Interação Humano-Robo: Robô Recepcionista

Bolsista Marcos Vinicius Cruz (CTI) mvcruz@cti.gov.br

Resumo

O objetivo geral associado ao projeto proposto é contribuir para a pesquisa na área de Interação Humano Robô que é uma área essencialmente multidisciplinar envolvendo conhecimentos das ciências exatas, biológicas e humanas através do projeto da DRVC associado à área de IHR que é o projeto "Informações Não-Verbais na Interação Robô Aplicado a um Robô Recepcionista". Adicionalmente considera como objetivo geral produzir conhecimento teórico e prático, realizar provas de conceito, experimentos de usabilidade e pesquisar técnicas associadas à área de interação humano robô usando como base a plataforma do robô recepcionista.

Palavras-chave: Interação Humano Robô, Agente virtual, Recepcionista.

1. Introdução

O objetivo da pesquisa em IHR (Interação Humano Robô) é definir modelos de comportamento dos seres humanos em relação à interação com o robô para orientar o projeto deste e o desenvolvimento algorítmico que permitiria uma interação mais natural e efetiva entre humanos e robôs. A pesquisa abrange desde como os seres humanos trabalham com veículos não tripulados, até a colaboração entre pares com robôs antropomórficos.

Na indústria, a integração de humanos com robôs está crescendo, como pode servisto na preocupação de empresas de automação industrial, como em (KNIGHT, 2015). Outro exemplo de potencial uso de robos em robótica pessoal onde, no futuro, os robôs poderão ser utilizados para assistir pessoas, fisicamente e mentalmente. Observe que a robótica pessoal é uma necessidade premente, principalmente, considerando o aumento na proporção da população idosa, sendo que uma tecnologia fundamental para o desenvolvimento dessas aplicações é o desenvolvimento da Interação Humano Robô.

Neste trabalho será apresentado as melhorias realizadas no arcabouço do robô recepcionista ao longo do período de 12 meses e algumas funcionalidades adicionadas ao robô.

2. Resumo de Atividades Desenvolvidas

Durante o período o bolsista realizou uma série de melhorias no projeto do robô recepcionista, dentre elas:

a. **Uso de alfabeto fonético:** O robô recepcionista tinha problemas ao reconhecer nomes próprios estrangeiros, então foi proposta a solução de avaliar o nome foneticamente para achar alguma proximidade na base de dados. Nomes como "Balashov" e "Panepucci" que não eram encontrados agora são apresentados como possíveis saídas do sistema.

b. **Criação de estados latentes:** Existiam problemas no gerenciamento das rotinas do robô como: cumprimentar; responder; se despedir. Algumas vezes saudações eram dadas inúmeras vezes, então foi criado um modelo temporal de estados que funciona como um ciclo restrito. Onde cada ação e realizada apanas caso uma anterior tenha ocorrido ou se existir a probabilidade de uma pessoa estar presente na frente da recepcionista para a tomada de ação.

c. **Revisão de funções no projeto:** Após avaliação do módulo de gerenciamneto de resposta do robô recepcionista, foram encontrados trechos de códigos que geravam alguns bugs, e os mesmos foram corrigidos apenas reformulando as operações.

d. **Arvore de decisões para respostas curtas:** Algumas respostas como: “O que é o CTI”; “Quem sou eu?”; “O que é você?”; entre outras, são perguntas feitas com uma alta frequência. Então elas foram classificadas como “hotQuestions” cujas as perguntas deviam ter respostas imediatas. Antes as respostas foram implementadas por vários *if/else* o qual não cobria todas as perguntas e muitas vezes não respondia. Atualmente foi desenvolvida uma árvore de decisão onde é feita uma busca em todos os nó em ordem de achar a melhor resposta e ela é facilmente atualizável.

e. **Chamadas de sistema:** Foi desenvolvida um módulo para enviar mensagens diretamente ao SO para administrar o microfone, ligando e desligando o mesmo. Pois o robô recepcionista, ocasionalmente, ativava o mecanismo de fala quando não tinha alguém interagindo.

Após a realização desses ajustes foram conduzidos experimentos indoor. O primeiro foi realizado com colaboradores do CTI para verificar a aceitabilidade do robô recepcionista; o segundo monitorado para verificar as reações das pessoas e como elas interagiam com o agente.

Para melhorar a captação de áudio do robô recepcionista foi desenvolvido um mixer com suporte a múltiplos microfones, com intuito de entender o que o usuário está dizendo

O módulo de detecção facial foi substituído por um módulo de detecção mais robusto comparado ao que era usado anteriormente, porém consome mais recursos da máquina. Porém já foi adaptado para uma arquitetura cliente-servidor onde toda a parte de processamento é realizado em outra máquina. Além de ter sido integrado detecção de máscara no módulo de visão computacional**.**

Um sistema de monitoramento foi adicionado ao robô recepcionista que gera registro de suas interações e estados internos do sistema, para que seja avaliado caso ocorra algum erro durante seu funcionamento. Além disso, está sendo utilizado um programa para verificar o uso de recursos do computador onde o robô recepcionista se encontra.

3. Modularização do robô recepcionista

A versão atual do robô recepcionista foi desenvolvida para trabalhar em um sistema local, onde todos os seus sistemas são executados em uma única máquina. Então conforme o sistema é expandido com novas funcionalidades há maior consumo de memória e recursos de processos do computador, isso acaba gerando gargalos no escalonamento de processos pois existe um maior custo na execução das tarefas.

Esse problema ocorre em escalonamentos verticais, onde você precisa adicionar mais poder (CPU, RAM) para uma máquina existente. Visto isso propusemos uma nova arquitetura para

com escalonamento horizontal para melhorar a performance do robô recepcionista. Utilizando uma arquitetura distribuída podemos aumentar a pool de recursos para vários sistemas independentes, assim reduzindo os custos de processamento das máquinas e garantir que caso um dos serviços pare o sistema não irar gerar falha.

Para isso foi necessário algumas adequações no Sistema de gerenciamento de respostas verbais e não verbais, sendo estas descritas abaixo:

a. **Redução do escopo de respostas:** Algumas das funções do robô estavam gerando muito erros durante a migração. Então estas foram removidas e sendo adicionadas periodicamente durante o desenvolvimento, como a ligação para ramais.

b. **Reformulação na interpretação de sentenças:** A busca e a comparação das sentenças estavam com um grande tempo de execução, requisições ao banco de dados sem otimização e a comparação usando falhando em alguns casos. Então as mesmas foram reformuladas.

c. **Adição de uma análise fonética:** Existia um problema nas respostas quando era requisitado informações de pessoas com nome estrangeiro. Pois o sistema de voz não consegue interpretar essas palavras devido a uma limitação da sua plataforma que apenas consegui processas palavras em uma única língua. Então quando é retornado um erro na análise de nomes próprios é feito uma análise fonética nesse nome.

d. **Adição de um sistema de linguagem natural:** Para melhorar a análise de sentenças foi adicionado uma rede neural baseada em linguagem natural para as interpretações de sentenças. A ideia principal é implementar todas as regras de forma que elas sejam interpretadas pela rede neural. Atualmente a rede neural analisa as frases que são frequentemente usados em um cotidiano de uma recepcionista e retorna uma resposta com a maior probabilidade de ser dita naquele cenário.

A figura abaixo reflete as diferenças da arquitetura nova para antiga, sendo à esquerda a arquitetura nova e à direita a arquitetura antiga:

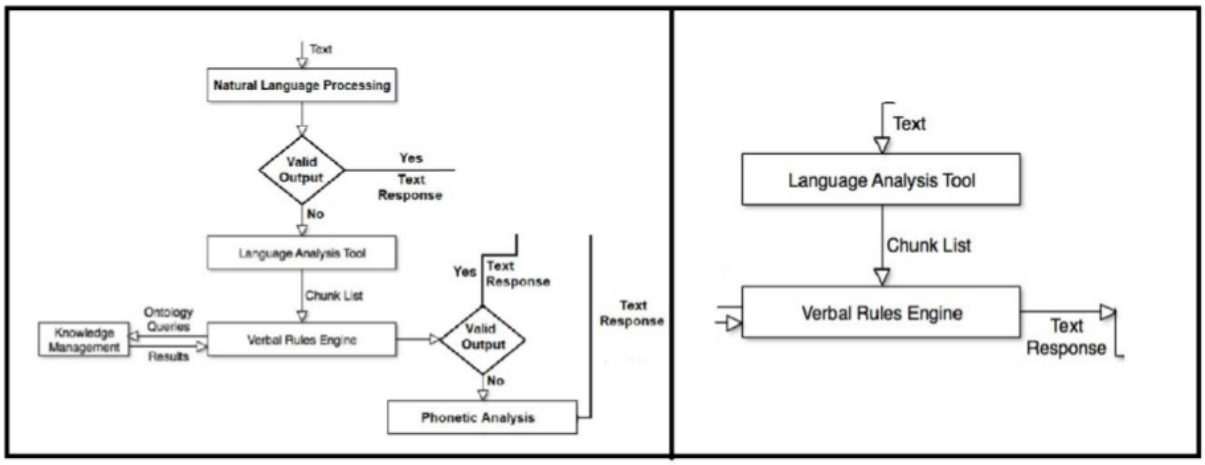

Figura 1 – Comparação entre a arquitetura antiga e a atual

Alguns elementos do sistema foram mantidos, pois são estruturas fundamentais para análise de respostas. Como o interpretador gramatical, que classifica os elementos sentenças, e a base de dados do robô, que serve como conhecimento à ser inserido em suas respostas.

4. Sistema para Detecção de Pessoas que opera a 5 metros

Anteriormente o sistema de detecção de pessoas era realizado por um programa que utiliza o algoritmo HaarCascade para a detecção de pessoas. Porém atualmente existem modelos para redes neurais mais acurados que o algoritmo HaarCascade. Então houve a motivação de investigar se a implementação de uma rede neural para a detecção de pessoas apresentaria melhores resultados.

Inspirado em (KOMAR, 2018), foi utilizado o módulo CAFFE que é disponibilizado na biblioteca de visão computacional OPENCV. A opção por utilizar esse módulo é porque ele tem uma boa acurácia tanto para pessoas próximas, quanto para pessoas que estão distantes da câmera (sendo o mesmo treinado por uma robusta base de dados).

```
def __init__(self):
    print("[INFO] loading model...")
    self.net = cv2.dnn.readNetFromCaffe(abs_path_proto, abs_path_model)
    super(Detection, self)
```

Figura 2 – Inicialização do módulo CAFFE via código

Quando é utilizado o módulo de rede neural do OPENCV com o modelo CAFEE é necessário setar 2 arquivos:

- Um arquivo *.prototxt* que define o modelo da arquitetura da rede neural (número decamadas, função de ativação, etc)
- Um arquivo *.cafeemodel* que contém os pesos para a camada atual.

O detector facial de deep learning do OpenCV é baseado na estrutura Single Shot Detector (SSD) com uma rede base ResNet (ao contrário de outros SSDs OpenCV que normalmente usam MobileNet como rede base). Sendo esse o detector mais acurado dentro da biblioteca (KOMAR, 2018).

O sistema de detecção baseado em deep learning possui uma alta precisão porém, em contrapartida, o custo de processamento é muito alto. No caso do robô recepcionista onde todos os serviços rodam local (na própria máquina) ele concorrem por processamento e uso de memória na máquina local. Isso afeta diretamente o desempenho da rede neural.

Para resolver esse problema. A solução foi implementar um sistema de detecção de pessoas como um serviço em nuvem onde o servidor é modularizado baseado em um conjunto de microprocessos, onde cada um exerce somente uma única função. Após o processamento da imagem, o servidor retorna para o cliente uma flag que indica se uma pessoa foi detectada, sendo que no caso da perda de um frame não ocorrerá interferência no processamento dos frames subsequentes.

Após a realização dos testes, notou-se uma melhora no funcionamento do robô recepcionista e diminuiu a latência de resposta da mesma ao exercer suas tarefas. A única limitação dessa implementação é a necessidade do robô recepcionista estar conectada na rede para enviar as informações ao servidor de imagens. Existe um sistema de detecção de pessoas que roda local caso haja algum erro na comunicação entre o robô e o servidor.

5. Sistema para Detecção de Nomes Estrangeiros Baseados em Fonemas

A implementação de análise fonética no robô recepcionista foi implementado para suprir a necessidade de interpretação de nomes estrangeiros. Pois muitos sobrenomes na base de conhecimento do robô não estão na nossa língua nativa. Então o módulo de conversão de fala para textos gera erros ao interpretar as palavras que não são da língua portuguesa.

Então foi realizada uma adaptação no trabalho proposto por (SHAN, 2014) utilizando o algoritmo Soundex para realizar a conversão dessas palavras. O objetivo principal deste algoritmo é converter a palavra em um código, com base em várias regras, sendo assim, a comparação entre 2 palavras é feita pelo seu código Soundex. Seu funcionamento é descrito abaixo utilizando um estudo de caso.

A palavra WATCHER, por exemplo, possui o código Soundex W-326 e a palavra WUATCHER possui também o mesmo código Soundex, consequentemente, quando a comparação for feita a palavra será foneticamente a mesma, pois W-326 = W-326.

Para realizar a conversão, é necessário conhecer as regras para converter cada caractere para um código Soundex respectivo. Logo abaixo apresentaremos essas regras:

- **Regra 1:** Todo código Soundex possui obrigatoriamente 1 letra e 3 números, onde a primeira letra é sempre a primeira letra da palavra. Perceba a palavra WATCHER como o código gerado foi W-326 (primeira letra = W).
- **Regra 2:** A tabela abaixo representa qual número cada caractere deve ser substituído. Por exemplo, significa que se você encontrar o caractere “D” na palavra, substituía este pelo número “3”.

TABELA DE CODIFICAÇÃO – CONVERSÃO PARA FORMATO CANÔNICO

Números	Letras que representam
1	B, F, P, V
2	C, G, J, K, Q, S, X, Z
3	D, T
4	L
5	M, N
6	R

Tabela 1 – Codificação da palavra para a estrutura do Soundex

- **Regra 3:** Alguns caracteres que não serão encontrados na tabela acima, isso porque eles devem ser simplesmente descartados pois não influenciam na fonética da palavra. São eles: A, E, I, O, U, H, W, e Y.
- **Regra 4:** Caso a palavra possua alguma letra “dupla”, tal como: GG, RR, ZZ e etc, você deverá considerar apenas como 1 letra. Ex: Em vez de atribuir para o RR = 66, nós faríamos RR = 6.

- **Regra 5:** Caso a palavra possua duas letras diferentes, uma ao lado da outra, com o mesmo código Soundex, você deverá considerar apenas como 1 código. Ex: P e F possuem o mesmo código Soundex, sendo assim, em vez de fazermos PF = 11, faríamos PF = 1. Outro exemplo: em vez de CK = 22, faríamos CK = 2.
- **Regra 6:** Caso a palavra possua um prefixo tal como: Van, Con, De, Di, La ou Le, você pode codificar tanto seu prefixo como a palavra sem o prefixo e usar um ou outro código.

Por exemplo: a palavra VanDeusen pode ser V-532 (Van) ou D-250 (Deusen). Outra coisa importante é que prefixos como Mc e Mac não valem para essa regra.

- **Regra 7:** Se uma vogal (A,E,I,O,U) separar duas consoantes que possuem o mesmo código Soundex, somente a consoante à direita da vogal é codificada, consequentemente, a consoante à esquerda não é codificada.
- **Regra 8:** Se o caractere "H" ou "W" separar 2 consoantes que possuem o mesmo código Soundex, então apenas a consoante do lado direito deste caractere será codificado.
- **Regra 9:** De acordo com a Regra 1, todo código deve ter 4 caracteres (1 letra + 3 números). Ocorreque algumas vezes, depois de testar todas as regras acima, o tamanho do código gerado é menor que 4, para este caso em específico adicionamos uma quantidade de números "0" ao fim do código até completar 4 caracteres.

6. Conclusão

Mesmo devido à série de aprimoramentos realizados no robô recepcionista, é notável que existe um extenso trabalho à ser feito para torná-la funcional a longo termo e serão preciso mais testes de interação com usuários para validação de futuras funcionalidades.

Como trabalho futuro será adicionado ao sistema de visão computacional o reconhecimento de pessoas (já implementado porém é necessário realizar a integração com o sistema e validação) e emoções. Além de aprimorar a arquitetura do robô recepcionista para uma arquitetura híbrida baseada em comportamentos como proposto em (FERLAND, 2017).

Referências

CHOWDHARY, K. R. *Natural language processing. In: Fundamentals of Artificial Intelligence. Springer, New Delhi*, p. 603-649, 2020.

FERLAND, François et al. *Coordination mechanism for integrated design of Human-Robot Interaction scenarios.* Paladyn, Journal of Behavioral Robotics, v. 8, n. 1, p. 100-111, 2017.

KNIGHT W. *SMART ROBOTS Can Now Work Right Next to Auto Workers.* MIT Technology Review, 2013.

KOMAR, Myroslav et al. *Deep neural network for image recognition based on the caffe framework.* In: 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP). IEEE, p. 102-106, 2018.

SHAH, Rima. *Improvement of Soundex algorithm for Indian language based on phonetic matching.* International Journal of Computer Science, Engineering and Applications, v. 4, n. 3, p. 31, 2014.