando o cérebro em ação.

Para Russel e Norvig (2013) depois de conseguir traçar uma boa teoria a respeito do cérebro humano, é possível desenvolver uma inteligência artificial com base no cérebro humano. Ao desenvolver o programa inteligente, se os comportamentos de entrada e saída e a sincronização entre as funções do programa coincidirem com os comportamentos das teorias do cérebro humano que foram fundamentadas, é a evidência de que alguns dos mecanismos do programa podem estar atuando no cérebro humano. Os desenvolvedores do GPS (do inglês, “General Problem Solver”) ou Resolvedor Geral de Problemas, estavam mais interessados em comparar os passos do raciocínio utilizado pelo programa para resolver os mesmos problemas humanos do que em resolver os problemas de maneira correta.

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

Para Rosa (2011) quando abordamos o Processamento de Linguagem Natural (PLN), podemos defini-lo de formas diferentes, mas levando em consideração a apenas seu principal conceito, a capacidade de comunicação com o modo de fala da espécie humana, pois podemos dizer que os seres humanos se diferenciam de outras espécies principalmente pela sua capacidade de comunicação verbal e escrita, diferente de outras espécies por nós conhecidas, que fazem uso de sinais e gestos, os seres humanos são os únicos que tem a capacidade de se comunicar confiavelmente e com um número restrito de palavras e frases.

Podemos também citar outras propriedades que são exclusivamente de humanos, mas como dito, a principal diferenciação entre seres humanos e outras espécies é a capacidade de comunicação verbal e escrita, isto é o que nos torna sociáveis, logo quando tratamos do assunto PLN, estamos tratando de uma forma de comunicação entre Homem-máquina o mais próximo possível da realidade da comunicação humana, sendo isto que levou um dos maiores nomes da tecnologia, a propor um teste, conhecido como o Teste de Turing (TURING, 1950).

O teste em questão foi publicado no ano de 1950, em seu artigo Maquinaria computacional e inteligência, onde ele buscava resposta para inteligência das máquinas, onde almejava descobrir se uma máquina era capaz de pensar, sendo este conteúdo para a época era considerado um questionamento inconclusivo (TURING, 1950).

Para Turing (1950) era inconclusivo por não haver parâmetros o suficiente para avaliar esse conceito entre máquinas e pensamentos, por este motivo, para ter essa questão respondida foi que Turing a reformulou para desenvolver um teste, onde o intuito era saber se uma máquina poderia ou não ter pensamentos semelhantes à de um ser humano, e se a simulação desses pensamentos era eficaz ou não.

Para apoio ao seu teste e como já dito que esses testes eram algo distante na década de 50, ele propôs um “jogo” descrito por ele mesmo como o “O Jogo Da Imitação”. A frase “O Teste de Turing” às vezes é usada de forma mais geral para se referir a alguns tipos de testes comportamentais para a presença de mentes, ou pensamentos, ou inteligências em qualquer máquina, algo que era impossível, nada faria com que uma máquina se assemelhasse 100% a um humano (TURING, 1950).

Antes mesmo da formulação do teste já havia grandes nomes que abordará este assunto, como por exemplo Descartes (2011), que em seu Discurso do Método dirigia a palavra a questão sobre máquinas e pensamentos.

Para Descartes (2011), se houver máquinas que tenham a aparência e órgãos semelhantes a macacos, ou quaisquer outros animais irracionais, não seremos capazes de reconhecer que são de natureza completamente diferente desses animais, porém, se houver outras máquinas que se pareçam conosco fisicamente e imitem nosso comportamento tão moralmente quanto possível, sempre teremos duas maneiras bastante certas de verificar se eles não são homens de verdade. Entre esses métodos, o primeiro é que eles nunca podem usar palavras ou outros símbolos para organizá-los, da mesma forma que expressamos nossas ideias a outras pessoas.

Podemos muito bem imaginar que uma máquina feita de um modo que possa se expressar com a pronúncia de palavras, e até mesmo algumas dessas palavras sendo associadas a ações corporais que causam certas aflições ou demais sensações, como por exemplo, quando tocado em um determinado ponto, mas não significando que ela possa possuir mais de uma maneira diferente em resposta a qualquer coisa que ocorra, diferente de seres humano que muitas as vezes possui singularidades para reagir a cada ação. (DESCARTES, 2011).

Conforme Descartes (2011), a segunda maneira é que mesmo que eles façam muitas coisas melhor do que qualquer um de nós, eles inevitavelmente falharão em algumas outras coisas. Dessa forma, seria possível descobrir que suas ações não dependem do conhecimento,

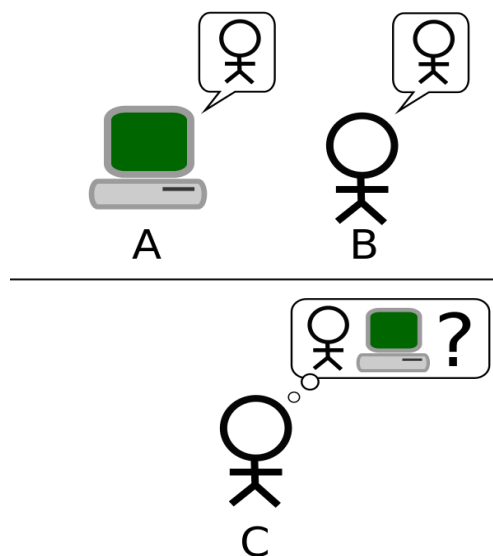
mas de distribuição ordenada de seus órgãos. Pois, embora a racionalidade seja uma ferramenta universal aplicável a todas as ocasiões, tais órgãos precisam de alguma disposição específica para cada ação específica, portanto, acaba sendo impossível que uma máquina tenha diversas formas agir em diferentes ocasiões da vida, ao contrário dos seres humanos.

Turing (1950) descrevia seu “jogo” da seguinte forma:

- I.** Suponhamos que tenha uma Pessoa (B)
- II.** Um interrogador (C);
- III.** uma máquina (A).

Turing (1950) dizia que o interrogador ficava em uma sala separada da pessoa (B) e da máquina (A), sendo o objetivo se caracteriza no interrogador determinar qual é dos dois é máquina (A) e qual é a pessoa (B), conforme exemplificado na figura 1.

Figura 1: Representação do jogo.



Fonte: Alura (2020)

De acordo com Turing (1950), o interrogador conhece a outra máquina e a pessoa pelos rótulos ‘A’ e ‘B’, porém, ao início do jogo, não sabe se com quem ele está falando é outra pessoa ou a máquina, e ao final do diálogo responde se ‘A’ é a pessoa e ‘B’ é a máquina ou ‘A’ é a máquina e ‘B’ é a pessoa. O interrogador tem a opção de fazer perguntas à pessoa e à máquina, como por exemplo "B, por favor, pode me dizer o comprimento de seu cabelo?"

O objetivo da máquina é tentar fazer com que o interrogador conclua de forma errada que a máquina é uma outra pessoa; o objetivo da outra pessoa é tentar ajudar o interrogador a identificar corretamente a máquina (TURING, 1950).

Para Turing (1950) este é um dos principais testes de inteligência de máquina, e ainda mais importante, é mostrado que desde antes mesmo da tecnologia alcançar os patamares atuais havia grandes interesses em saber até que nível uma máquina poderia chegar se tratando da igualdade da linguagem natural humana usado no dia a dia por uma máquina.

Segundo Obermeier (1987) o processamento de linguagem natural comumente é dividido em seis grandes áreas:

- I.** Interfaces em língua natural para a base de dados;
- II.** Tradução de máquina (De uma língua natural para outra);
- III.** Programas de indexação inteligente para sumarização de grandes quantidades de texto;
- IV.** Geração de texto para produção automática de documentos padrões;
- V.** Sistemas de fala para permitir interação de voz com computadores e
- VI.** Ferramentas para desenvolver sistemas de processamento de Línguas Naturais para aplicações específicas.

Como evidenciado por Rosa (2011) desde o princípio dos cinemas podemos ver a exploração fictícia destas áreas, sempre com uma abordagem futurística, sendo inúmeras vezes retratada por meio de assistentes, demonstrando grandes capacidades muito importantes, entre elas podemos citar:

- I.** Entendimento da língua natural (Até mesmo a leitura labial dos passageiros);
- II.** Geração de língua natural;
- III.** Síntese de fala;
- IV.** Recuperação de informações;
- V.** Reconhecimento de fala;
- VI.** Extração de informações;

VII. Inferência.

De acordo com Rosa (2011) praticamente todos os conceitos de extrema importância para o processamento de linguagem natural abordados na ficção, algo que atualmente vemos cada vez mais se expandindo e evoluindo, tornando-se algo presente no dia a dia de muitas pessoas e empresas.

Segundo Rosa (2011), indo ao ponto da linguística, temos alguns conceitos básicos de extrema proeminência, onde podemos considerar palavra como sendo a expressão básica da linguística, das quais são consideradas como símbolos que auxiliam a se expressar sobre indivíduos, propriedades de indivíduos, relações entre indivíduos e propriedades de relação entre indivíduos. Onde as palavras são grafadas como uma cadeia de caracteres do alfabeto nas quais organizadas em uma estrutura interna de subcadeias conectada são classificadas como morfemas.

Logo o morfema principal de uma palavra é atribuído como raiz, onde uma raiz pode-se aficar a outros morfemas, a exemplo, prefixos e sufixos, cada morfema tem uma atribuição estabelecida na composição de uma palavra, este processo de composição é denominado de análise morfológica, além disto também há outros pontos comumente usados na PLN, mas não necessariamente a implementação de todos eles (ROSA, 2011):

- I.** Análise Léxica: dentro da PLN pode ser utilizada para a tokenização, que é uma prática para desunir um texto inteiro em palavras, conhecidas e chamadas de tokens. Assim que separados em tokens, com o uso do processamento de linguagem natural, se necessário é possível realizar a análise morfológica.
- II.** Análise Morfológica: é responsável pela análise de palavras isoladas, onde cada palavra é classificada em uma categoria de acordo com as regras que regem a língua portuguesa.
- III.** Análise Sintática: Trata a estrutura das palavras em uma frase. Os resultados obtidos na etapa da análise morfológica servem de entrada para a análise sintática. análise sintática é o processo de analisar uma cadeia de palavras para descobrir a sua estrutura frasal, de acordo com as regras de uma gramática.
- IV.** Análise Semântica: em síntese a análise semântica é responsável por tratar as estruturas criadas na análise sintática. Analisa o significado das palavras.

Para Rosa (2011) dentro do processamento de linguagem natural também é trabalhado com classes de palavras, substantivos, adjetivos, numerais, pronomes, verbos, advérbios e conectivos são as classes da língua portuguesa brasileira.

Segundo Rosa (2011), a “língua” é uma forma de comunicação, a mesma é retratada em camadas, onde ela estrutura em sistema com regras complexas em níveis, considerando o nível mais baixo os sons e ritmos, até os mais altos. Onde cada nível é lido com um aspecto do processo de comunicação e forma um subsistema inteiro como seus próprios elementos e regras de combinação.

Entre outros tópicos relacionado a linguística e a PLN há longos caminhos a se percorrer, passando entre gramática, sintaxe e semântica, a própria interação entre os níveis da língua, os problemas relacionados a ambiguidade que podem ocorrer quando as expressões básicas ou palavras podem se referir a coisas diferentes existente sendo essa a ambiguidade lexical, é a ambiguidade semântica, quando se tem mais de um significado para a frase, e até os grafos de derivação utilizado algumas vezes para derivar a completamente a sentença e construindo no grafo (ROSA, 2011).

Para Russel e Norvig (2013) estes são conceitos fundamentais para a continuação de conteúdo com o processamento de linguagem, pois um dos recursos de maior importância se tratando de PLN é classificação textual, também é conhecida como categorização. O reconhecimento de linguagem e a classificação de gênero são exemplos de classificação de texto, assim como a análise de sentimento (classificação de filmes ou análises de produtos como positivos ou negativos) e descoberta de spam (classificação de e-mails como spam ou não spam).

Ainda para Russel e Norvig (2013) outro tópico fundamental é a extração de informações, que é o processo de obtenção de conhecimento olhando para o texto e encontrando pontos chaves do texto a ser utilizado. Uma tarefa típica é extrair exemplos de endereços de páginas da web com campos de banco de dados para ruas, cidades, estados e códigos postais; ou condições de tempestade de informações meteorológicas, incluindo campos de temperatura, velocidade do vento e precipitação.

Em uma área limitada, isso pode ser feito com alta precisão. À medida que o campo se torna cada vez mais comum, são necessários modelos de linguagem e técnicas de aprendizagem mais complexos Russel e Norvig (2013).

Segundo Russel e Norvig (2013) com este conteúdo temos o surgimento de uma das primeiras aplicações da PLN, a tradução automática de texto, historicamente, a tradução automática tem três aplicações principais. A tradução aproximada fornecida pelos serviços online gratuito fornece a "essência" da frase ou documento estrangeiro, mas contém erros. As traduções pré-editadas utilizadas para publicar seus documentos e materiais de vendas em vários idiomas. O texto-fonte original é escrito em um idioma limitado e é facilmente traduzido automaticamente, e os resultados geralmente são editados por humanos para corrigir quaisquer erros. As traduções com restrição de fonte são totalmente automatizadas, mas apenas em idiomas altamente uniformizada, como boletins meteorológicos.

E cada vez mais utilizado, porém com muito a ser desenvolvido, temos integrado junto ao processamento de linguagem natural o reconhecimento de voz, o reconhecimento de fala é a tarefa de reconhecer uma série de palavras faladas por uma pessoa. A técnica tornou-se um dos ramos da inteligência artificial mais importantes (RUSSEL e NORVIG 2013).

Para Russel e Norvig (2013) o reconhecimento de fala algo é difícil de se trabalhar porque a voz do locutor é vaga e muitas das vezes possui muitos ruídos. Como um exemplo conhecido, quando falado rapidamente, a frase "Iluminação" soa quase o mesmo que "Eliminação" entre demais outros exemplos de ambiguidades.

O reconhecimento de voz depende da qualidade de todos os seus componentes, modelos de linguagem, modelos de pronúncia de palavras, modelos de telefone e algoritmos de processamento de sinal usados para extrair características espectrais de sinais acústicos (RUSSEL E NORVIG, 2013).

De acordo com Russel e Norvig (2013) a precisão de um sistema de reconhecimento de voz depende de muitos fatores, como a qualidade do som, o vocabulário que é de extrema importância, quanto mais vasto, maiores as possibilidades de reconhecimentos de mais palavras.

Segundo Adams (2018) mesmo que todas essas tecnologias sejam utilizadas, o processamento de linguagem natural ainda precisa se desenvolver dentro de seu próprio conceito. Em outras palavras, devemos continuar buscando melhorias na compreensão e no uso da linguagem natural. Isso porque, seja para solucionar as necessidades das pessoas que utilizam a máquina a qualquer hora do dia ou da noite, a comunicação efetiva entre a máquina e o usuário está se tornando cada vez mais importante.

"Uma das coisas que sempre me vejo tendo que explicar às pessoas é que a tecnologia da fala ainda não é um problema resolvido. Não é mais algo estranho, é mais natural e normal. Podemos afirmar que as pessoas se sentem mais confortáveis com a tecnologia, e a tecnologia está aperfeiçoando com o tempo" (ADAMS,2018).

2.3 REDES NEURAIS ARTIFICIAIS

Segundo Kovács (2006) redes neurais são uma forma da máquina imitar os neurônios biológicos humanos. Um programa desenvolvido para analisar e resolver problemas de uma forma semelhante aos neurônios biológicos. Tendo sua origem remoldada dos modelos matemáticos e de engenharia de neurônios biológicos. O neurônio biológico ou célula nervosa foi descoberta anatomicamente e descrita com alto nível de detalhes. O neurônio biológico é delimitado por uma fina membrana muscular, que possui diversas funções biológicas, mas também existe uma função de grande importância responsável pelo funcionamento elétrico da célula nervosa. Iniciando no corpo celular ou na soma, a parte central dos processos metabólicos da célula nervosa, iram projetar extensões filamentosas chamados de axônio e dendritos.

De acordo com Kovács (2006) redes neurais trouxe consigo diversos benefícios. A tomada de decisão de uma rede neural primeiramente será de sua estruturação. Logo após sua habilidade de aprender, o grande diferencial de algumas tecnologias, o aprendizado, com isso podendo generalizar. Com isso poderá tomar decisões de saída, mesmo que não tenha sido treinada, a rede neural irá generalizar, para produzir a saída mais adequada possível. Com isso poderá atuar em problemas de grande complexibilidade. Porém a rede neural não trabalha de forma individual, sendo integrada a engenharia de sistemas. Diluindo um problema em um número de tarefas e inserindo um subconjunto de tarefas a rede neural. A rede neural é composta de diversas formas, como:

- I. **Adaptabilidade:** a rede neural tem uma forte habilidade de adaptação de seus pesos sinápticos as transformações do meio ambiente. Assim, mesmo que a rede neural tenha sido desenvolvida, treinada ou adaptada para um específico ambiente, irá facilmente ser adaptada as modificações, transformações do ambiente. Quando estiver atuando em um ambiente não estacionário, onde terá alterações com o tempo, a rede neural poderá ser desenvolvida para mudar seus pesos sinápticos em "real-time". Quanto maior a adaptabilidade do sistema, maior sua estabilidade, mantendo-se estável quando o ambiente tiver passado por alguma alteração. Porém, sua adaptabilidade deve ser altamente calibrada, pois perturbações insignificantes devem ser ignoradas, pois podem

afetar o desempenho do sistema, porém é necessário a adaptação para pequenas alterações que sejam significantes. (GROSSBERG,1998).

II. Tolerância a falhas: rede neural aplicada de forma física (hardware), tem grande vantagem em ser inerentemente tolerante a falhas. Sendo degradada tranquilamente quando exposta a operações adversas. Se houver um problema ou defeito com algum de seus neurônios, será afetado em qualidade o padrão de armazenamento. Porém com a distribuição da informação armazenada na rede neural, o dado teria que ser massivo para que a rede seja afetada seriamente. Com isso aumenta em grande porcentagem seu nível de segurança, pois só terá uma perda considerável quando tiver uma falha global. (KERLIRZIN E VALLET, 1993).

III. Uniformidade de análise e projetos: segundo Kerlirzin e Vallet (1993) as redes neurais compartilham uma universalidade de como processamento de informações. Pois uma notação terá uso por todo domínio que a rede neural está envolvida. Podendo ser mostrada em diversos modos, assim como:

- a. Neurônios, que iram representar uma forma comum em todas as redes neurais.
- b. A redes neurais podem ser criadas através da introdução da forma homogênea de módulos.
- c. A semelhança irá possibilitar o compartilhamento de conceitos e aprendizados em diferentes formas de aplicações de redes neurais (KERLIRZIN E VALLET, 1993).

IV. Informação contextual: para Kerlirzin e Vallet (1993) todo conhecimento adquirido será representado pela estrutura e estado de ativação da rede neural. A atividade de cada neurônio presente nesta rede, irá afetar ao conjunto completo de neurônios da rede. Sendo assim toda informação será tratada de forma natural pela rede neural.

2.3.1 REDES NEURAIAS COM MULTICAMADAS

A partir da segunda camada de uma rede neural alimentada, será estruturada por uma ou mais camadas ocultas, onde todo este conjunto de nós, foram nomeados de neurônios ocultos. Onde estes neurônios ocultos têm como principal função, atuar na entrada externa e saída da rede neural de uma boa forma. Quando acrescentado uma ou mais camada oculta a uma rede neural, a possibilita a extração de estatísticas de ordem elevada. Mesmo com sua conectividade local, toda rede irá ter perspectiva global, pois ela é estruturada com uma

dimensão extra de interações neurais e conexões sinápticas. Quando o tamanho de entrada for grande, a skill dos neurônios ocultos de realizar a extração de forma elevada se torna muito valiosa. (CHURCHLAND e SEJNOWSKI,1992).

Segundo Churchland e Sejnowski (1992) os nós de fonte da camada de entrada fornecerão elementos do padrão de ativação, ou seja, vetores de entrada, que iram fornecer sinais de entradas voltados aos neurônios. Logo após termos os sinais de saída da segunda camada, que serão os sinais de entrada para terceira camada, os sinais de saída da terceira camada serão os sinais de entrada da quarta camada e assim continuará por toda rede neural. Os sinais que iram sair da última camada (camada final) será a representação do trabalho global de toda a rede, que atuaram na entrada da primeira camada (camada de entrada).

2.3.2 REPRESENTAÇÃO DE CONHECIMENTO

Segundo Kovács (2006) existem duas formas de representação de conhecimento, a primeira será a forma como a informação será exposta, como ela será explícita; a segunda forma será como a informação foi codificada fisicamente para ser usada subsequentemente. Toda aplicação terá o direcionamento voltada para um objetivo.

Aplicações na área do mundo reais das máquinas “inteligentes”, poderá ser considerada uma solução de boa qualidade, quando tiver uma boa representação de conhecimento. Existe uma alta diversidade de forma de representação de conhecimento por toda a rede neural, que pode se iniciar no parâmetro de entrada e se expandir por toda rede interna, que possibilitará uma solução de maior qualidade (KOVÁCS, 2006).

Segundo Haykin (2000) redes neurais tem uma atribuição de grande importância, onde necessita aprender o modelo do mundo onde ela está em atuação e manter o modelo estável para o mundo real, com o objetivo de alcançar todos os propósitos da aplicação. Sendo o conhecimento do mundo dividido em duas formas, que são:

1. Primeiro o estado de já conhecido do mundo, que terá representação de tudo aquilo que já era familiar, ou seja, as informações que já eram conhecidas. Este modelo foi nomeado de informação previa (HAYKIN, 2000).
2. O segundo será as observações (medidas) do mundo, que será constituída por detectar por meio de sensores e entender as informações que estão no mundo da aplicação da rede neural. A informações obtidas através desta técnica, tendem a ser ruidosas, sendo assim estão mais expostas a erros devido a ruídos nos

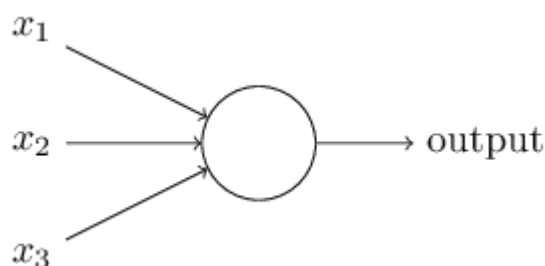
sensores e não conformidades sistêmicas. Mesmo com essa dificuldade, desta maneira que é retirada as informações do ambiente necessárias, voltadas para adaptar e treinar a rede neural (HAYKIN, 2000).

Segundo Rosenblatt (1957) o perceptron foi criado no fim da década de 1950, na universidade de cornell inspirado em trabalhos anteriores, sendo uma autêntica rede de múltiplos neurônios do tipo discriminadores lineares, nomeados de perceptron. Sendo um perceptron uma rede criada com diversas camadas, tem os neurônios de entrada, que iram receber as informações, e iram enviar para segunda camada de neurônios, iram receber e já disponibilizar com a informação.

E a segunda camada irá atuar com a lógica implantada nela. E assim sucessivamente até que chegar camada de saída (camada final). Ressaltando que as camadas que ficam no meio, que não são camadas nem de entrada nem de saída, foram nomeadas de camadas ocultas (ROSENBLATT,1957):

Um perceptron é uma forma matemática que será atribuída a ela diversas entradas, x_1 , x_2 , x_3 , onde irá conduzir a uma única saída, conforme figura 2 (KOVÁCS, 2006):

Figura 2: Representação do perceptron.



Fonte: Dagli (2020)

Rosenblatt (1957) teve a ideia de criar uma regra para realizar o cálculo da saída. Entrando com pesos, w_1, w_2, \dots , números reais, demonstrando a grande importância das entradas para gerar a saída. Com a saída do neurônio 0 ou 1.

Sendo imposta por uma soma ponderada, $\sum w_j x_j$, maior ou menor que um valor limiar (threshold). Threshold também faz parte dos números reais, sendo também um parâmetro do neurônio. A Figura 3 evidencia a expressão algébrica representando sua funcionalidade (ROSENBLATT,1957):

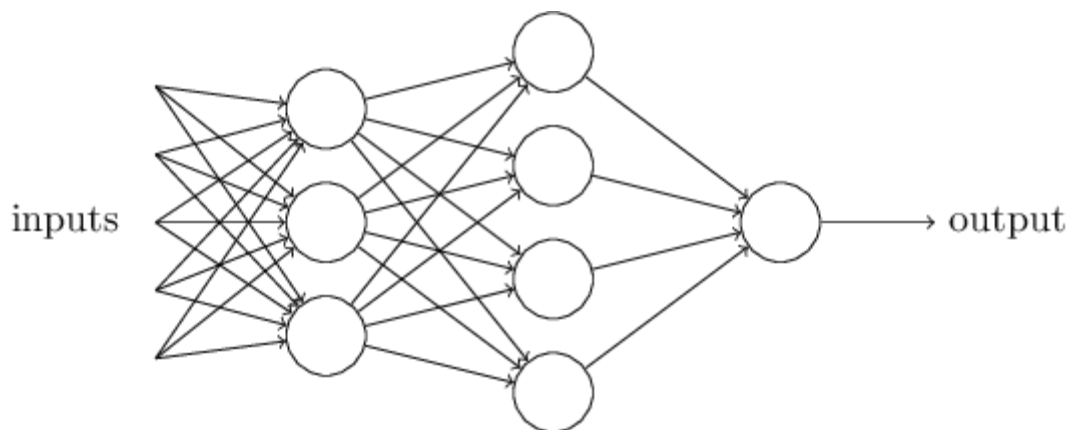
Figura 3: Termos algébricos threshold

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

Fonte: Ahmad (2018).

Perceptron não toma decisões perfeitas como o ser humano, mas perceptron tem a capacidade de pesar diversas formas de evidências para decidir qual decisão tomar. Sendo assim uma rede bem estruturada de perceptrons pode decidir por tomar decisões bem complexas e bem-feitas, assim como demonstrado na figura 4 (ROSENBLATT,1957):

Figura 4: Diferentes pesos de evidências do perceptron.



Fonte: Ahmad (2018).

Segundo Rosa (2011) temos uma diversidade de tarefas que podem ser efetuadas com uma rede neural. Podendo ser utilizada para reconhecimento de letra, utilizada também para leitura, entendimento de sentença e diversas outras tarefas. Sendo também de grande valor no armazenamento de dados e em recuperação de dados salvos em memória. Longe de ser uma aplicação perfeita, isenta de erros, mas representa muito bem a inteligência artificial, imitando com grande semelhança a tomada de decisão humana.

2.3.3 REDES NEURAIIS CONVOLUCIONAIS

Uma rede neural convolucional é uma variação das redes de Perceptrons de Múltiplas Camadas, onde possui foco na visão computacional para processamento de dados visuais. Ela é capaz de aplicar filtros em dados visuais para estabelecer relações de semelhança entre os pixels da imagem ao longo do processamento da rede neural, realizando a transformações dos dados na camada de entrada antes de passar pra o processamento em uma rede neural tradicional. Esse tipo de rede é muito utilizado na detecção e reconhecimento de imagens e vídeos.

A rede neural convolucional consiste em múltiplas partes com funções diferentes em cada uma, onde inicialmente é comum aplicar sobre os dados de entrada da rede camadas de convolução. Uma camada de convolução é composta por diversos neurônios, cada um fica responsável por aplicar um filtro em uma parte específica de uma imagem. A combinação das entradas de um neurônio, utilizando os pesos de cada uma das conexões produz uma saída para a camada seguinte. Os pesos atribuídos a cada conexão representam uma matriz com o filtro de convolução de imagens, também conhecido como *kernel* ou máscara.

O tamanho do filtro, especificado na matriz de convolução, determina o tamanho da vizinhança que cada neurônio da camada irá processar.

Após as camadas de convolução é aplicado o *pooling*. A função dessa camada é reduzir a dimensionalidade dos dados na rede, agrupando um conjunto de dados semelhante para obter as características mais importantes que correspondam aquela imagem.

O resultado da repetição dessas camadas com a aplicação do *pooling* geram um conjunto de *features*, onde será aplicada o Flattening para transformar as matrizes geradas em cada etapa da aplicação do *pooling* em vetores para conseguir trabalhar com as redes neurais convencionais.

2.4 APRENDIZADO DE MÁQUINA (MACHINE LEARNING)

Aprendizado de Máquina (*Machine Learning* em inglês), segundo Mitchell (1997) é a construção de programas que melhorem seu desempenho por meio de exemplos.

Segundo Coppin (2010), o Aprendizado de Máquina está diretamente ligado com a Inteligência, pois a partir do momento que o sistema é capaz de aprender e executar uma tarefa, ele pode ser chamado de inteligente. Para que seja possível o aprendizado de máquina, é necessária uma grande quantidade de dados para gerar o conhecimento para a máquina, que são hipóteses geradas a partir dos dados obtidos.

De acordo com Monard (2003), o Aprendizado de Máquina é uma área da Inteligência Artificial que possui o objetivo de desenvolver técnicas computacionais capazes de adquirir conhecimento de forma automática. Através da obtenção de experiências, é possível desenvolver um sistema de aprendizado que é capaz de tomar decisões baseadas nestas experiências anteriores que tiveram soluções bem-sucedidas.

Os algoritmos de Aprendizado de Máquina geram hipóteses a partir de uma quantidade de dados para que seja possível avaliar os conceitos induzidos por esses algoritmos em determinados problemas (LUDERMIR, 2021).

A inferência indutiva é um dos principais métodos utilizados para adquirir novos conhecimentos e realizar a tomada de decisão com base em eventos futuros em Aprendizado de Máquina. Ela permite obter conclusões genéricas sobre um conjunto particular de dados de exemplos. É caracterizada como um raciocínio que parte de um conhecimento específico e o generaliza, partindo da parte para um todo. Através da indução, efetuando a inferência indutiva sobre os exemplos gera-se as hipóteses que podem ou não preservar a verdade. As chances de as generalizações realizadas pela inferência indutiva serem corretas varia de acordo com a qualidade dos dados apresentados nos exemplos (MONARD, 2003).

A indução é um dos recursos mais utilizados pelo cérebro humano para derivar novos conhecimentos. Através da indução, Arquimedes descobriu a primeira lei da hidrostática e o princípio da alavanca, Kepler descobriu as leis do movimento planetário, Darwin descobriu as leis da seleção natural das espécies. Apesar de muito utilizada, ela deve ser usada com um cuidado, pois caso o número de dados de exemplo para avaliar for insuficiente, as hipóteses que serão obtidas poderão ter pouco valor (MONARD, 2003).

Segundo Mitchell (1997), durante muitos anos os algoritmos foram desenvolvidos utilizando diferentes paradigmas de aprendizado, tais como estatístico, conexionista, *instance-based*, genético e sistemas de aprendizado simbólico. Os sistemas de aprendizado simbólico comumente são utilizados em situações em que os aprendizados adquiridos precisam ser interpretados por humanos. O conhecimento induzido por algoritmos de aprendizagem

simbólico é geralmente representado por árvores de decisão ou por um conjunto de regras de produção.

Segundo Coppin (2010), existem diversos métodos de aprendizagem de máquina. Entre os métodos existem, o aprendizado por hábito, que possui a característica do sistema inteligente aprender de acordo com experiências anteriores, apenas armazenando os dados que podem ser classificados e generalizados. Há também o método de aprendizagem por conceito, que consiste em analisar todas as hipóteses e demonstrar qual é a hipótese correta para determinada situação, realizando uma subdivisão, partindo da hipótese mais genérica para a hipótese mais específica e próxima do correto.

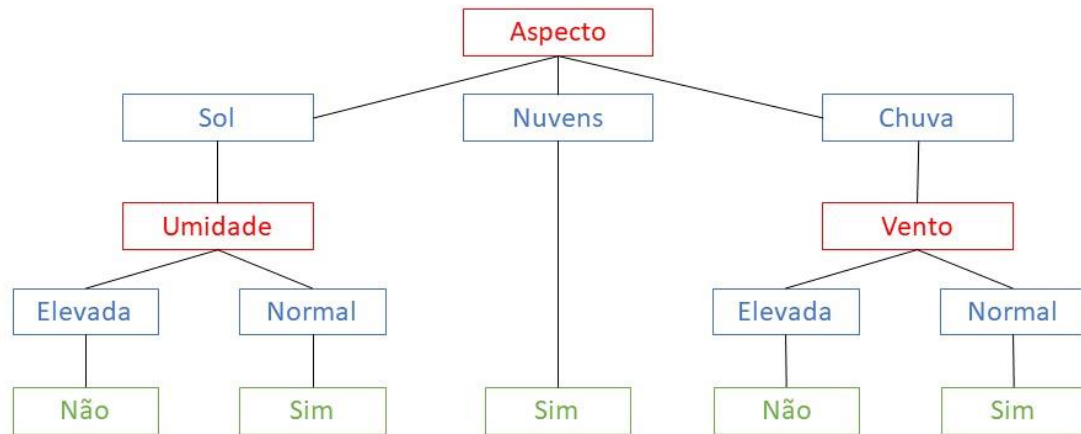
A preocupação inicial dos algoritmos de Aprendizagem de Máquina era obter a melhor precisão para o classificador de acordo com a disponibilidade de uma quantidade limitada de exemplos. Cada um dos algoritmos possui seu próprio formato para os dados de entrada bem como para representar o conhecimento induzido, tais como Árvores de Decisão ou Regras de Produção (PRATI et al., 2002).

Segundo Prati et al. (2002), uma Árvore de Decisão é uma estrutura de dados recursivamente definida como:

- I.** Um nó folha, que indica uma classe, ou
- II.** Um nó de decisão, que contém um teste de um valor de um atributo da árvore. Para cada um dos valores possíveis de um atributo, pode-se existir uma outra árvore de decisão, denominada subárvore.

Para Prati et al. (2002), é necessário representar graficamente uma árvore de decisão segundo a figura 5 abaixo referente a árvore de decisão para jogar Tênis. Os retângulos em vermelho representam os nós de decisão, os retângulos em azul identificam os possíveis resultados dos nós de decisão e os retângulos em verde identificam os nós folhas.

Figura 5: Árvore de Decisão para jogar Tênis



Fonte: Gomes (2019)

Uma Regra de Produção é geralmente representada na forma:

if <condição> then <classe = Ci>

Onde a <condição> é uma disjunção de conjunções de testes para os atributos e Ci representam um dos possíveis valores para a classe. A regra de produção que representa a mesma hipótese da árvore de decisão representada na figura 6 pode ser definido como o conjunto abaixo (PRATI et al., 2002):

Figura 6: hipótese da árvore de decisão

```

IF Aspecto = Nuvens THEN CLASS = Sim
IF Aspecto = Sol AND umidade = Elevada THEN CLASS = Não
IF Aspecto = Sol AND umidade = Normal THEN CLASS = Sim
IF Aspecto = Chuva AND Vento = Fraco THEN CLASS = Não
IF Aspecto = Chuva AND Vento = Forte THEN CLASS = Sim
  
```

Fonte: Gomes (2019)

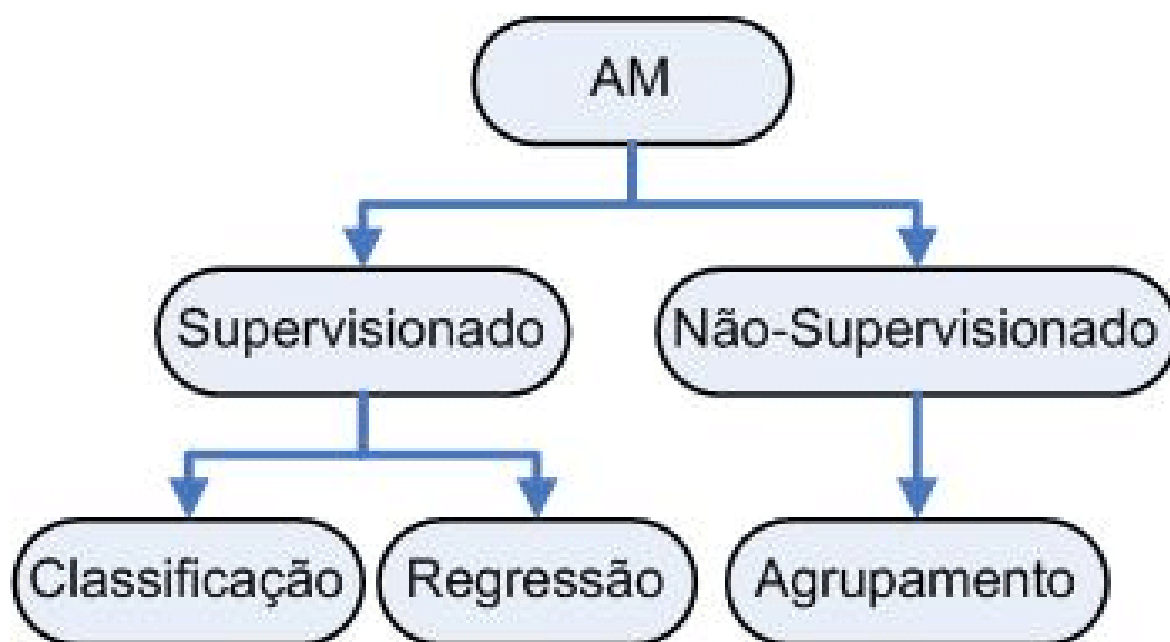
É importante notar que, geralmente, a variedade de informações apresentadas pelo algoritmo é decorrente da avaliação gerada no próprio conjunto de exemplos fornecidos ao algoritmo. Com isso, a preocupação maior está no comportamento futuro do algoritmo,

utilizando novos exemplos que ainda não foi visto pelo indutor, ou algoritmo de aprendizagem, durante a indução do classificador. Para obter então um melhor resultado, é possível utilizar uma avaliação com amostras diferentes das que foram utilizadas pelo algoritmo para realizar o treinamento inicial (PRATI et al., 2002).

De acordo com Araújo (2015), na Aprendizagem de Máquina, o aprendizado indutivo é realizado a partir de processos de exemplos externos do sistema de aprendizado. Ele pode ser dividido em dois tipos, supervisionado e não-supervisionado.

Segundo Araújo (2015), o aprendizado supervisionado, é fornecido ao algoritmo de aprendizado, um conjunto de dados de treinamento para os quais o rótulo da classe associada é conhecido. Para cada exemplo apresentado, é necessário apresentar a resposta desejada. Por exemplo, no caso de classificação de imagens, deve-se informar como distinguir imagens, como de gatos e cachorros, para que o algoritmo possa obter um rótulo para identificar a qual classe o exemplo pertence. Cada exemplo é descrito por um vetor de valores de características ou atributos de determinado problema e o rótulo da classe associada a ele. O objetivo do algoritmo de indução é construir um classificador que consiga determinar corretamente a classe de novos exemplos que serão apresentados e ainda não estão rotulados. Para rótulos de classe discretos, esse problema é conhecido como classificação e para valores contínuos como regressão.

Figura 7: Hierarquia de Aprendizado de Máquina



Fonte: Carvalho (2002)

Segundo a figura acima, é possível identificar a hierarquia de aprendizado de máquina descrita anteriormente, identificando os tipos de aprendizagem classificando com supervisionado e não-supervisionado (ARAÚJO, 2015).

No aprendizado não-supervisionado, os exemplos são fornecidos ao algoritmo sem rótulos. O algoritmo de indução, ou indutor, analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira, formando agrupamentos ou clusters. De acordo com as similaridades dos atributos dos exemplos, o algoritmo tenta agrupá-los e executar uma análise para determinar o que cada agrupamento significa no contexto do problema que está sendo proposto a análise (CHEESEMAN & STUTZ 1990).

No Aprendizado de Máquina, existem diversos paradigmas a seu respeito, como: Simbólico, Baseado em Exemplos, Estatístico, conexionista e Genético, que podem ser descritos simplesmente como (MONARD, 2003):

- I. **Simbólico:** Um sistema de aprendizagem simbólico consiste em construir representações simbólicas de um conceito através da análise de seus exemplos e contraexemplos apresentados. Essas representações são geralmente representadas em forma de expressão lógica, árvore de decisão, regras de produção e rede semântica.
- II. **Baseado em Exemplos:** Ou baseados em instâncias (*instance-based em inglês*), são sistemas baseados em exemplos consistem em utilizar um exemplo similar cuja classe já é conhecida para identificar quando um novo exemplo pertencer a mesma classe, classificando exemplos nunca visto com base em exemplos conhecidos. Um Raciocínio Baseado em Casos é um exemplo clássico de uma representação de um paradigma de Aprendizagem de Máquina baseado em exemplos, onde são analisados os casos que já ocorreram para realizar a classificação de novos casos.
- III. **Estatístico:** A ideia geral consiste em utilizar modelos estatísticos para encontrar uma boa aproximação do conceito que foi induzido pelo algoritmo. Dentre os métodos estatísticos, os principais são os de aprendizado Bayesiano, que utilizam um modelo probabilístico baseado no conhecimento prévio do problema, o qual é combinado com exemplos de treinamento para determinar a probabilidade final de uma hipótese.
- IV. **Conexionista:** Sistemas com algoritmos conexionistas são baseados em Redes Neurais, que são estruturas distribuídas formadas por um grande número de unidades de processamento conectadas entre si inspiradas no modelo biológico do sistema nervoso,

e por conta da conexão dessas unidades é definido o conexionismo. As Redes Neurais possuem grande potencial na resolução de problemas, porém requer um maior processamento e desenvolvimento de um algoritmo eficaz.

- V. **Genético:** Uma Aprendizagem de Máquina com esse paradigma é derivado no modelo de evolução natural, possuindo uma analogia direta com a teoria da evolução de Darwin, onde sobrevivem os mais adaptados ao ambiente. Consiste em uma classificação de elementos de classificação que competem para fazer a predição. Os que possuírem uma baixa performance serão descartados enquanto os que possuírem uma performance mais alta permanecem e produzem variações de si mesmo.

De acordo com Ludermir (2021), a utilização do Aprendizado de Máquina para solucionar os problemas necessita de alguns pré-requisitos. Sempre é necessário manter um bom conjunto de exemplos, utilizando técnicas para melhorar a qualidade dos dados e, conseqüentemente, melhorar a eficácia do algoritmo. Também é necessário identificar qual o melhor algoritmo para solucionar determinado problema, e logo após, definir quais serão os parâmetros do algoritmo escolhido. O sistema de aprendizado deve estar constantemente atualizado, de acordo com as atualizações recorrentes no conjunto de dados.

2.5 APRENDIZADO PROFUNDO (DEEP LEARNING)

O deep learning, ou aprendizado profundo é uma subárea do conceituado campo de Aprendizado de Máquina, que se baseia em abordagens computacionais inspiradas na estrutura e funcionamento do cérebro para a construção e treinamento de redes neurais artificiais, é uma forma poderosa de aprendizado de máquina para resoluções de problemas que envolvam percepções de imagem ou reconhecimento de voz (NIELSEN, 2019).

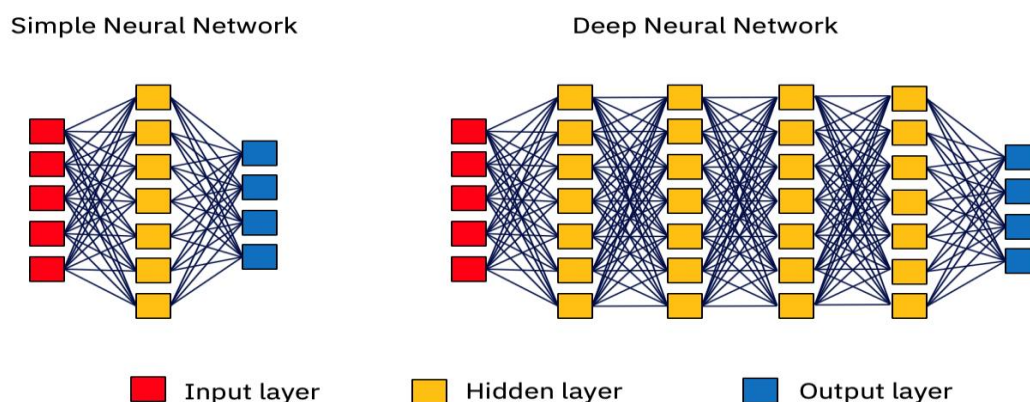
“As ondas de Deep Learning atingiram as costas da linguística computacional por vários anos, mas 2015 parece ser o ano em que toda a força do tsunami atingiu as principais conferências de Processamento de Linguagem Natural (PNL).”
(MANNING, 2015)

Segundo Goodfellow et al. (2016) o aprendizado profundo utiliza um processamento múltiplo em camadas para descobrir padrões e estruturar em conjuntos de dados. Cada camada

aprende um conceito de acordo com sua camada subsequente construída, quanto maior o nível da camada, mais abstratos são as definições aprendidas.

Seu conceito baseia-se no estudo de redes neurais, que busca copiar o comportamento do cérebro humano para processar as informações que o sistema recebe, um exemplo da utilização é utilizar uma rede neural para interpretar formas geométricas, onde seriam reconhecidas as bordas na primeira camada e, em seguida, adiciona as definições das formas mais complexas nas camadas subsequentes. Um algoritmo de aprendizado profundo é um algoritmo que faz uso de uma rede neural para resolver um determinado problema. Uma rede neural é um algoritmo de inteligência artificial que utilizar uma entrada, fazendo com que essa entrada atravessasse uma rede de neurônios chamada de camadas e, em seguida, apresenta a saída conforme a figura 7 (CERON, 2020):

Figura 8: Rede Neural Simples e Rede Neural Profunda (Deep Learning)



Fonte: Ceron (2020)

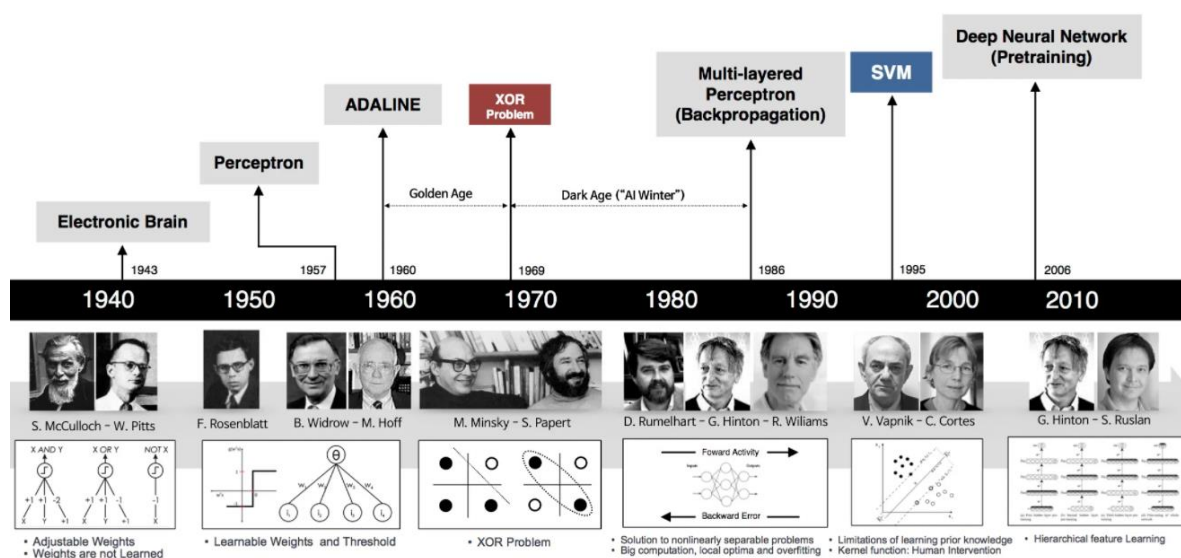
Segundo Ceron (2020) Quanto mais camadas de seus neurônios, mais profunda é a rede. Caso a saída estiver correta, isto é ótimo. Já Se a saída estiver incorreta, espera-se que o algoritmo veja e "ajuste" a conexão do neurônio para que na próxima vez que seja fornecida uma entrada, ele retorne a resposta correta.

Através da combinação de um conjunto de dados, processadores rápidos e algoritmos sofisticados, as máquinas podem passar a realizar tarefas que antes só poderiam ser executadas por humanos, como por exemplo tomadas de decisões, reconhecimento de voz e imagens, utilizando camadas de neurônios matemáticos para processar os dados, de forma que a informação passa por cada camada, com a saída da camada anterior fornecendo a entrada para camada seguinte. A primeira camada de uma rede é chamada de camada de entrada, a última é chamada de camada de saída, enquanto as camadas que estão entre as duas são chamadas de

camadas ocultas. Cada camada é composta por um algoritmo simples e uniforme que contém um tipo de função de ativação. (MADAN e MADHAVAN, 2020)

Um dos aspectos do Deep Learning é a extração de recursos, que utiliza um algoritmo para gerar de forma automática “recursos” significativos dos dados para fins de aprendizado, treinamento e compreensão. Para chegarmos ao Deep Learning, passamos por grandes marcos nas evoluções das redes neurais conforme evidenciado na figura 8 (FUSINSKA, 2017):

Figura 9: Marcos no desenvolvimento das redes neurais.



Fonte: Fusinska (2017)

Como exibido na figura 7, segundo Chagas (2019) ao final da década de 1970 aconteceu o que se chamou de “o inverno da IA”, esse fenômeno afetou tanto o deep learning quanto a inteligência artificial de modo geral, foi a década em que ocorreu um grande impacto nos investimentos da área, sendo assim limitando o avanço da inteligência artificial e das redes neurais profundas.

Segundo Madan e Madhavan (2020) Nos últimos anos houve um passo evolutivo bem significativo, quando os computadores passaram a se tornar mais rápidos no processamento de dados, e GPUs foram desenvolvidas o que significou um salto no tempo de processamento trazendo um aumento de velocidade computacional significativo.

Mas conforme dito por Manning (2015), e vindo para o recente período, o ano de 2015 foi de extrema importância quando tratamos a integração de deep learning e processamento de linguagem natural, sendo o ano em questão marcado por diversos eventos ao redor de planeta,

como por exemplo o Deep Learning Workshop, Facebook AI Research Lab entre outros grandes eventos.

Segundo Manning (2015) o aprendizado profundo deu início a inúmeros avanços tecnológicos incríveis nos recentemente, existem diversos casos sucessos, como por exemplo o sistema Google Voice voicemail transcriptions, onde muitas vezes não eram totalmente inteligíveis, onde que com o uso da uma grande rede neural profunda os erros de transcrições foram diminuídos em 49%, e permanecem em evolução.

“O próximo grande passo para o Deep Learning é o entendimento natural da linguagem, que visa dar às máquinas o poder de entender não apenas individualmente palavras, mas frases e parágrafos inteiros” (LECUN, 2014)

As técnicas de deep learning continuam em pleno avanço tecnológico, especialmente quando várias camadas são usadas. No entanto, o uso das redes neurais profundas ainda apresenta limitações. É inegável que a inteligência artificial está cada vez mais próxima da realidade, mas ainda há um longo caminho a percorrer. Aceitar o aprendizado profundo em diferentes campos do conhecimento permite que toda a sociedade se beneficie dessa tecnologia cada vez mais moderna (CHAGAS, 2019).

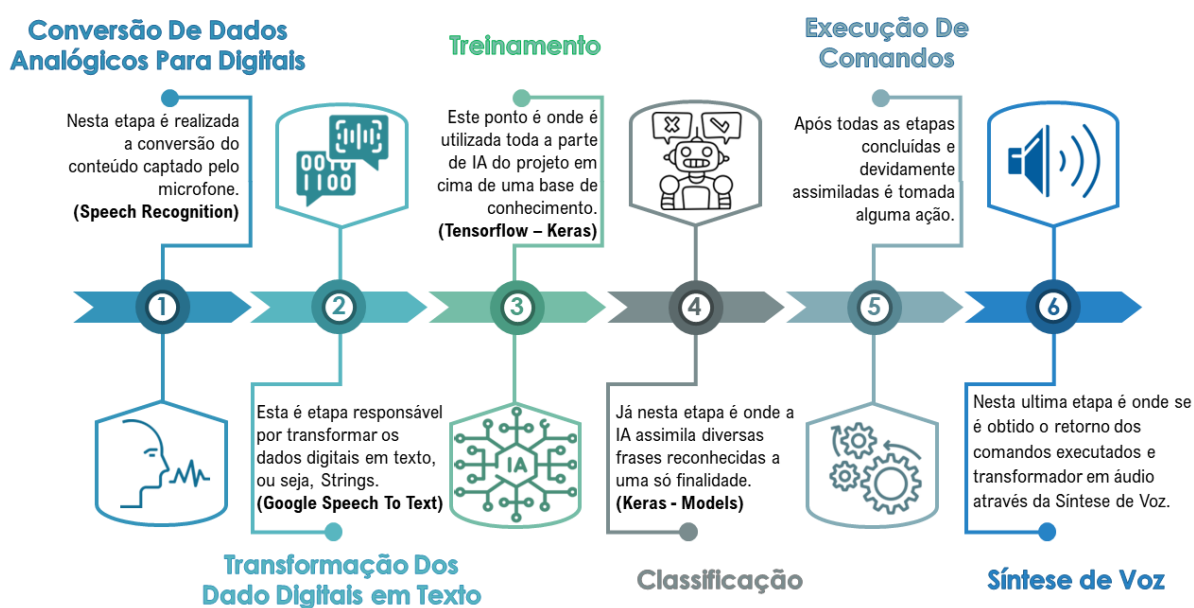
Para Chagas (2019) no que diz respeito à inteligência artificial, é visível que este tipo de tecnologia tem a capacidade de aprender, embora seja importante, ela possui características lineares e não moldáveis como do ser humano, o que é uma diferença enorme e indispensável para determinados campos do conhecimento, sendo assim havendo dificuldades de implementá-la em alguns casos.

Em qualquer caso, o uso de métodos de aprendizagem profunda permitirá que as máquinas ajudem a sociedade em várias atividades, expandam as capacidades cognitivas humanas e alcancem um maior desenvolvimento nessas áreas do conhecimento (CHAGAS, 2019).

3. PROJETO

Para o desenvolvimento deste trabalho de conclusão de curso, foi utilizado uma série de tecnologias para a construção do assistente virtual, sendo desenvolvido empregando a linguagem de programação Python, para que assim pudesse ser demonstrado o uso das ferramentas que envolvem a área da inteligência artificial com relação a assistentes virtuais, sendo o organizado da seguinte forma o projeto:

Figura 10: Resumo do projeto desenvolvido



Fonte: Elaboração Própria

Para o processamento de linguagem natural, foi utilizado um kit de ferramentas de reconhecimento de voz do Google, o Speech-To-Text. Através do resultado do reconhecimento de voz obtido pela ferramenta do Google e a transformação em texto, conforme evidenciado na imagem acima, são executadas determinadas tarefas no assistente virtual para atender as diversas necessidades dos usuários trabalhando em cima do processamento de linguagem natural e até mesmo utilização para pessoas que necessitam da acessibilidade.

Também é utilizado a biblioteca TensorFlow para demonstrar a utilização do aprendizado de máquina e aprendizado profundo no processamento de linguagem natural, com o TensorFlow, em seu modulo do Keras é que foram implementadas as redes neurais utilizadas para aprendizado da máquina de acordo com o as necessidades vigentes e funcionalidades do assistente virtual construído.

Sendo após a captação do áudio e tratamento realizado pela biblioteca Google Speech-To-Text, e logo em seguida a implementação da parte a qual compete inteligência artificial e processamento de linguagem é que se obtido a execução dos comandos instituídos no projeto de acordo com as necessidades dos usuários do assistente virtual, e sendo assim finalizado com retorno através da sintetização de voz.

3.1 SPEECH RECOGNITION

Como o ato de falar nada mais é do que criar vibrações no ar, com o auxílio de um conversor analógico-digital (ADC), essas vibrações podem ser convertidas em dados digitais que um computador pode entender. E para isto é que foi feito o uso da biblioteca Speech Recognition.

Utilizando a biblioteca foi feita a coleta e digitalização do som, fazendo medições precisas de ondas em toda a faixa de frequência. A biblioteca traz um algoritmo que filtra o som digitalizado para remover o ruído indesejado, às vezes até dividindo-o em diferentes bandas de frequência (a frequência é o comprimento de uma onda sonora, percebida pelos humanos como diferenças de tom).

O algoritmo também ajusta o som para um volume constante, pois nem sempre as pessoas falam na mesma velocidade, sendo assim o passo primário, necessitando desta etapa para que possa dar continuidade aos demais tratamentos e implementações de funcionalidades.

3.2 GOOGLE SPEECH-TO-TEXT

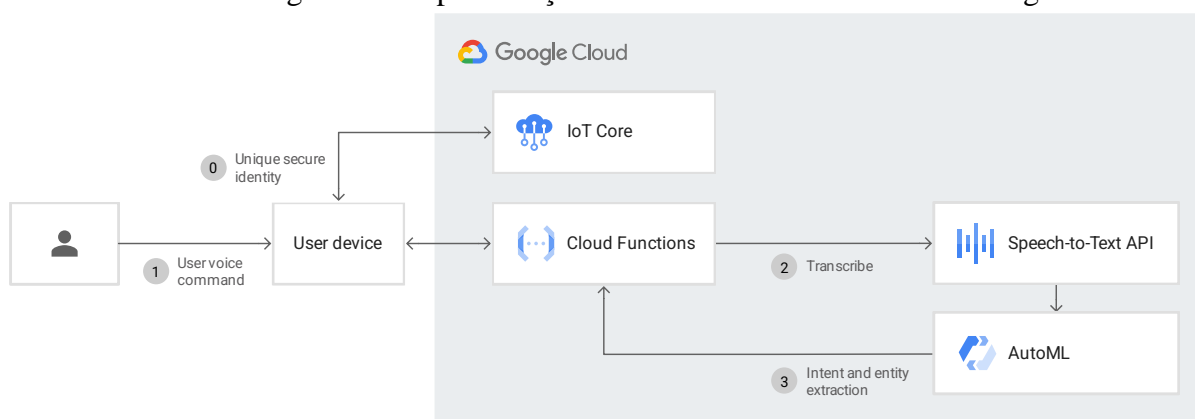
Speech-to-Text é uma tecnologia que pode transcrever a fala (linguagem falada), ou seja, convertê-la em texto. Há uma variedade de ferramentas que podem ser usadas para realizar essa tarefa, temos como por exemplo o Watson da IBM, o Azure da Microsoft, porém a utilizada no projeto deste assistente foi a tecnologia do Google.

Usado para a conversão de voz em texto de maneira precisa usando uma API desenvolvida com as tecnologias de IA do Google, com o uso dos algoritmos de rede neural para aprendizado profundo e mais avançados da API para reconhecimento automático de fala

(ASR), foi agregado uma maior qualidade no reconhecimento de fala do projeto e também uma maior acurácia, isto consumindo uma API em nuvem e com um modelo de voz muito mais amplo que os outros testados.

A API utiliza promove o uso comandos de voz mais precisos, podendo até mesmo ser utilizada combinando as funções com a API Text-to-Speech experiências ativadas por voz em aplicativos de IoT (Internet das Coisas), auxilia também na diminuição de consumo de hardware na aplicação feita, visto que todo os processos de IA, ML, IoT rodam em nuvem, como pode ser visto da ilustração a seguir:

Figura 11: Representação de funcionamento biblioteca Google



Fonte: Google (2015)

A API Speech-to-Text tem três métodos principais para realizar o reconhecimento de fala, sendo:

- I.** O reconhecimento síncrono: Envia dados de áudio para a API Speech-to-Text, realiza o reconhecimento desses dados e retorna o resultado após todo o áudio ter sido processado. As solicitações de confirmação de sincronização são limitadas a até um minuto de dados de áudio.
- II.** O reconhecimento assíncrono: Envia dados de áudio para a API Speech-to-Text e inicia operações de longa duração. Usando esta ação, você pode pesquisar periodicamente os resultados de reconhecimento.
- III.** A reconhecimento de streaming: As solicitações de streaming destinam-se a fins de reconhecimento em tempo real, como a captura de áudio em tempo real de um microfone. O reconhecimento de streaming fornece resultados temporários ao capturar áudio, por exemplo, quando o usuário ainda está falando.

Sendo neste projeto utilizado o reconhecimento por streaming, recebendo os resultados do reconhecimento de fala em tempo real conforme a API processa a entrada de áudio transmitida a partir do microfone utilizado na aplicação.

3.3 TENSORFLOW

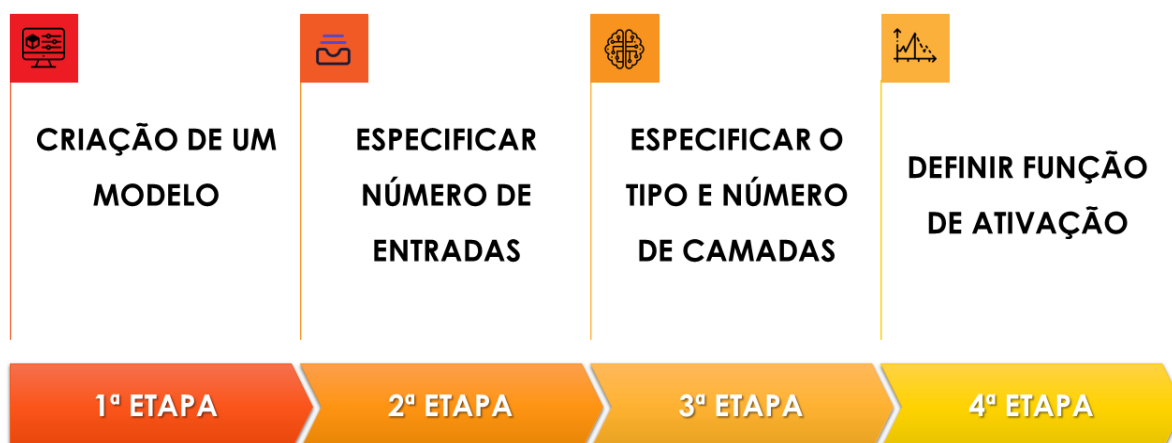
O TensorFlow é uma plataforma de código aberta para o uso de machine learning, computação numérica e outras inúmeras tarefas, sendo desenvolvida pela empresa Google, onde possui um ecossistema abrangente e flexível de ferramentas para que possa ser possível a implantação de tecnologias de machine learning em diversos projetos.

3.3.1 KERAS

No desenvolvimento deste projeto, foi utilizado a API Keras que faz parte do TensorFlow. Ela é uma API de alto nível para criar e treinar modelos de aprendizado profundo de forma rápida, fácil de usar e facilmente extensível ao longo das necessidades do projeto.

Através dessa API, conseguimos criar o modelo, um modelo Keras é definido como uma série de camadas, sendo utilizado nesta aplicação um modelo sequencial, para que isto fosse feito foram necessárias algumas etapas fundamentais, como evidenciados na imagem a seguir:

Figura 12: Implementação do Keras ao projeto

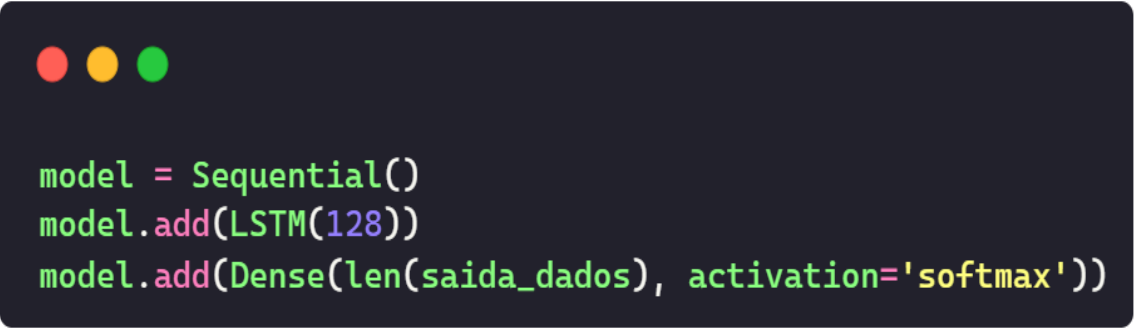


Fonte: Elaboração Própria

A início foi necessário garantir que a camada de entrada tenha o número correto de entradas, como um dos conceitos da aplicação desenvolvida foi de que ela pudesse ser facilmente expandida, isso foi realizado de forma dinâmica, adaptando-se de acordo com o tamanho das sentenças cadastrados no dataset (base de conhecimento da aplicação).

Após isto é necessário discriminar o tipo e o número de camadas, sendo na aplicação utilizada camadas totalmente conectadas do modelo sequencial, que concede ao projeto a possibilidade de trabalhar com um modelo capaz de empilhar várias camadas, podendo ser iguais ou até mesmo diferentes, como adveio ao projeto, que foram definidas e utilizadas camadas LSTM (Long Short Term Memory) e também camadas DENSE, utilizando número de camadas variáveis de acordo com saída dos dados. Como pode ser visto no seguinte trecho:

Figura 13: Modelos de redes neurais utilizadas para treinamento do projeto



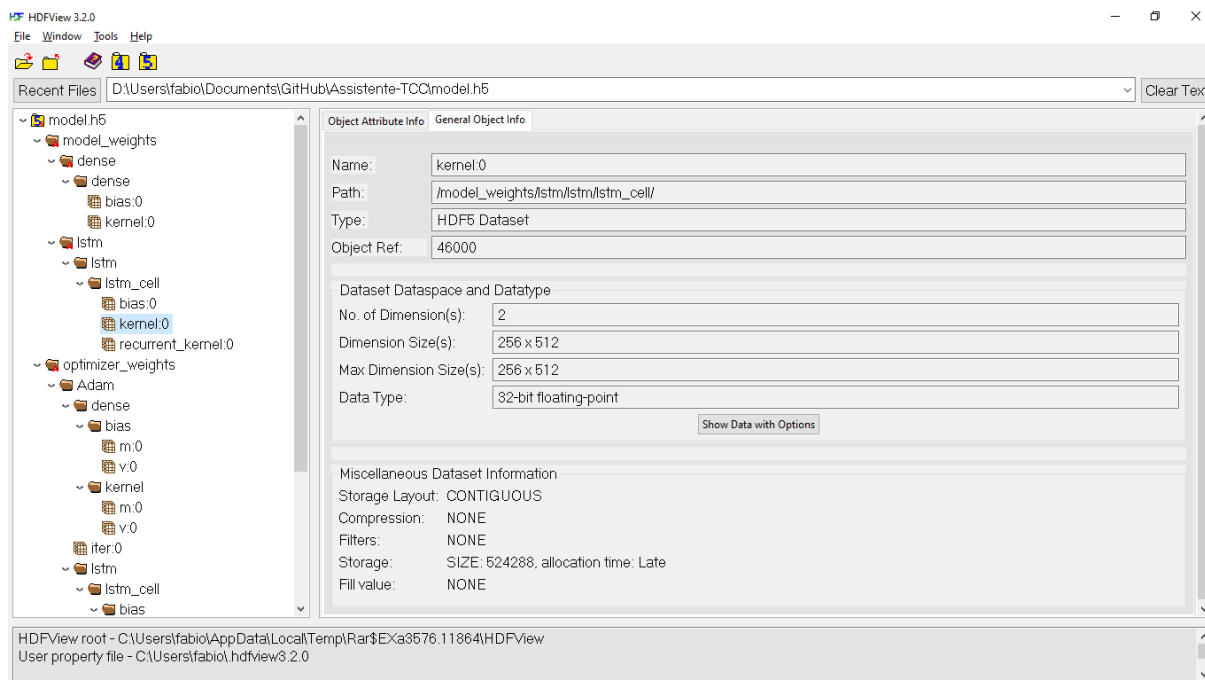
```
model = Sequential()  
model.add(LSTM(128))  
model.add(Dense(len(saida_dados), activation='softmax'))
```

Fonte: Elaboração Própria

E sendo utilizado a função de ativação softmax, a função de ativação softmax é usada na classificação de redes neurais. Ele força a saída da rede neural a representar a probabilidade de que os dados pertençam a uma das classes definidas, o qual é um dos pilares deste projeto, onde no daset foi definida um series de sentenças e a quais grupos ou classes as mesmas pertencem, assim possibilitando a previsibilidade de outras frases e assimilando a uma ação especifica a ser tomada pelo assistente. Sem a função de ativação softmax, as saídas dos neurônios são apenas valores numéricos, com o mais alto representando a classe vencedora.

Após concluídas essas etapas é que é salvo o modelo criado na primeira etapa, para que assim possa utilizar inúmeras vezes o assistente sem que o mesmo necessite de um treinando quando requisitado, para o modelo foi criado um arquivo “.H5”, os arquivos H5 são arquivos de dados salvos em Hierarchical Data Format (HDF). Contém conjuntos de dados científicos multidimensionais, como pode ser visto no arquivo modelo utilizado no projeto:

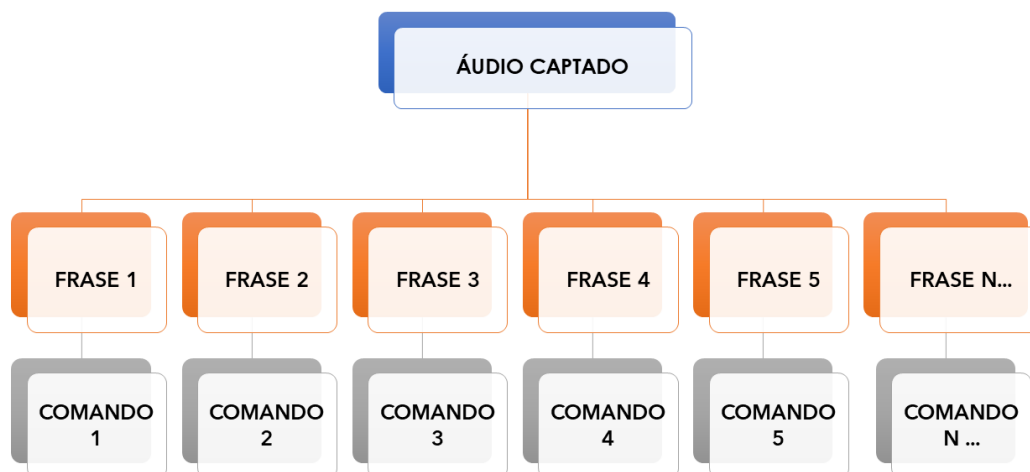
Figura 14: Arquivo conjunto de dados do modelo criado



Fonte: Elaboração Própria

Sendo este um dos principais pilares dentro do projeto, o treinamento para o processamento de linguagem natural, onde após treinado utilizando a inteligência artificial abre-se a possibilidade de teste e a otimização de funções do assistente, tornando-o mais dinâmico e efetivamente utilizado a tecnologia de processamento de linguagem natural, onde, se não houvesse estas implementações toda a parte da codificação haveria de ser estática e extremamente verbosa, como mostrado na figura abaixo:

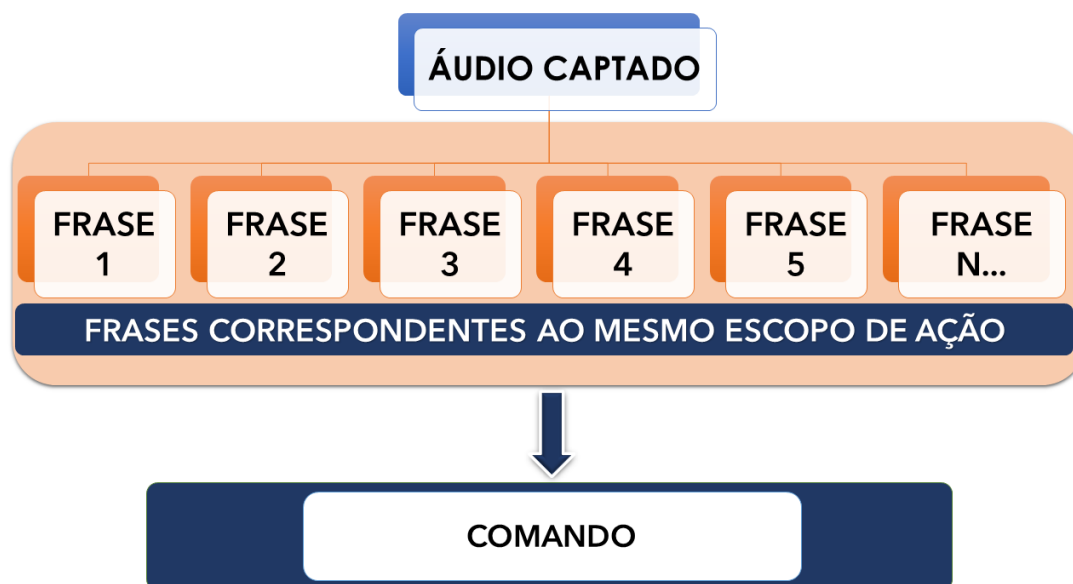
Figura 15: Modelo depreciado



Fonte: Elaboração Própria

E como descrito, quando trabalhado com inteligência artificial, com as redes neurais e com o aprendizado de máquina, a codificação e expansão do projeto torna-se de fato menos verboso e totalmente expedível, como evidenciado na imagem a seguir:

Figura 16: Modelo de PLN implementado



Fonte: Elaboração Própria

3.4 CLASSIFICAÇÃO E COMANDOS

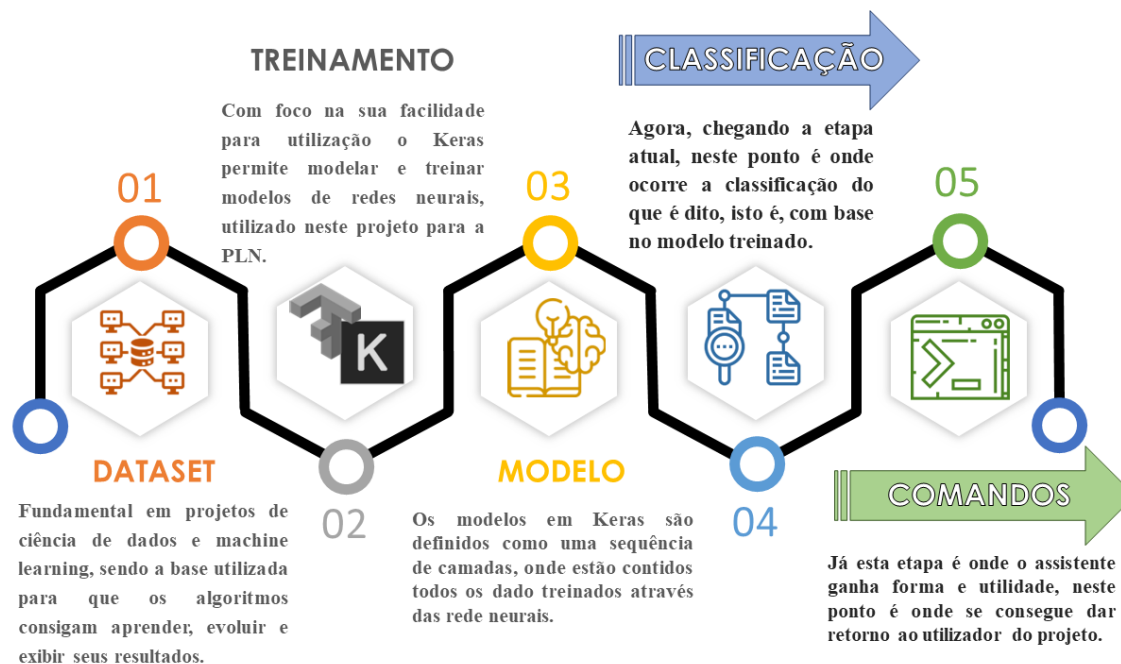
Posterior ao treinamento e complementando o projeto, vem um dos pilares mais importantes do assistente virtual e do processamento de linguagem natural, a classificação de texto, esta etapa é a responsável por colocar a prova a funcionalidade do assistente, sendo a classificação correta um pré-requisito para a etapa seguinte, funcionando como na sequência evidenciada na figura 17.

Toda a parte de classificação é baseada no modelo treinado, e é através modelo que o assistente será capaz entender e assimilar as frases ditas por nós seres humano e categorizá-las, compreendendo o real desejo do usuário com aquela fala.

Já na parte a qual trata-se dos comandos executados pelo assistente, o mesmo depende da qualidade e assertividade da classificação das frases ditas, com isto a categorização se torna certa e com baixíssimas falhas, sendo assim estas duas etapas estão altamente ligadas, ou propriamente dito, a execução dos comandos e funcionalidades do assistente depende

extremamente, se não, exclusivamente da categorização correta durante a conversa com assistente, como pode ser evidência na figura 18 um exemplo de execução de comando.

Figura 17: Etapas do projeto até a classificação



Fonte: Elaboração Própria

Figura 18: Classificação em execução

```
Run: google_main
- Audio Captado: boa noite
-----
- Assistente: Ouvindo...
Fala Reconhecida: valor do euro
- Assistente: Ouvindo...
-----
- Audio captado: valor do euro - Grupo: cotacao|retornarCotacaoAtual
-----
- Assistente: O valor do Euro é R$ 5.14, valor atualizado às 21 horas e 36 minutos
```

Fonte: Elaboração Própria

E como pode ser evidenciado no exemplo anterior, sempre que há o reconhecimento de uma frase também há a tentativa de categorizá-la, como citado e reconhecido na ilustração anterior e seguindo a nomenclatura sempre após a frase são os dados de classificação.

Sendo a classificação descrita da seguinte forma: “Grupo: GRUPO PERTECENTE | AÇÃO”, com isto sabe-se exatamente qual ação será tomada pelo assistente, onde neste caso é a cotação de uma moeda, especificamente o Euro, que é obtido através de uma função solicitada pelo assistente, como pode ser visto na captura de tela a seguir:

Figura 19: Exemplo de função categorizada por grupo

```
def obter_euro():
    with urllib.request.urlopen("https://economia.awesomeapi.com.br/json/last/EUR-BRL") as url:
        data = json.loads(url.read().decode())

        now = datetime.datetime.now()

        valor = round(float(data['EURBRL']['ask']), 2)

        resposta = 'O valor do Euro é R$ {}, valor atualizado às {} horas e {} minutos'.format(valor, now.hour, now.minute)

    return resposta
```

Fonte: Elaboração Própria

Assim finalizando a etapa de classificação e execução de comandos, etapa da torna o projeto construído praticamente um legítimo assistente virtual, faltando somente um único ciclo, o qual será descrito no modulo seguinte.

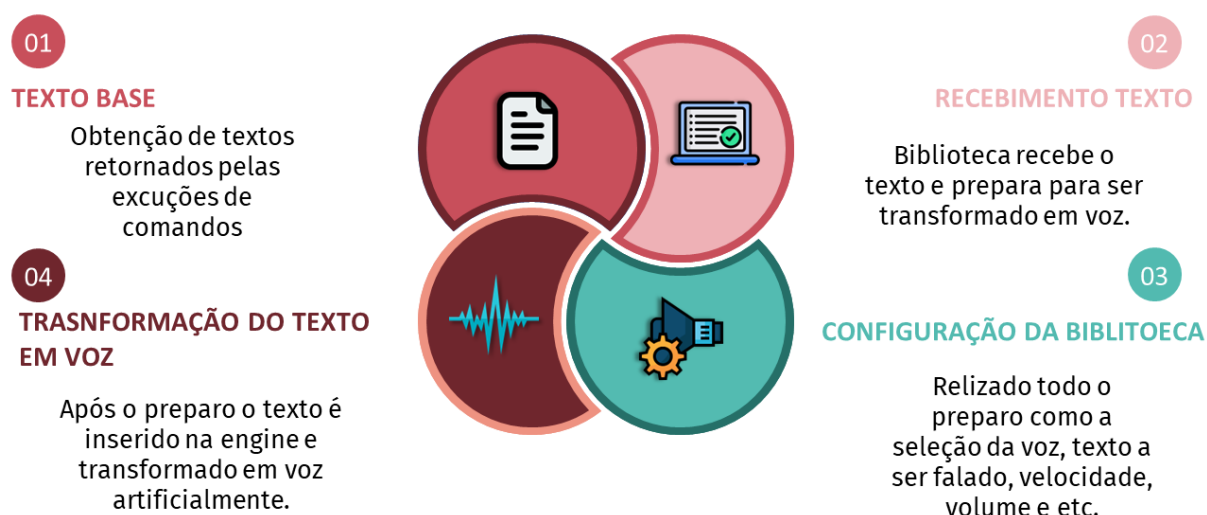
3.5 SÍNTESE DE VOZ

Este estágio do projeto é a última etapa a ser implementada, este ponto consiste em produzir a fala humana artificialmente através de um sistema TTS (Text-To-Speech), ou seja, transformando um texto em fala através de um narrador, sendo um processo totalmente ao inverso do realizado anteriormente pela biblioteca do Google (Google Speech-To-Text). Neste ciclo a síntese de voz funciona conforme evidenciado na figura 20.

E para que isto aconteça, no projeto foi utilizado a biblioteca pyttsx3 necessitando de um texto base para operar, este texto base é proveniente da etapa anterior, onde a execução de comando tem um retorno do tipo String, o qual cede a possibilidade de utilizar a transformar o texto em voz.

O PYTTX3 como descrito é uma biblioteca Python text-to-speech (TTS), a qual promove uma rápida instalação e a mesma offline para python3 e python2.

Figura 20: Estágios para sintetização de voz



Fonte: Elaboração Própria

Consequentemente tornando-se uma biblioteca muito útil, especialmente pelo uso offline no projeto, desta maneira não ocorrendo nenhum atraso no discurso reproduzido artificialmente e não dependendo apenas da Internet para a conversão de textos.

Em síntese sendo desta maneira concluída o projeto, onde este último estágio o qual a um retorno direto ao usuário, sanando suas dúvidas e o respondendo quando a execução de comandos.

4. RESULTADOS E TESTES

Segue nos próximos tópicos, capturas de telas e informações a respeito de resultados e testes ao longo do programa.

4.1 PROCESSAMENTO DE LINGUAGEM NATURAL

Para o desenvolvimento deste projeto e para a implementação do processamento de linguagem natural foram utilizadas algumas ferramentas essenciais como um reconhecedor de voz (), um tradutor de fala em texto (), esses componentes compõem o kit de ferramentas básicas para que se possa haver a troca de frases entre a pessoa e a máquina.

A primeira de ferramenta é a responsável por gerenciar o microfone do dispositivo para captação de áudio e realizar a transformação de sinais analógicos em sinais digitais, para isto foi utilizada a biblioteca Speech Recognition.

Figura 21: Etapa inicial da implantação do processamento de linguagem natural



Fonte: Elaboração Própria.

A segunda biblioteca utilizada são as bibliotecas de conversão de voz em texto, neste projeto foram utilizadas duas bibliotecas, a Google Speech To Text, esta segunda etapa complementa a anterior.

Figura 22: Complementação da implantação do processamento de linguagem natural



Fonte: Elaboração Própria.

Para conversão de voz em texto foi optado pela API do Google Speech-To-Text, em diversos testes realizados a mesma apresentou algumas vantagens sobre as demais API's testadas, como por exemplo alguns principais pontos:

- I.** Modelo de voz amplo;
- II.** Maior robustez;
- III.** Menor gasto computacional;
- IV.** Maior facilidade de implementação;
- V.** Resultados mais precisos;
- VI.** Informações mais claras e objetivas.

Porém a biblioteca utilizada possui uma desvantagem, onde não é possível utilizá-la offline.

Mesmo devido a este ponto negativo foi optado pela biblioteca do Google devido aos pontos positivos apresentados serem superiores a outras bibliotecas.

Após o ser captado e convertidos os dados de voz esta ferramenta transforma a voz recebida em texto, ou seja, string's, assim dando início ao processo de implementação do processamento de linguagem natural através da manipulação desses dados.

Dado início ao processo de manipulação dos dados de entrada para a PLN no projeto foi utilizado a tecnologia do Tensorflow para que assim possa ser implementada todas as tratativas relacionadas ao treinamento do processamento de linguagem natural.

4.2 CRIAÇÃO DE UM DATASET

Ao início da criação da aplicação foi criado um dataset, ou seja, um conjunto de dados, o principal insumo para análise e treinamento da aplicação, para criação do conjunto de dados da aplicação foi utilizado um arquivo do formato YAML para que o mesmo pudesse ser editado e expandindo, este foi responsável por conter os comandos básicos do qual após o treinamento a aplicação tomou como referência de classificação, na imagem a seguir pode ser visualizado a estrutura do dataset em questão:

Figura 23: Comandos para treinamento

```

1  comandos:
2  #Comandos relacionados ao Horário
3  - comando:
4    entrada: que horas são
5    grupo: horario
6    acao: retornarHorario
7  - comando:
11 - comando:
15 - comando:
19 - comando:
23 #Comandos relacionados a Data
24 - comando:
28 - comando:
32 - comando:
36 - comando:
40 #Comandos relacionados ao Clima
41 - comando:
42   entrada: previsão do tempo
43   grupo: clima
44   acao: retornarClima
45 - comando:
49 - comando:
53 - comando:
57 - comando:
61 - comando:
65 - comando:
66   entrada: chuva
67   grupo: clima
68   acao: retornarClima
69 #Comandos relacionados a Cotação de Moedas
70 - comando:
71   entrada: dolar
72   grupo: cotacao
73   acao: retornarCotacaoAtual
74 - comando:
78 - comando:
82 - comando:
86 - comando:
90 - comando:
94 - comando:
98 - comando:
102 #Comandos relacionados a abrir softwares e sites
103 - comando:
104   entrada: abrir navegador
105   grupo: abrir
106   acao: abrirNavegador
107 - comando:
111 - comando:
115 - comando:
119 - comando:

```

Fonte: Elaboração Própria

Sendo o dataset a base de conhecimento para treinamento do IA, tendo uma prática implementação, onde havendo a necessidade de adição de novas funcionalidades para categorização é necessário apenas poucas linhas a serem inseridas no dataset, e sem haver a necessidade de alteração em código fonte.

4.3 TREINAMENTO

Para o treinamento foi realizado o uso de bibliotecas do TensorFlow com enfoque em inteligência artificial, machine learning e deep learning, onde dela foi realizado o uso do Keras e seus componentes, também foram utilizados o Numpy e a própria biblioteca yaml para que fosse possível a manipulação e treinamento do dataset.

Como no cadastro de informações e comandos inserido no arquivo yaml são dados categóricos foi necessário realizar alguns tratamentos com o uso das bibliotecas mencionadas acima, para início de tratamento foi necessário utilizar um dos artifícios da machine learning, o One Hot Encoding para tratar os parâmetros de dados categóricos dito.

Neste caso, os conjuntos de dados encontramos colunas que contêm recursos categóricos (valores de string), como o grupo e ação, como os dados são rótulos de string, os modelos de aprendizado de máquina não podem usar esses dados, neste caso sem o One Hot Encoding os dados em String estariam da seguinte forma:

Figura 24: Dados sem One Hot Enconding



Fonte: Elaboração Própria

Já quando aplicamos o One Hot Encoding, os rótulos são codificados e é atribuído um valor numérico a esses rótulos, nesta técnica, cada parâmetro de classificação será uma coluna separada para o rótulo, ficando a seguinte maneira:

Figura 25: One Hot Encoding aplicado

	G_Horário	G_Cotação	A_ retornarHorario	A_ retomarCotacaoAtual
Frase 1	1	0	1	0
Frase 2	0	1	0	1

Fonte: Elaboração Própria

Sendo desta maneira o treinamento primordial do projeto, onde o One Hot Encoding é utilizado para categorização dos dados, e em conjunto com Numpy que é utilizado para iniciar a matriz com zeros do qual o algoritmo utilizará, acontecendo da seguinte forma:

Figura 26: Tratamento dos dados para a implementação do One Hot Encoding

```
dados_entrada = np.zeros((len(entradas), maior_sequencia, 256), dtype='float32')
for i, inp in enumerate(entradas):
    for k, ch in enumerate(bytes(inp.encode('utf-8'))):
        dados_entrada[i, k, int(ch)] = 1.0
```

Fonte: Elaboração própria

E após esse tratamento realizado foi possível produzir o treinamento da inteligência artificial, este treinamento foi realizado com 500 épocas, épocas são o número de vezes que são os dados para rede neural, esperando que o resultado nos apresente um bom desempenho ajustados com as características que queremos, no projeto foi realizado os treinamentos com a quantidade citada de épocas, como evidenciado na figura abaixo:

Figura 27: Quantidade de épocas treinadas

```

.....

Epoch 220/500
2/2 [=====] - 0s 19ms/step - loss: 0.1163 - accuracy: 0.9836
Epoch 221/500
2/2 [=====] - 0s 19ms/step - loss: 0.1134 - accuracy: 0.9836
Epoch 222/500
2/2 [=====] - 0s 17ms/step - loss: 0.1091 - accuracy: 0.9836
Epoch 223/500
2/2 [=====] - 0s 21ms/step - loss: 0.1059 - accuracy: 0.9836
Epoch 224/500
2/2 [=====] - 0s 17ms/step - loss: 0.1029 - accuracy: 0.9836
Epoch 225/500
2/2 [=====] - 0s 21ms/step - loss: 0.0993 - accuracy: 0.9836
Epoch 226/500
2/2 [=====] - 0s 21ms/step - loss: 0.0970 - accuracy: 0.9836
Epoch 227/500
2/2 [=====] - 0s 20ms/step - loss: 0.0938 - accuracy: 0.9836
Epoch 228/500
2/2 [=====] - 0s 19ms/step - loss: 0.0911 - accuracy: 0.9836
Epoch 229/500
2/2 [=====] - 0s 19ms/step - loss: 0.0885 - accuracy: 0.9836
Epoch 230/500
2/2 [=====] - 0s 19ms/step - loss: 0.0857 - accuracy: 0.9836
Epoch 231/500
2/2 [=====] - 0s 19ms/step - loss: 0.0833 - accuracy: 1.0000
.....

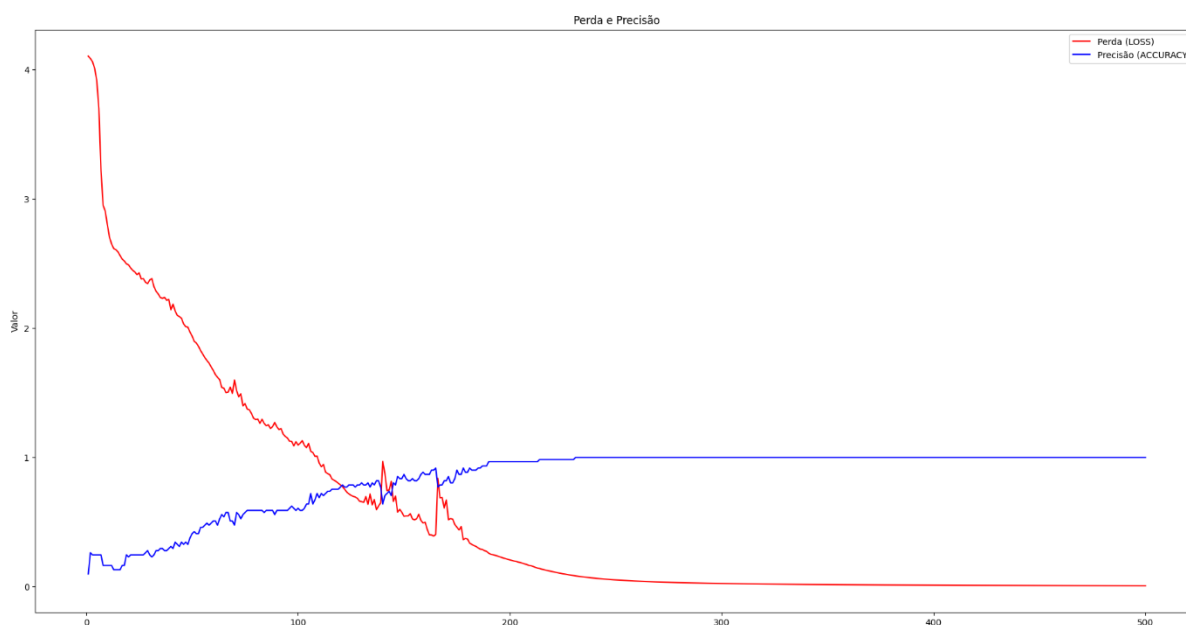
Epoch 490/500
2/2 [=====] - 0s 20ms/step - loss: 0.0071 - accuracy: 1.0000
Epoch 491/500
2/2 [=====] - 0s 19ms/step - loss: 0.0070 - accuracy: 1.0000
Epoch 492/500
2/2 [=====] - 0s 23ms/step - loss: 0.0070 - accuracy: 1.0000
Epoch 493/500
2/2 [=====] - 0s 22ms/step - loss: 0.0070 - accuracy: 1.0000
Epoch 494/500
2/2 [=====] - 0s 18ms/step - loss: 0.0069 - accuracy: 1.0000
Epoch 495/500
2/2 [=====] - 0s 18ms/step - loss: 0.0069 - accuracy: 1.0000
Epoch 496/500
2/2 [=====] - 0s 20ms/step - loss: 0.0069 - accuracy: 1.0000
Epoch 497/500
2/2 [=====] - 0s 22ms/step - loss: 0.0068 - accuracy: 1.0000
Epoch 498/500
2/2 [=====] - 0s 20ms/step - loss: 0.0068 - accuracy: 1.0000
Epoch 499/500
2/2 [=====] - 0s 19ms/step - loss: 0.0068 - accuracy: 1.0000
Epoch 500/500
2/2 [=====] - 0s 20ms/step - loss: 0.0068 - accuracy: 1.0000

```

Fonte: Elaboração própria

Como pode ser visto no exemplo anterior, o projeto foi treinado com uma grande quantidade de épocas, sendo partir de um ponto começa a tornar-se indiferente a quantidade de épocas, visto que os atributos se estabilizam, como demonstrado no exemplo abaixo:

Figura 28: Demonstração perca e precisão



Fonte: Elaboração própria

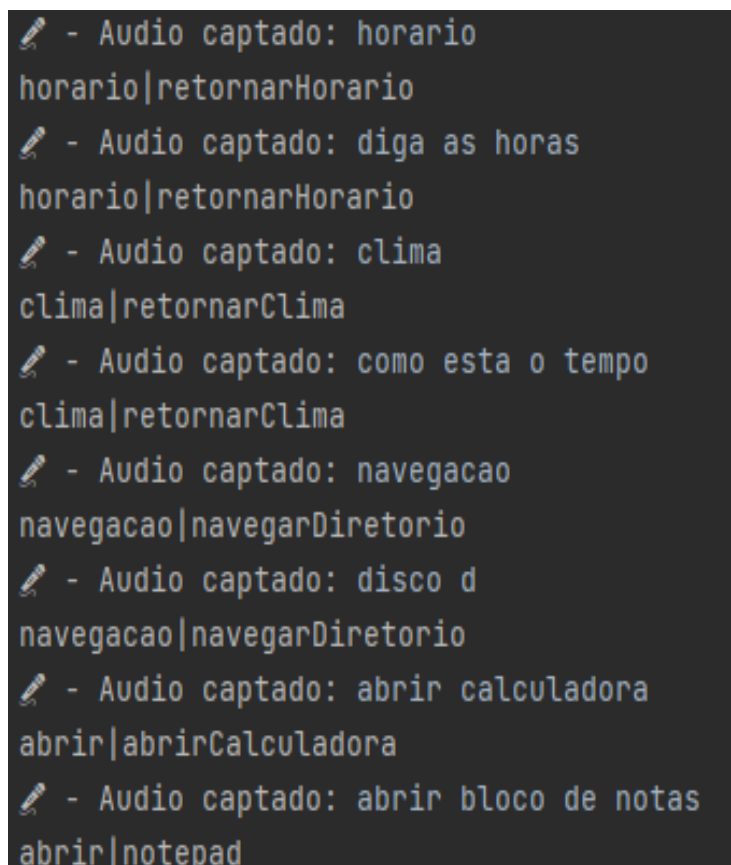
Por fim o processo de treinamento ocorre a cada volta ou a cada época, no projeto foi mantido a quantidade de épocas em 500, e após a realização do treinamento, depois de utilizar todas as épocas, tem-se um modelo treinado, que aprendeu diversas características sobre os dados apresentados e com uma ótima precisão em relação ao dataset.

4.3.1 CATEGORIZAÇÃO POR GRUPO

Após realizar o reconhecimento da fala e adquirir o texto correspondente, o resultado adquirido é submetido a uma avaliação em grupos.

Os grupos são pré-determinados contendo as palavras que são similares e que executarão uma mesma ação se a fala reconhecida pertencer ao determinado grupo. Os grupos são submetidos ao treinamento para que a inteligência artificial possa reconhecer palavras que são semelhantes as palavras ditas e reconhecidas. Abaixo, são demonstrados exemplos em que foram ditas palavras diferentes que são pertencentes a mesma ação e foram reconhecidas corretamente pela assistente virtual, conforme a figura seguinte:

Figura 29: Treinamento e classificação dos grupos



```
🖋️ - Audio captado: horario  
horario|retornarHorario  
🖋️ - Audio captado: diga as horas  
horario|retornarHorario  
🖋️ - Audio captado: clima  
clima|retornarClima  
🖋️ - Audio captado: como esta o tempo  
clima|retornarClima  
🖋️ - Audio captado: navegacao  
navigacao|navegarDiretorio  
🖋️ - Audio captado: disco d  
navigacao|navegarDiretorio  
🖋️ - Audio captado: abrir calculadora  
abrir|abrirCalculadora  
🖋️ - Audio captado: abrir bloco de notas  
abrir|notepad
```

Fonte: Elaboração Própria

4.4 EXECUÇÃO DE COMANDOS

De acordo com a classificação, os comandos são executados de acordo com o grupo a qual a determinada fala pertence. Abaixo estão contidos exemplos da execução de alguns comandos contidos nessa assistente virtual:

4.4.1 HORÁRIO

O grupo de horário possui as palavras chaves correspondentes as ações no grupo. Caso seja reconhecido uma palavra que pertença a essas palavras chaves ou que seja semelhante de acordo com o treinamento realizado, o comando de informar o horário atual será executado e apresentado ao usuário da assistente virtual, conforme a imagem abaixo:

Figura 30: Comando de horário

```
❖ - Assistente: Ouvindo...
🎧 - Audio captado: me diga as horas - Grupo: horario|retornarHorario
São 20 horas e 19 minutos.
```

Fonte: Elaboração Própria

4.4.2 DATA

Da mesma forma que é informado o horário, também é informada a data atual. Após a classificação dos grupos de acordo com a fala, o comando é executado, conforme as imagens abaixo:

Figura 31: Comando de data

```
❖ - Assistente: Ouvindo...
🎧 - Audio captado: qual a data de hoje - Grupo: data|retornarData
Hoje é dia 19 de May de 2022
```

Fonte: Elaboração Própria

4.4.3 COTAÇÃO ATUAL

No grupo referente a cotação atual, são executados comandos referentes a informação do valor de determinada moeda no momento que foi solicitado. Nessa assistente virtual, foram implementados os comandos para retornar o valor do dólar e do euro no grupo de cotação. De acordo com a fala obtida referente a moeda desejada para obter a cotação, a assistente retorna o valor, informando ao usuário tanto na fala como em texto, conforme as imagens abaixo:

Figura 32: Comando de cotação de dólar

```
❖ - Assistente: Ouvindo...
🎧 - Audio captado: qual o preco do dolar - Grupo: cotacao|retornarCotacaoAtual
O valor do Dólar é R$ 4.93, valor atualizado às 20 horas e 32 minutos
```

Fonte: Elaboração Própria

Figura 33: Comando de cotação de euro

```

❖ - Assistente: Ouvindo...
-----
✎ - Audio captado: euro - Grupo: cotacao|retornarCotacaoAtual
-----
❖ - Assistente: O valor do Euro é R$ 5.22, valor atualizado às 20 horas e 51 minutos

```

Fonte: Elaboração Própria

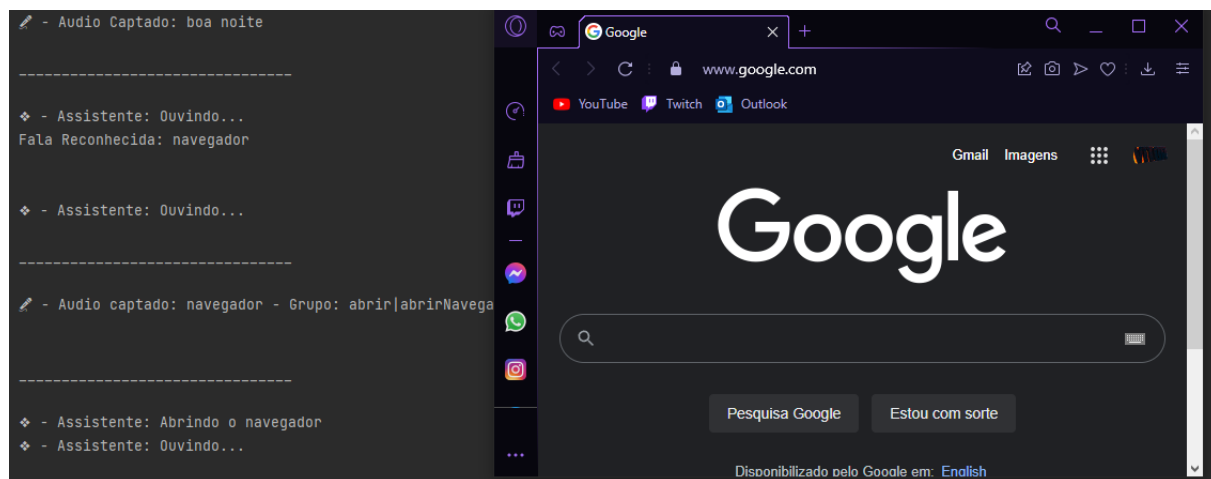
4.4.4 SOFTWARES E SITES

Este grupo executa ações referentes a abertura de sites e de softwares no Windows, de acordo com as palavras chaves existentes dentro desse grupo.

Este grupo executa as seguintes ações:

- Navegador: Esta ação executa a abertura do navegador padrão definida no Windows para poder navegar pela internet, conforme imagem abaixo:

Figura 34: Comando de abrir navegador

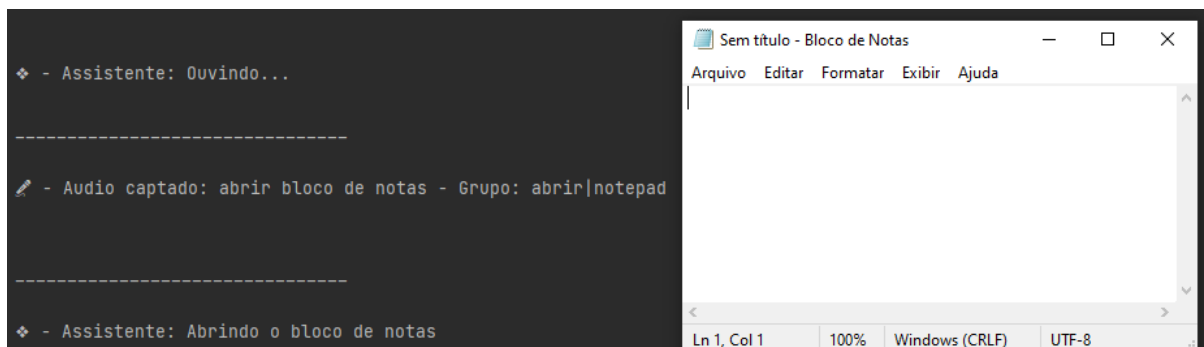


Fonte: Elaboração Própria

Em conjunto com o comando de abrir o navegador, também é possível abrir diretamente na página do Google, Youtube, Facebook ou Instagram.

- Bloco de notas: Esta ação executa a abertura do bloco de notas no Windows para realizar anotações, conforme imagem abaixo:

Figura 35: Comando de abrir bloco de notas



Fonte: Elaboração Própria

- Calculadora: Esta ação realiza a abertura da calculadora do Windows, conforme imagem abaixo:

Figura 36: Comando de abrir calculadora



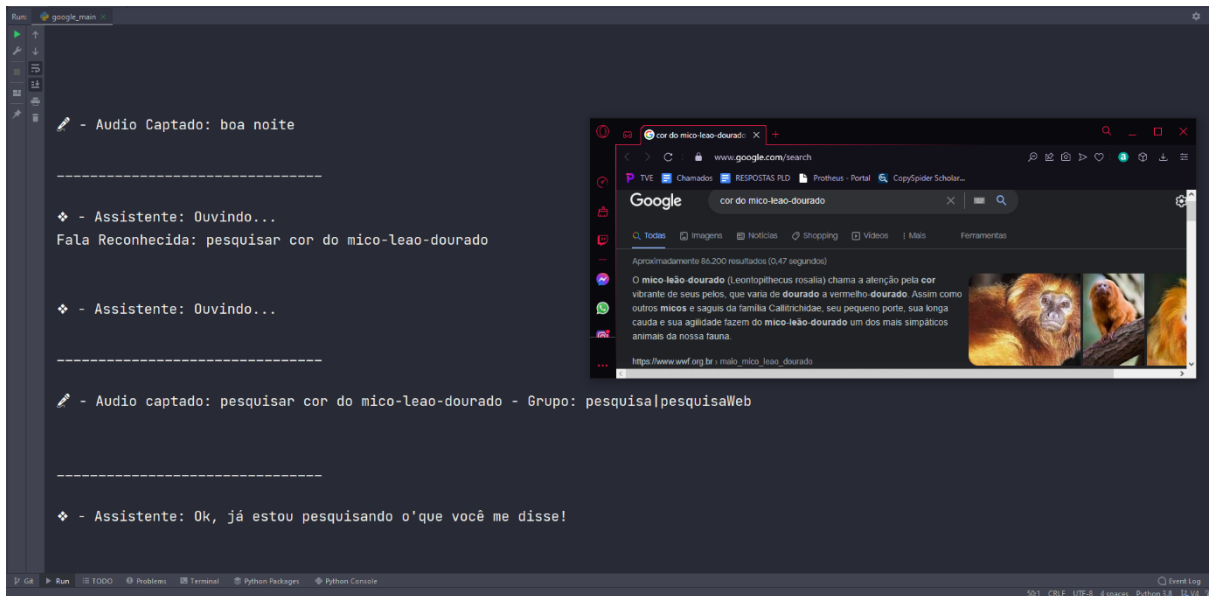
Fonte: Elaboração Própria

4.4.5 PESQUISA

O grupo de pesquisa está relacionado a execução de pesquisas na web pelo site do Google para realizar buscas de maneira rápida, informando o que deseja buscar logo após o comando de busca.

A ação de pesquisa ocorrerá no navegador padrão e será aberta uma página do Google contendo a busca correspondente ao que foi dito a assistente virtual, conforme a imagem abaixo:

Figura 37: Comando de pesquisas web



Fonte: Elaboração Própria

4.4.6 NAVEGAÇÃO

O grupo de navegação é referente a ação de realizar a navegação pelas pastas presentes no Windows, conforme são reconhecidas as falas.

A navegação é realizada continuamente enquanto o usuário desejar. Conforme são informados os diretórios que o usuário deseja navegar, a assistente mostra quais subdiretórios estão presentes naquele diretório e seus arquivos, conforme imagem abaixo:

Figura 38: Comando de navegação

```

❖ - Assistente: Ouvindo...
❖ - Assistente: Não foi possível identificar a fala. Fale novamente.
Fala Reconhecida: navegar
❖ - Assistente: Qual disco inicial ?
- Audio captado: d

-----

❖ - Assistente: Qual diretorio ?

['$RECYCLE.BIN', 'Arquivos', 'desktop.ini', 'eula.1028.txt', 'eula.1031.exe', 'install.ini', 'install.res.1028.dll', 'install.res.1031.dll', 'install.res.3082.dll', 'Jogos', 'msdownld.tmp', 'Recovery', 'Steam',

```

Fonte: Elaboração Própria

Caso o usuário deseje continuar navegando, ele informa o próximo diretório, caso o usuário queira retornar a um diretório anterior, ele pode informar o comando de voltar e caso queira abrir o arquivo, o usuário pode falar o comando de abrir e em seguida o arquivo que deseja abrir, conforme imagem abaixo:

Figura 39: Comando de navegação 2

```

✎ - Audio captado: jogos
-----
❖ - Assistente: Qual diretório ?

['Call of Duty Modern Warfare', 'DeadByDaylight', 'Epic Games', 'Fortni
-----
✎ - Audio captado: riot games
-----
❖ - Assistente: Qual diretório ?

['LoR', 'VALORANT']
-----
✎ - Audio captado: voltar
-----
❖ - Assistente: Qual diretório ?

['Call of Duty Modern Warfare', 'DeadByDaylight', 'Epic Games', 'Fortni
-----
✎ - Audio captado: voltar
-----
❖ - Assistente: Qual diretório ?

['$RECYCLE.BIN', 'Arquivos', 'desktop.ini', 'eula.1028.txt', 'eula.1031
.exe', 'install.ini', 'install.res.1028.dll', 'install.res.1031.dll',
'install.res.3082.dll', 'Jogos', 'msdownld.tmp', 'Recovery', 'Steam',

```

Fonte: Elaboração Própria

Figura 40: Comando de abertura

```

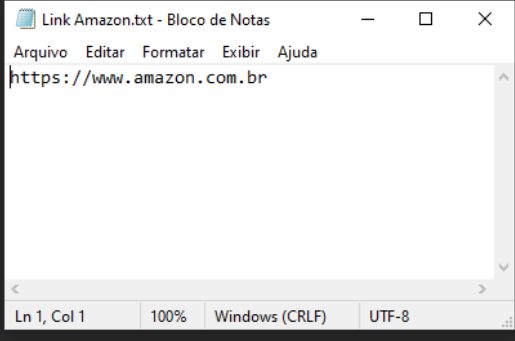
❖ - Assistente: Qual diretório ?

['$RECYCLE.BIN', 'Anotacoes', 'Arquivos', 'desktop.ini', 'eula.1028.txt', 'eula.1031.txt', 'eu
.ini', 'install.exe', 'install.ini', 'install.res.1028.dll', 'install.res.1031.dll', 'install
.dll', 'install.res.3082.dll', 'Jogos', 'msdownld.tmp', 'Recovery', 'Steam', 'System Volume I
-----
✎ - Audio captado: anotacoes
-----
❖ - Assistente: Qual diretório ?

['Link Amazon.txt']
-----
✎ - Audio captado: link amazon
-----
❖ - Assistente: Qual diretório ?

['Link Amazon.txt']

```



Fonte: Elaboração Própria

5. CONCLUSÃO

O objetivo central deste trabalho de conclusão de curso foi realizar o desenvolvimento de um assistente virtual com o uso do processamento de linguagem natural, com isto, trazendo a capacidade de assimilar comandos de voz e a que contexto está relacionado o comando em questão classificando-a desta forma, foi realizada a implementação de funcionalidades essenciais como a obtenção de dados da internet, como clima, endereços, informações básicas, implementação de pesquisas por comando de voz, a execução de comandos, manipulações de aplicação do sistema operacional, e também a manipulação de diretórios de sistema operacional por voz, assim tornando o assistente também uma ótima ferramenta de acessibilidade para que possuem alguma necessidade especial, todos esses comandos foram realizados através da utilização da inteligência artificial e seus componentes.

Todo este processo foi realizado em máquinas de padrão residencial, até mesmo por este motivo foi realizado o uso de algumas bibliotecas que ofereciam menor custo computacional e melhor rendimento para a tarefa exercida, como foi caso da ferramenta do Google, onde todo processamento é realizado do lado servidor e não cliente.

Concluindo, o projeto conseguiu demonstrar o aprendizado de máquina em conjunto com a PLN de uma forma satisfatória, obteve bons resultados de classificações utilizando as redes neurais com aprendizado profundo.

Através desse projeto foi possível identificar funcionalidades que uma assistente pode desempenhar no dia a dia que podem auxiliar e facilitar algumas tarefas, através do processamento de linguagem natural e do aprendizado é possível realizar muitas tarefas e automatizar muitas operações, otimizando funcionalidades que exijam de maior esforço para serem realizadas.

Mediante a construção deste projeto foi perceptível a complexidade da área de inteligência artificial e seus subcampos, foi perceptível as inúmeras possibilidades do que se pode criar com o uso desta tecnologia, a facilidade o auxílio e até mesmo a inclusão de pessoas com alguma necessidade especial com a ajuda desta tecnologia.

BIBLIOGRAFIA

ADAMS, Jeff. Breet Kinsella: Jeff Adams CEO da Cobalt Speech & Language e ex-Alexa

AHMAD, Syed. An introduction to neural networks. In: An introduction to neural networks. [S. l.]: Syed Ahmad, 21 mar. 2018. Disponível em: <https://codeburst.io/an-introduction-to-neural-networks-part-1-47ba9cfcd88>. Acesso em: 26 maio 2021.

ASR e NLU Teams - Voicebot Podcast Ep 59 [set. 1991]. Entrevista concedida ao Voicebot.ia

ALURA. PLN: O que é processamento de linguagem natural. Brasil, 2020. Disponível em: <https://www.alura.com.br/artigos/assets/o-que-e-nlp/imagem-1.png>. Acesso em: 4 ago. 2021.

ARARIBÓIA, G. Inteligência artificial: um curso prático. Rio de Janeiro: LTC, 1988. xvi, 282 p. ISBN 8521605919.

ARAÚJO, Allysson Alex de Paula. Uma Arquitetura Utilizando Algoritmo Genético Interativo E aprendizado de Máquina Aplicado Ao Problema do Próximo Release. [S. l.], 2015. Disponível em: https://www.researchgate.net/publication/320273267_Uma_Arquitetura_utilizando_Algoritmo_Genetico_Interativo_e_Aprendizado_de_Maquina_aplicado_ao_Problema_do_Proximo_Release. Acesso em: 30 ago. 2021.

CARVALHO, L. A. V. Data Mining: A mineração de dados no Marketing, Medicina, Economia, Engenharia e Administração. São Paulo: Ed. Érica, 2002.

CERON, Rodrigo. A Inteligência Artificial hoje: dados, treinamento e inferência. [S. l.]: IBM, 6 jan. 2020. Disponível em: <https://www.ibm.com/blogs/systems/br-pt/2020/01/a-inteligencia-artificial-hoje-dados-treinamento-e-inferencia/>. Acesso em: 30 maio 2021.

CHAGAS, Edgar Thiago De Oliveira. Deep Learning e suas aplicações na atualidade. Revista Científica Multidisciplinar Núcleo do Conhecimento. Ano 04, Ed. 05, Vol. 04, pp. 05-26 Maio de 2019. ISSN: 2448-0959

CHARNIAK, Eugene; MCDERMOTT, Drew. Introduction to Artificial Intelligence. [S. l.: s. n.], 1985.

CHEESEMAN, Peter; STUTZ, John. On finding the most probable model. In: SHRAGER, Jeff; LANGLEY, Pat. Computational Models of Scientific Discovery and Theory Formation. [S. l.: s. n.], 1990. p. 73-96. ISBN 978-1558601314.

CHURCHLAND, Ps; SEJNOWSKI, TJ. 1992. The computational brain. Cambridge, MA: MIT Press.

COPPIN, Ben. Inteligência artificial. Rio de Janeiro: LTC, 2010.

DAGLI, Rishit. Debugging your Neural Nets and checking your Gradients. Towards data science, 14 abr. 2020. Disponível em: <https://towardsdatascience.com/debugging-your-neural-nets-and-checking-your-gradients-f4d7f55da167>. Acesso em: 28 maio 2021.

DESCARTES, René. Discurso do Método. [S. l.]: Nova Fronteira, 2011. 200 p. ISBN 9788520928363.

FUSINSKA, Barbara. Networks are like onions: Practical Deep Learning with TensorFlow. [S. l.], 21 jul. 2017. Disponível em: <https://pt.slideshare.net/barbarafusinska/networks-are-like-onions-practical-deep-learning-with-tensorflow>. Acesso em: 30 maio 2021.

GOMES, Alexandre Miguel Gonçalves. Aplicação de Machine Learning no Combate Ao Branqueamento de Capitais e Ao Financiamento do Terrorismo. 2019. Mestrado (mestrado Em Métodos Quantitativos para a Decisão Económica e Empresarial) - Universidade de Lisboa, [S. l.], 2019. Disponível em: <https://www.repository.utl.pt/bitstream/10400.5/19977/1/DM-AMGG-2019.pdf>. Acesso em: 30 ago. 2021.

GOODFELLOW, Ian et al. Deep Learning. Massachusetts: MIT Press book, 2016. 800 p. ISBN 978-0262035613. Disponível em: <https://www.deeplearningbook.org/>. Acesso em: 30 maio 2021.

GOOGLE (Califórnia). Casos de uso Google Speech To Text. Cloud Speech-to-Text: Google, 2015. Disponível em: <https://cloud.google.com/speech-to-text#section-7>. Acesso em: 25 maio 2022.

GROSSBERG, S. 1988 “competitive learning: From interactive activation to adaptive resonance,” in Neural networks, Cambridge, MA:MIT Press.

- HAUGELAND, John. Artificial Intelligence. [S. l.: s. n.], 1985.
- HAYKIN, Simon. Redes Neurais: Princípios e Prática. 2ª. ed. [S. l.]: Bookman, 2000. 898 p. ISBN 978-8573077186.
- HEBB, D. O. He Organization of Behavior: A Neuropsychological Theory. 1. ed. [S. l.: s. n.], 2002. 335 p. ISBN 978-0805843002.
- KERLIRZIN, P; VALLET, F. 1993. “Robustness in multilayer perceptrons,” Neural Computation. Vol.5, pp. 447-482.
- KOVÁCS, Zsolt L. Redes Neurais Artificiais: Fundamentos e Aplicações. 4ª. ed. [S. l.]: Livraria da Física, 2006. 174 p. ISBN 978-8588325142.
- KURZWEIL, Raymond. The Age of Intelligent Machines. [S. l.]: MIT Press, 1990. 580 p. ISBN 978-0262111218.
- LECUN, Yann. Facebook Envisions AI That Keeps You From Uploading Embarrassing Pics. [S. l.]: Cade Metz, 12 set. 2014. Disponível em: <https://www.wired.com/2014/12/fb/>. Acesso em: 30 maio 2021.
- LUDERMIR, Teresa Bernarda. Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências. [S. l.], 2021. Disponível em: <https://www.scielo.br/j/ea/a/wXBdv8yHBV9xHz8qG5RCgZd/?lang=pt>. Acesso em: 28 maio 2021.
- LUGER, F. George; STUBBLEFIELD, William A. Inteligência Artificial. [S. l.: s. n.], 1993.
- MADAN, Piyush; MADHAVAN, Samaya. An introduction to deep learning. An introduction to deep learning. [S. l.], Março 2020. Disponível em: <https://developer.ibm.com/articles/an-introduction-to-deep-learning/>. Acesso em: 26 maio 2021.
- MAES, P. (1994). Agents that reduce work and information overload. Communications of the ACM, 37(7):31–40.
- MANNING, Dr. Christopher D. (2015). Computational Linguistics and Deep Learning Computational Linguistics, 41 (4), 701–707.

MCCARTHY, John. Actions and other events in situation calculus. [S. l.: s. n.], 2002. 615-628 p.

MCCULLOCH, W.S; PITTS, W.A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115–133(1943). <https://doi.org/10.1007/BF02478259>.

MITCHELL, Tom M. Aprendizado de máquina 1ª edição. [S. l.: s. n.], 1997.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre Aprendizado de Máquina. [S. l.], 2003. Disponível em: <https://dcm.ffclrp.usp.br/~augusto/publications/2003-sistemas-inteligentes-cap4.pdf>. Acesso em: 28 maio 2021.

NIELSEN, Michael A. Neural Networks and Deep Learning. Neural Networks And Deep Learning: [s. n.], 2019. Disponível em: <http://neuralnetworksanddeeplearning.com/>. Acesso em: 30 maio 2021.

OBERMEIER, Phd. Klaus K. Natural-language processing. p. 225-232 [S. l.]: Ellis Horwood, Ltd., dezembro/1987. ISBN 978-0470215289.

PRATI, Ronaldo Cristiano et al. Padronização da Sintaxe e Informações sobre Regras Induzidas a Partir de Algoritmos de Aprendizado de Máquina Simbólico. USP, 2002. Disponível em: <https://dcm.ffclrp.usp.br/~augusto/publications/2002-Prati-REIC>. Acesso em: 25 maio 2021.

ROSA, João Luís Garcia. Fundamentos da inteligência artificial. Rio de Janeiro: LTC, 2011. 228 p. ISBN 978-8521605935.

ROSENBLATT, F. The perceptron: a perceiving and recognize automaton. Report 85-460-1, Project PARA. Ithaca, NY: Cornell Aeronautical Laboratory, 1957.

RUSSEL, Stuart; NORVIG, Peter. Inteligência Artificial. 3ª. ed. [S. l.]: GEN LTC, setembro 2013. 1016 p. ISBN 978-8535237016.

TURING, Alan. Computing Machinery and Intelligence. [S. l.: s. n.], 1950. Disponível em: <http://www.inf.ufsc.br/~alvares/INE5633/ComputingMachineryAndIntelligence.doc>. Acesso em: 27 maio 2021.

VARGAS, Ana Caroline Gomes; CARVALHO, Aline Marins Paes; VASCONCELOS, Cristina Nader. Um estudo sobre Redes Neurais Convolucionais e sua aplicação em detecção de pedestres. Universidade Federal Fluminense, 15 nov. 2021. Disponível em: <http://gibis.unifesp.br/sibgrapi16/e proceedings/wuw/7.pdf>. Acesso em: 3 nov. 2021.