

## Sumário

### Introdução1

Objetivo2

### Revisão bibliográfica4

Digite o título do capítulo (nível 2)5

Digite o título do capítulo (nível 3)6

## Introdução

Desde os anos 50, os diferentes estudos na área de IA consistem em criar e manter comportamentos inteligentes e com paridade humana nas máquinas, que se sintetizou em "Como fazer as máquinas compreenderem as coisas?" (MINSKY, 1968).

Então é notado que a inteligência artificial não é algo que nasceu só no século XXI, Turing, 1950, já acreditava nisso, e deu início aos testes se baseando no modelo 'jogo da imitação' que agora também é reconhecido como 'teste de Turing', pelo pensamento "As máquinas podem pensar?".

A partir dos estudos feitos pela tese de Turing e com o crescimento da tecnologia nos últimos anos, foram desenvolvidos 'chatbots', robôs inteligentes ou robôs de bate-papo, para a simulação da interação humana.

O principal objetivo da ideia dos chamados chatbots é de o computador executar uma conversa entre máquina e humano, simular de maneira convincente como um ser humano se comportaria com outro parceiro de conversa, passando assim no teste de Turing.

Conforme, Dahiya, 2017, é comparando os padrões que se implementa um chatbot, a partir dessas comparações, as descobertas dentro do sistema são discutidas e então a conclusão é tirada no fim, devolvendo a resposta.

Os chatbots são usados principalmente em sistemas de diálogo para vários fins práticos, incluindo atendimento ao cliente ou aquisição de informações.

Um usuário pode perguntar a um chatbot uma pergunta ou um comando, e o chatbot responde ou executa a ação solicitada.

De acordo com o guia definitivo sobre chatbots, existem dois tipos principais disponíveis, um cujas funções são baseadas em um conjunto de regras e outro é a versão mais avançada que usa inteligência artificial.

Segundo Fábio Moreno, 2015, atualmente é válido notar que a interação com a paridade humana está muito mais perto com os recursos da inteligência artificial.

Dentro do ambiente estudantil foram encontrados problemas relacionados à comunicação entre alunos, professores e coordenadores para esclarecimento de dúvidas, esses problemas remetem a dificuldade de encontrar respostas rápidas sobre cursos, sobre as aulas, horários e dúvidas frequentes. Porque para acessar tais informações, os alunos devem entrar em contato com o coordenador, com a secretaria ou com o professor e esperar o tempo em que ele se encontra disponível para respondê-lo, e isso resulta em atraso no acesso às informações e aumento da mão de obra do lado da universidade.

## Objetivo

Este trabalho tem como objetivo a criação de um chatbot inteligente, configurado para receber perguntas e devolver em um tempo hábil a resposta, e a investigação das melhores tecnologias para o tornar apropriado para gerar uma comunicação digital e simplificar o atendimento para os alunos do curso em Ciência da Computação do Centro Universitário SENAC.

## Revisão Bibliográfica

### História e contexto

#### O que são chatbots?

Um chatbot, na tradução literal um robô de conversa, é um sistema humano-computador com linguagem natural e tem como objetivo simular uma conversa inteligente com um ou mais usuários por meio de voz ou texto.

Eles são usados como um mecanismo inteligente por meios de sites, aplicativos e outras plataformas digitais para conversar e responder usuários da forma mais humana possível, assim mantendo um diálogo amigável, tirando dúvidas e até mesmo oferecendo suporte o mais rápido possível de determinada plataforma em que se encontra.

#### Linha histórica dos chatbots

Em 1950, Alan Turing propôs a questão “As máquinas podem pensar?”, e a partir disso os estudos e desenvolvimento de chatbots começaram a surgir.

Para isso foi necessário pensar que as máquinas teriam que possuir uma certa inteligência para que conseguissem pensar e imitar a conversa humana, o que trouxe o nome para o Teste de Turing como “O jogo da imitação”.

O termo “inteligência artificial” foi criado pelos cientistas Newell, Simon, e J. C. Shaw, em 1956, deu início a tentativa do processamento simbólico, que seriam os sistemas que manipulassem símbolos ao invés de serem somente baseados em números.

E então desde os anos 50, os diferentes estudos na área de IA consistem em criar e manter comportamentos inteligentes e com paridade humana nas máquinas, que se sintetizou em “Como fazer as máquinas compreenderem as coisas?” (MINSKY, 1968).

Com os estudos gerados a partir da tese de Turing, foram desenvolvidos esses ‘chatbots’, para a simulação da interação humana, e o primeiro foi desenvolvido por Joseph Weizenbaum, em 1966, com o intuito de simular uma terapeuta que fazia perguntas e interagiu com o usuário de acordo com os termos inseridos durante toda a conversa, chamada Eliza.

Weizenbaum ficou surpreso como muitas pessoas não conseguiam distinguir Eliza de um psicólogo real, era implementado trabalhando com a semântica da interação do usuário, como o ‘pattern matching’. O principal método era procurar chaves, o sujeito e o verbo na frase transformando “eu” em “você”, por exemplo, além de utilizar o resto da frase para construção da mensagem (WEIZENBAUM, 1966). Devido a essa simplicidade nos códigos, Eliza ainda não conseguia manter um diálogo prolongado com o usuário, e muitas vezes retornando o que ele

dizia como forma de pergunta, isso fez com que mesmo sendo um avanço na tecnologia da época, ainda assim não passasse no teste de Turing.

Com o passar dos anos, a partir da década de 70, novos robôs inteligentes foram sendo desenvolvidos, como por exemplo, Parry, em 1972, na Universidade de Stanford, simulando uma pessoa com esquizofrenia.

Em 1988, surgiu o Jabberwacky, desenvolvido para ser um robô de conversação que passasse no teste de Turing, com uma conversa bem-humorada, não cumpria com todos os requisitos para passar no mesmo.

Em 1992, o chatbot Dr. Sbaiso, feito para o MS DOS, funcionava de forma com que a interação fosse feita de forma falada, com a voz, ainda sendo totalmente inovador, a voz não se parecia com um humano e seus dados ainda eram bem limitados, parecido com a Eliza de Weizenbaum.

Já em 1995, foi desenvolvida a ALICE (Artificial Linguistic Internet Computer Entity), um dos softwares mais famosos na área de inteligência artificial e chatbots, foi programado em AIML e baseado em .XML, mesmo não passando no teste de Turing, ALICE fornecia respostas pré-programadas de acordo com a interação do usuário e ganhou alguns prêmios nessa área, hoje é um software open source, que pode ser modificado e estudado por programadores do mundo todo.

Com o avanço do processamento de dados, da internet e da computação de um modo geral, a IBM lançou o seu chatbot, Watson, em 2006,

(CONTINUAR)

## Teste de Turing

O teste de Turing, ou jogo da imitação, tem como objetivo analisar se o chatbot consegue manter uma conversa computacional que seja praticamente imperceptível de que não seja humana.

O modo com que Turing aplicou esse teste na época, foi com que houvesse três pessoas, um interrogador, ou juiz, em uma sala separada, conversando com dois candidatos de sexos diferentes por meio de uma tele impressora, para que a voz não fosse uma característica que interferisse na decisão final e que com base nas repostas o interrogador conseguiria descobrir o sexo de cada um.

A pergunta “As máquinas podem pensar?” (TURING, 1950) é substituída por “O que vai acontecer se a máquina se passar pela parte A neste jogo” (TURING, 1950), o que traz à tona se as máquinas conseguem se passar por seres humanos em uma conversa, enganando então o interrogador ou o juiz da situação.

Nos dias atuais, segundo Luka Bradesko, a forma que o teste é aplicado, é com o observador, sendo o humano, que questiona ou dialoga com alguém através de um link no computador.

Esse “alguém” pode ser o chatbot, e tem como objetivo fazer com que o observador acredite que é outro humano, se o objetivo for alcançado, o chatbot então passa no teste de Turing.

## Arquitetura e Design de chatbots

### Funcionamento

#### Arquitetura

(IMAGEM DO FLUXO)

Segundo o Dr. John Woods e Sameera A., o design e a arquitetura de um chatbot pode ser dividida em três partes, são:

- Interface: é a parte que desempenha as atividades entre o bot e o usuário.
- Classificar: é a parte central, encontrada entre a interface (onde se recebe o input) e o Graphmaster, e seu funcionamento tem como objetivo filtrar esse input, classificá-lo em segmentos e a passagem pelos seus componentes lógicos, transferindo então a frase e/ou input para o Graphmaster.
- Graphmaster: O Graphmaster é a parte que desempenha a função de construção da resposta, lidando então com as instruções das bases de dados e organizando os conteúdos para fazer a devolução da resposta de maneira com maior paridade e entendimento humano.

As partes que constroem a arquitetura do chatbot necessitam de alguns pontos principais que serão abordados nas próximas sessões.

### Estratégias de comunicação

#### Conversão de fala para texto

A fala é um dos mais poderosos meios de comunicação, e mais natural também. Segundo Sameera Abdul-Kader, a fala é amplamente aceita como o futuro da interação com aplicativos de computador e móveis.

De acordo com pesquisas neurológicas indicam que a fala ativa mais do cérebro do que as outras 8 funções de processamento. E no livro “With Wired for Speech” de Clifford Nass e Scott Brave, mostra que ao incorporar o processamento de voz, os chatbots serão capazes de gerar uma interface sobre telefones e também rádios.

A conversão de voz em texto é chamada de ASR (automatic speech recognition) ou RAV (reconhecimento automático de voz) e o objetivo é alcançar o reconhecimento de voz de um extenso vocabulário independente de quem fala.

No livro “Statistical Pronunciation Modeling for Non-Native Speech Processing.” de Gruhn, Minker e Nakamura, mostra que a implementação e melhoria desse reconhecimento pode ser medida através de alguns fatores, tais como:

- Tamanho do vocabulário: A variação de caracteres, letras maiúsculas e minúsculas, dígitos, e milhões de palavras em vários idiomas.
- Independência do locutor: capacidade de reconhecer locutores específicos, ou seja, gerar respostas específicas usando a identidade do locutor.
- Co-articulação: capacidade de processar um fluxo contínuo de palavras. Requer tokenização e segmentação adequadas da entrada stream.
- Tratamento de ruído: capacidade de filtrar o que é a fala e o que é ruído, por exemplo, música de fundo, tráfego, etc.)
- Microfone: capacidade de processar a fala em variadas distâncias do microfone.

O processo ASR é dito então como não determinístico, porque para cada tentativa de comunicar uma palavra, o som pode ser diferente por causa do ruído ambiente, estado emocional, distância do microfone, cansaço, entre outros fatores, mas pode ser modelado como um processo estocástico. Dado um som X, gera então o fonema mais provável, palavra, frase ou sentença de todas as palavras na língua.

## Processamento de linguagem natural

Segundo Tur e Mori, no livro “Spoken language understanding: systems for extracting semantic information from speech.”, o processamento de linguagem natural, conhecido também como NLP (natural language processing), tem como objetivo obter a saída do reconhecimento automático de voz e gerar uma representação estruturada do texto, conhecida como SLU (spoken language understanding) ou no caso de uma entrada de texto, NLU (natural language understanding).

Explorando formas de extrair semanticamente informações e significados escritos e falados para criar estruturas de dados gramaticais que podem ser processados pelo gerador de respostas.

No livro “The conversational interface: talking to smart devices.” de McTear, Callejas e Griol, traz uma das formas de extrair um significado de uma linguagem natural. O chamado DA, ato de diálogo, então ele tem de reconhecer a função das frases, se são sugestões, perguntas, comandos, entre outros. Então após o reconhecimento dessa função, ela é classificada e um modelo estatístico de aprendizado de máquina é construído, usando uma série de recursos para classificar, como por exemplo, “por favor” retorna uma função de solicitação, “você é” retorna uma função de pergunta de resposta binária (sim/não), e informação sintática e semântica.

Segundo Kral e Cerisara, para iniciar o modelo de sistema de reconhecimento DA é necessário definir as principais funções, isso inclui escolher as classificações de uma forma que funcionem de uma forma generalizada para serem reutilizadas em outras frases, mas específicas o suficientes para continuar sendo relevantes para o texto alvo. Um conjunto de classificações que ganham destaque em chatbots que utilizam DAs são: DA em marcação de várias camadas (DAMSL), Switchboard SWBD-DAMSL, gravador de reunião, VERBMOBIL e Map-Task.

James Allen e Mark Core, explicam que o esquema DASL classifica a frase em quatro dimensões, sendo elas:

- **Status comunicativo:** A classifica como não interpretável, abandonada ou fala interna.
- **Nível de informação:** A classifica como tarefa, gerenciamento de tarefas, gerenciamento de comunicação ou outro.

- **Funções voltadas ao futuro:** Codificam qualquer informação que afetará conversas e classificações futuras em oito subdimensões, sendo:
  - Declaração: afirmar, reafirmar ou outro.
  - Influenciar: opção aberta ou diretiva de ação.
  - Solicitação de informações.
  - Comprometer o orador a ações futuras: oferecer ou compromisso.
  - Convencional abertura ou fechamento.
  - Performativo explícito.
  - Exclamação.
  - Outros.
- **Funções voltadas ao passado:** Codificam a relação entre o texto atual e o anterior em:
  - Acordo.
  - Entendimento.
  - Resposta
  - Relação de informação.

Dan Jurafsky, Liz Shriberg e Debra Biasca, afirmam que o Switchboard SWBD-DAMSL é uma adaptação do DAMSL para automatização de conversas telefônicas. Dhillon, Bhagat, Carvey, e Shriberg, afirmam que o gravador de reunião (MRDA) é semelhante ao Switchboard, mas com classificações de 72 horas de reuniões, e lidando bem com rodeios e complicações típicas durante reuniões, como sobreposição de alto-falantes, frequência de abandono de comentários, interações e tomadas de voz. E o Map Task, segundo Carletta, Isard, Doherty-Sneddon, Kowtko, e Anderson, é uma hierarquia de níveis, onde o primeiro classifica as transações, o segundo são jogos conversacionais que classificam padrões como pares de perguntas e respostas, e o terceiro inclui 19 movimentos conversacionais.

Voltando no livro “The conversational interface: talking to smart devices.” de McTear, Callejas e Griol, temos que a principal responsabilidade do reconhecimento de voz não é apenas entender a função das frases, mas também compreender o significado do próprio texto.

Para extrair o significado do texto, temos que converter os textos não estruturados, sendo eles saídas do ASR ou o texto escrito como entrada, em objetos de dados gramaticais, que serão processados pelo DA.

O primeiro passo neste processo de extração é quebrar uma frase em tokens que representam cada parte do seu componente, sendo palavras, dígitos, sinais de pontuação ou outros. Essa transformação para tokens pode ser mais complexa devido a entradas ambíguas ou mal formadas, como contradições, abreviações e pontuações, que para desenvolverem uma série de estruturas de dados diferentes para serem processadas pelo gerenciador de diálogos podem ser analisados utilizando as seguintes técnicas:

- **Amontoado de palavras:** Tem como objetivo formar um modelo de espaço vetorial, para isso são ignoradas a estrutura, a ordem e a sintaxe das frases, contando o número de ocorrências de cada palavra, com isso as palavras de parada, como artigos, são removidas, e as variantes morfológicas passam pelo processo de lematização onde são armazenadas como um instância do lema básico. Possui uma abordagem simples, por ignorar a sintaxe das frases, e por esse motivo não é tão precisa para problemas mais complexos.
- **LSA (Latent Semantic Analysis) ou análise semântica latente:** Tem uma atuação parecida com o amontoado de palavras. Conceitos são a unidade básica de

comparação analisada a partir da frase. Depois, as palavras que se repetem são agrupadas. É criado então, uma matriz onde cada linha representa uma palavra, cada coluna um documento e o valor de cada célula é a frequência da palavra no documento. É calculado a distância entre o vetor que representa cada texto e documento, usando a decomposição de valor singular para reduzir a dimensionalidade da matriz e determinar o documento mais próximo.

- **Expressões regulares:** Frases podem ser tratadas como expressões regulares e podem ser padronizadas com os documentos no banco de dados existente na base de conhecimento do bot. Por exemplo, em um dos documentos na base de conhecimento do bot lida com o caso em que o usuário insere a frase: "meu nome é \*". "\*" É o caractere coringa e indica que essa expressão regular deve ser acionada sempre que o bot ouvir ou ler a frase "meu nome é" seguida por qualquer coisa. Se o usuário disser "meu nome é Jack", essa frase será analisada em várias expressões regulares, incluindo "meu nome é \*" e acionará a recuperação desse documento.
- **Marcações de partes do texto:** Essa marcação classifica cada palavra no texto de entrada de acordo com sua classe gramatical, podendo ser substantivo, verbo, adjetivo e outros. Essas classificações podem ser baseadas em regras criadas manualmente para especificar a classe gramatical de palavras ambíguas de acordo com o contexto da frase, também podem ser criadas usando modelos estocásticos que treinam em frases marcadas com a parte correta do texto. No gerenciador de diálogos, a marcação de parte do texto pode ser usada para armazenar informações relevantes no histórico de diálogos. E também é usado na geração de respostas para indicar o tipo de objeto da resposta desejada.
- **Reconhecimento de entidades nomeadas (NER):** Nesse caso, o nome de pessoas, lugares, grupos e locais são extraídos e classificados de acordo. Os pares de nomes podem ser armazenados pelo gerenciador de diálogos no histórico para acompanhar o contexto da conversa. A extração de relação vai um passo adiante para relações de identidade (por exemplo, "quem fez o quê a quem") e classifica cada palavra nestas frases.
- **Rotulagem de função semântica:** Nesse processo, o predicado é rotulado primeiro, logo após vem seus argumentos. Classificadores proeminentes para rotulagem de função semântica foram treinados no FrameNet e PropBank, bancos de dados com frases já classificadas com suas funções semânticas. Esses pares de palavras-funções semânticas podem ser armazenados pelo gerenciador de diálogos no histórico de diálogos para manter o controle do contexto.
- **Criação de estrutura de dados gramaticais:** Frases e enunciados podem ser armazenados de forma estruturada em formalismos gramaticais, como gramáticas livres de contexto, que são estruturas de dados semelhantes a árvores que representam sentenças contendo frases nominais e verbais, cada uma delas contendo substantivos, verbos, sujeitos e outras construções gramaticais, e gramáticas de dependência que por outro lado, focam nas relações entre as palavras.

## Gerador de resposta

De acordo com Wallace, R. S., o gerador de respostas é o componente central da arquitetura de um chatbot. Recebendo uma representação estruturada do texto falado e devolvendo uma resposta para entregar ao usuário, que em seguida também passa a ser guardado no gerenciador de diálogo.



Para a tomada de decisão sobre a resposta a ser dada, existem três componentes importantes:

- Uma base de conhecimento, ou banco de dados, que irá analisar de acordo com o que foi implementado.
- Um histórico de dados de diálogos, modelos mais complexos de chatbots possuem a capacidade de armazenamento do histórico.
- Uma fonte de dados externa, ou a "inteligência do senso comum" que possibilita o bot a fazer pesquisas em sites de busca para ser alimentado.

Modelos baseados em regras tem como principal base de conhecimento os documentos que contém um <padrão> e um <modelo>. Assim que o chatbot recebe uma entrada que se encaixa no <padrão>, faz com que retorne o modelo correspondente como saída. Wallace diz que geralmente esses pares são feitos a mão, e trabalham identificando expressões regulares.

Já Zhao Yan et al., trazem a especificação desses pares serem feitas a mão como um problema, e passam o conceito do modelo baseado em recuperação de dados, trabalhando com <status> e <resposta>, ao invés de <padrão> e <modelo>. Então ele recebe uma entrada e procura no histórico os pares de <status> e <resposta> correspondente. Mas, esse modo de responder ainda traz um grande desafio em como conduzir a forma com que é feito a correspondência, ele parece mais intuitivo para achar o <status> mais parecido com os dados, e como uma forma mais efetiva seria comparar entre a <resposta>, porque se uma palavra aparecer mais de uma vez entre a <resposta> e a entrada faz com que se torne uma resposta de saída mais efetiva, mas não só porque as palavras aparecem novamente na entrada e na <resposta>, e é corresponde ao <status> não significa que é uma combinação exata, porque não se sabe se a <resposta> é realmente apropriada para aquele caso. Da mesma forma, se o sistema encontrar essa correspondência entre as respostas

## Base de conhecimento

A base de conhecimento de um chatbot é a principal vertente de sua inteligência. Todos os dados que ele recebe por meio dessas bases são utilizados para a construção dos modelos para a devolutiva correta ao input do usuário.

Essas bases de dados podem ser coletadas de várias formas, e esses dados podem ser armazenados, treinados para utilização de inteligência artificial para serem classificados como dados que podem ser devolvidos ao usuário com entendimento como interação humana, e podem ser coletados através de:

- **Fóruns de discussão online:** Jizhou Huang, Ming Zhou e Dan Yang geraram uma base de conhecimento que busca em fóruns online maneiras de respostas baseadas em <input><response>. Tendo benefícios como a quantidade de tópicos e assuntos diferentes abordados nesses fóruns com diferentes respostas ou soluções para a mesma pergunta. Mas como desvantagem traz a não garantia da qualidade das respostas, tendo em vista que são publicadas por vários usuários tendo eles conhecimento ou não sobre o assunto, podendo também trazer respostas curtas e não definidas, e também pode trazer erros de comunicação pensando que os fóruns são baseados em threads de resposta, então aquela <response> pode não corresponder exatamente aquele <input>. Para resolver isso, eles desenvolveram uma forma de tratar e treinar uma máquina de vetor classificador (SVM) de forma com que ele

identifique as respostas e perguntas diretamente do root, a thread que elas pertencem e a forma com que são usadas as palavras chaves. Depois dessa classificação pelo SVM, são aplicados filtros para remover respostas que utilizam palavras obscenas, ou com respostas pessoais do achismo (por exemplo, “meu”, “opinião”) e respostas que foram classificadas pelo SVM como fora da thread. Devolvendo então o par <thread title><response> para treinar o bot dessa forma.

- **AIML:** “Artificial intelligence markup language”, Madhumitha, Keerthana Mrs.Hemalatha, explicam que o AIML funciona como forma de simplificar o modelo de diálogo, trabalhando com a definição da classe objeto que é responsável para fazer o molde dos padrões de conversa, responde de acordo com a conexão entre as questões previamente implantadas localizadas nos arquivos AIML definidos. Também possui suas vantagens e desvantagens, entre as vantagens temos a fácil implementação e aprendizado, o modo simples do sistema de diálogo deixando *user-friendly* e o uso de XML representando uma forma de leitura para o computador, e entre suas desvantagem temos o conhecimento fornecido pelos arquivos, por precisar de atualização manual para implantação de novas perguntas e respostas por não existir extensões possíveis para o AIML Original, mas podendo trabalhar com o AIML 2.0 acessando recursos externos com comandos para o Assistente Virtual de respostas.
- **RiveScript:** Funciona como o AIML, só que com algumas funcionalidades a mais, como o uso simplificado de expressões regulares de gatilhos de input de usuários, e pode ter integração com mecanismos ou sites de pesquisa para respostas dinâmicas.
- **Computação em nuvem:**

## Gerenciamento de diálogo

O gerenciamento do diálogo funciona para transformar a resposta em uma forma mais parecida com a forma humana de responder, podendo utilizar estratégias de comunicação

Entre essas estratégias temos os truques de linguagem que são utilizados em caso de baixa probabilidade de resposta apropriada, Zhou Yu, Ziyu Xu, Alan W Black e Alex I. Rudnicky, mostram que isso pode fazer com que o bot pode continuar a conversa e responda como: uma mudança de tópico, ou seja, propondo outros tópicos para ter novos dados ou melhor forma de responder ao usuário, pode também fazer uma pergunta aberta, que seria além de dizer que não sabe responder especificamente o que o usuário disse, faz uma pergunta em troca, para assim manter a conversa, pode também pedir ao usuário que diga mais informações sobre o que ele deseja, então além de continuar a conversa ainda tem chances de fazer a devolutiva de uma resposta mais apropriada ao usuário.

Uma interface que traz uma fluidez na conversa com a proximidade humana traz alguns conceitos de design que são importantes como o modo de transformar a entrada do usuário em uma entrada com mais clareza se tiver duplo sentido, habilidade de eliminar restrições para continuar a conversa com outra pergunta se necessário, confirmar detalhes de tarefas que o bot deve tomar para decisões importantes e perguntar detalhes necessários que aparentemente o usuário se esqueceu de informar.

## Desenvolvimento

PRETENSÃO DO QUE FAZER, COMO, USANDO O QUE...

IDEIA <https://graspcoding.com/retrieval-based-chatbot-ai-free-code/>

Método utilizado

Construção do chatbot

Testes

Avaliações

Conclusão

## RASCUNHO DE IDEIAS

Escrever tudo o que vamos utilizar em relação a técnica e tecnologia, e descrever em Arquitetura e design

Phyton

Deteção de intenção e um banco de respostas

Aluno: “Quero saber o horário da aula”

Chatbot: “De qual dia?”

OU

Aluno: “Quero saber o horário da aula de segunda”

Chatbot: Envia horário da aula de segunda

Usar rede neural para testar e ensinar com várias frases de exemplo

Cada vez que ele não entender uma frase, guardá-la para alimentar a base depois