

Beatriz Paiva Alves e Lucas Breur

**Desenvolvimento de *chatbot* que simule
interações humanas para tirar dúvidas de alunos
do curso de Ciência da Computação do Centro
Universitário Senac**

São Paulo - Brasil

2021

Beatriz Paiva Alves e Lucas Breur

Desenvolvimento de *chatbot* que simule interações humanas para tirar dúvidas de alunos do curso de Ciência da Computação do Centro Universitário Senac

Monografia apresentada na disciplina Trabalho de Conclusão de Curso, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

Centro Universitário Senac - Santo Amaro

Orientador: Prof. M.Sc. Thiago Ribeiro Claro
Coorientador: Prof. M.Sc. Rodrigo Assirati Dias

São Paulo - Brasil

2021

Sumário

1	INTRODUÇÃO	3
1.1	Objetivo	4
2	REVISÃO BIBLIOGRÁFICA	5
2.1	História e contexto	5
2.1.1	O que são <i>chatbots</i> ?	5
2.1.2	Linha histórica dos chatbots	5
2.1.3	Teste de Turing	6
2.2	Arquitetura e design de <i>chatbots</i>	7
2.3	Arquitetura	7
2.3.1	Estratégias de comunicação	7
2.3.1.1	Conversão de fala para texto	7
2.3.1.2	Processamento de linguagem natural	8
2.3.1.3	Gerador de resposta	11
2.3.1.4	Base de conhecimento	12
2.3.1.5	Gerenciamento de diálogo	13
3	DESENVOLVIMENTO	15
3.1	Metodologia	15
	Referências	17

1 Introdução

Desde os anos 50, os diferentes estudos na área de Inteligência Artificial (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](#)).

É notado que a inteligência artificial não é algo que nasceu só no século XXI, Turing, em 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.

1.1 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.

2 Revisão Bibliográfica

2.1 História e contexto

2.1.1 O que são *chatbots*?

Um *chatbot*, na tradução literal é um robô de conversa, 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.

2.1.2 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.

O termo Inteligência Artificial foi criado pelos cientistas Newell, Simon, e J. C. Shaw, em 1956, deram início à 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 interagia 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. O principal método utilizado 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, trabalhando com a semântica da interação do usuário, como o *pattern matching* ([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. Sbaitso, 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 *Artificial Linguistic Internet Computer Entity* (ALICE), 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 de código aberto, 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, segundo Tahiana D'Edgemont, os tipos de uso de aplicações cognitivas incluem entender emoções, interpretar textos e imagens, dar respostas, ouvir sons entre outros.

O grande crescimento da tecnologia trouxe em 2010 a Siri da *Apple*, em 2012 o Google Now da *Google*, em 2015 a Alexa da *Amazon* e a Cortana da *Microsoft*, que são considerados grandes *chatbots* de assistência virtual usando linguagem natural para responder questões, fazer recomendações e ações utilizando os sistemas da internet para os usuários das plataformas que eles estão inseridos.

2.1.3 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?” é substituída por “O que vai acontecer se a máquina se passar pela parte 'A' neste jogo”, 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.([TURING, 1950](#))

Nos dias atuais, segundo Bradeško e Mladenić (2012), 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.

2.2 Arquitetura e design de *chatbots*

2.3 Arquitetura

Segundo Abdul-Kader e Woods (2015), o design e a arquitetura de um *chatbot* pode ser dividida em três partes, são:

1. *Responder*: é a parte que desempenha as atividades entre o bot e o usuário.
2. *Classifier*: é a parte central, encontrada entre o *Responder* (onde se recebe a entrada) e o *Graphmaster*, e seu funcionamento tem como objetivo filtrar essa entrada, classificá-la em segmentos e a passagem pelos seus componentes lógicos, transferindo então a frase e/ou entrada para o *Graphmaster*.
3. *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 seções.

2.3.1 Estratégias de comunicação

2.3.1.1 Conversão de fala para texto

A fala é um dos mais poderosos meios de comunicação, e mais natural também. Segundo Abdul-Kader e Woods (2015), a fala é amplamente aceita como o futuro da interação com aplicativos de computador e dispositivos móveis.

De acordo com pesquisas neurológicas indicam que a fala ativa mais do cérebro do que as outras 8 funções de processamento. Callaway e Sima'an (2006), mostram que aos *chatbots* incorporarem o processamento de voz, serão então capazes de gerar uma interface sobre telefones e também rádios.

A conversão de voz em texto é chamada de *Automatic Speech Recognition (ASR)* ou Reconhecimento Automático de Voz (RAV) e o objetivo é alcançar o reconhecimento de voz de um extenso vocabulário independente de quem fala.

Nass e Brave (2007), mostram 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 do fluxo de entrada.
- 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.

2.3.1.2 Processamento de linguagem natural

Segundo Tur e De Mori (2011), o processamento de linguagem natural, conhecido também *Natural Language Processing (NLP)*, tem como objetivo obter a saída do reconhecimento automático de voz e gerar uma representação estruturada do texto, conhecida como *Spoken Language Understanding (SLU)* ou no caso de uma entrada de texto, *Natural Language Understanding(NLU)*.

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.

Conforme Mctear, Callejas e Griol (2016), uma das formas de extrair um significado de uma linguagem natural é com o *Dialogue Act (DA)*, ou 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, Pavelka e Cerisara (2008), 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 DA's são: *Dialog Act Markup in Several Layers (DAMSL)* que é marcação do ato de dialogo em várias camadas, *Switchboard SWBD-DAMSL*, gravador de reunião, *VERBMOBIL* e *Map-Task*.

Allen e Core (1997), explicam que o esquema DAMSL 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.

Jurafsky, Shriberg e Biasca (1997), afirmam que o *Switchboard SWBD-DAMSL* é uma adaptação do DAMSL para automatização de conversas telefônicas. Shriberg et al. (2004) et. al, afirmam que o gravador de reunião 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 et al. \(1997\)](#), 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.

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. ([MCTEAR](#); [CALLEJAS](#); [GRIOL, 2016](#))

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 uma 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.
- *Latent Semantic Analysis (LSA)* 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 criada 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 calculada 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, um desses documentos 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 (REN):** 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.

2.3.1.3 Gerador de resposta

De acordo com [Wallace \(2009\)](#), 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. Richard diz que geralmente esses pares são feitos a mão, e trabalham identificando expressões regulares.

Já [Yan et al. \(2016\)](#), 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.

2.3.1.4 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 a entrada 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: [Huang, Zhou e Yang \(2007\)](#) 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 (MVC) de forma com que ele identifique as respostas e perguntas diretamente da raiz, a *thread* que elas pertencem e a forma com que são usadas as palavras chaves. Depois dessa classificação pela MVC, 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 pela MVC como fora da *thread*. Devolvendo então o par *<thread title><response>* para treinar o *bot* dessa forma.
- *Artificial Intelligence Markup Language (AIML)*: [Madhumitha. S Keerthana. B \(2019\)](#), 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 o deixando intuitivo 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 entrada de usuários, e pode ter integração com mecanismos ou sites de pesquisa para respostas dinâmicas.

2.3.1.5 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, Yu, Rudnicky e Black (2017), 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.

3 Desenvolvimento

Este capítulo apresenta uma descrição sobre a arquitetura e tecnologias escolhidas para o desenvolvimento de um *chatbot*. O público-alvo serão os alunos do curso de Ciência da Computação do Senac Santo Amaro, com o objetivo de tirar dúvidas e agilizar a comunicação que haveria entre a secretaria/o coordenador e os mesmos.

3.1 Metodologia

A fala é amplamente aceita como o futuro da interação com assistentes eletrônicos conforme estabelecido na seção 2.3.1.1, porém, com o intuito de tornar o escopo do projeto mais administrável, optou-se pela interação por texto. Desta forma, o processamento de linguagem natural será realizado por um *software NLU*.

As seguintes tecnologias serão utilizadas para atender a arquitetura em três partes exposta no início do capítulo 2.3:

1. *Responder*: composto por uma caixa de digitação e painel de visualização desenvolvidos em Python com o auxílio da biblioteca *tkinter*; esta é uma ponte em Python para criação de telas interativas através do *toolkit Tk*. O *Responder* é responsável por transmitir para e receber textos do *Classifier*.
2. *Classifier*: conforme citado anteriormente no capítulo em questão, é a seção responsável pelo processamento da entrada e posterior transferência para a última parte, o *Graphmaster*. Neste módulo encaixa-se o software *NLU*. Será utilizado o *Natural Language Processing Toolkit (NLTK)*, um *plugin* gratuito para Python mencionado também por [Abdul-Kader e Woods \(2015\)](#) como uma plataforma amplamente utilizada.
3. *Graphmaster*: o cérebro do *software* onde são produzidas as respostas para a entrada pré-processada recebida do *Classifier*. O treinamento será efetuado com auxílio de *deep learning* através da biblioteca *Keras*; esta permite um desenvolvimento de forma *‘fail fast, fail early’* em linha com nossa abordagem holística do projeto.

A base de dados será armazenada em nuvem na plataforma Azure e receberá os dados primeiramente através de um formulário, previamente alinhado com o coordenador do curso, enviado para os alunos de diferentes semestres do curso de Ciência da Computação com perguntas como:

1. Quais foram as principais dúvidas que você teve ao entrar no curso? (Campo aberto)

2. Quais dúvidas você tem com frequência para perguntar ao coordenador? (Campo aberto)
3. Ao entrar no curso ou até mesmo no dia a dia, você teve alguma das dúvidas que teve que esclarecer com o coordenador ou com os professores do seu curso? Selecione com base na frequência deste questionamento:
 - “Eu preciso saber matemática?”
 - “Qual nível de inglês eu preciso ter para conseguir me formar?”
 - “Quais matérias eu terei nesse semestre?”
 - “Quantas matérias de cálculo eu terei durante o curso?”
 - “Quais matérias eu terei de eletivas para esse semestre?”
 - “Quantas dp’s eu posso pegar para conseguir passar para o próximo semestre?”
 - “Quais professores irão me dar aula?”
 - “Durante o curso, quantas aulas eu terei?”
 - “Qual horário da minha aula hoje?”
 - “Quais as instalações do campus?”
 - “O campus possui instalação X?”
4. Em uma escala de 0 a 5, o quão importante você considera os seguintes requisitos para as respostas de suas dúvidas?
 - 0 - 5 : Agilidade
 - 0 - 5 : Interação humana
 - 0 - 5 : Interação por voz
 - 0 - 5 : Modo de resposta formal
 - 0 - 5 : Modo de resposta informal

O objetivo final é oferecer o serviço permanentemente para os alunos. Porém, somente durante o desenvolvimento será identificada a real complexidade do projeto e, ao término, a usabilidade do produto final. Desta forma o software será hospedado localmente até a apresentação final. Caso o experimento seja bem-sucedido e viável para aplicação real, considera-se a implementação no *website* da instituição, se tal laurel for concedido aos alunos deste trabalho. As ferramentas mencionadas foram escolhidas de tal forma a propiciar esta transição de forma fluida, como o banco de dados em nuvem e a biblioteca Keras, que permite a geração de modelos em *Javascript* que executam diretamente no navegador.

Referências

- ABDUL-KADER, Sameera A.; WOODS, Dr. John. Survey on Chatbot Design Techniques in Speech Conversation Systems. *International Journal of Advanced Computer Science and Applications*, The Science e Information Organization, v. 6, n. 7, 2015. DOI: [10.14569/IJACSA.2015.060712](https://doi.org/10.14569/IJACSA.2015.060712). Disponível em: <http://dx.doi.org/10.14569/IJACSA.2015.060712>.
- ALLEN, James; CORE, Mark. Draft of DAMSL: Dialog Act Markup in Several Layers. Unpublished manuscript. [S.l.], 1997.
- BRADEŠKO, Luka; MLADENIĆ, Dunja. A Survey of Chatbot Systems through a Loebner Prize Competition. In: v. C, p. 34.
- CALLAWAY, Charles; SIMA'AN, Khalil. Wired for Speech: How Voice Activates and Advances the Human-Computer Relationship. *Computational Linguistics*, v. 32, p. 451–452, set. 2006. DOI: [10.1162/coli.2006.32.3.451](https://doi.org/10.1162/coli.2006.32.3.451).
- CARLETTA, Jean et al. The Reliability of a Dialogue Structure Coding Scheme. *Comput. Linguist.*, MIT Press, Cambridge, MA, USA, v. 23, n. 1, p. 13–31, mar. 1997. ISSN 0891-2017.
- DAHIYA, M. A Tool of Conversation: Chatbot. *JCSE International Journal of Computer Sciences and Engineering*, 2017.
- HUANG, Jizhou; ZHOU, Ming; YANG, Dan. Extracting Chatbot Knowledge from Online Discussion Forums. In: p. 423–428.
- KRAL, Pavel; PAVELKA, Tomas; CERISARA, Christophe. Evaluation of dialogue act recognition approaches. In: 2008 IEEE Workshop on Machine Learning for Signal Processing. [S.l.: s.n.], 2008. P. 492–497. DOI: [10.1109/MLSP.2008.4685529](https://doi.org/10.1109/MLSP.2008.4685529).
- MADHUMITHA. S KEERTHANA. B, Mrs.Hemalatha. B. Interactive Chatbot Using AIML. In.
- MCTEAR, Michael; CALLEJAS, Zoraida; GRIOL, David. *The Conversational Interface*. [S.l.: s.n.], jan. 2016. ISBN 978-3-319-32965-9. DOI: [10.1007/978-3-319-32967-3](https://doi.org/10.1007/978-3-319-32967-3).
- MINSKY, Marvin. *Semantic Information Processing*. [S.l.]: Mass: the MIT Press, 1968.
- NASS, Clifford; BRAVE, Scott. *Wired for Speech: How Voice Activates and Advances the Human-Computer Relationship*. [S.l.]: The MIT Press, 2007. ISBN 0262640651.
- SHRIBERG, Elizabeth et al. The ICSI Meeting Recorder Dialog Act (MRDA) Corpus. In: PROCEEDINGS of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004. Cambridge, Massachusetts, USA: Association for Computational Linguistics, abr. 2004. P. 97–100. Disponível em: <https://www.aclweb.org/anthology/W04-2319>.

- JURAFSKY, D.; SHRIBERG, E.; BIASCA, D. *Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual*. [S.l.], 1997.
- TUR, Gokhan; DE MORI, Renato. Spoken Language Understanding: Systems for Extracting Semantic Information from Speech. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, mar. 2011. DOI: [10.1002/9781119992691](https://doi.org/10.1002/9781119992691).
- TURING, A. M. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, v. LIX, n. 236, p. 433–460, out. 1950. ISSN 0026-4423. DOI: [10.1093/mind/LIX.236.433](https://doi.org/10.1093/mind/LIX.236.433). eprint: <https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf>. Disponível em: <https://doi.org/10.1093/mind/LIX.236.433>.
- WALLACE, Richard. The anatomy of A.L.I.C.E. In: [s.l.: s.n.], jan. 2009. P. 181–210. ISBN 978-1-4020-9624-2. DOI: [10.1007/978-1-4020-6710-5_13](https://doi.org/10.1007/978-1-4020-6710-5_13).
- WEIZENBAUM, Joseph. ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 9, n. 1, p. 36–45, jan. 1966. ISSN 0001-0782. DOI: [10.1145/365153.365168](https://doi.org/10.1145/365153.365168). Disponível em: <https://doi.org/10.1145/365153.365168>.
- YAN, Zhao et al. DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents. In: PROCEEDINGS of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, ago. 2016. P. 516–525. DOI: [10.18653/v1/P16-1049](https://doi.org/10.18653/v1/P16-1049). Disponível em: <https://www.aclweb.org/anthology/P16-1049>.
- YU, Zhou; RUDNICKY, Alexander; BLACK, Alan. Learning Conversational Systems that Interleave Task and Non-Task Content. In: p. 4214–4220. DOI: [10.24963/ijcai.2017/589](https://doi.org/10.24963/ijcai.2017/589).