Reconhecimento de Fala para Robô Humanoide com o Auxílio de APIs

Trabalho em progresso

Luiza Amador Pozzobon[1], Guilherme da Silva Garcia[2], Anthony de Almeida[3], Carolina Rutili de Lima[4], Thais Marisco Brollo[1], Rodrigo Guerra[5], Giovani Rubert Librelotto[2]

[1]Curso de Engenharia de Controle e Automação
Universidade Federal de Santa Maria (UFSM) Santa Maria – RS – Brasil

[2]Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria (UFSM) – Santa Maria – RS – Brasil

[3]Curso de Sistemas de Informação
Universidade Federal de Santa Maria (UFSM) Santa Maria – RS – Brasil

[4]Programa de Pós-Graduação em Engenharia de Automação e Sistemas
Universidade Federal de Santa Catarina (UFSC) Florianópolis – SC – Brasil

[5]Departamento de Processamento de Energia Elétrica
Universidade Federal de Santa Maria (UFSM) Santa Maria – RS – Brasil

{luiza.pozzobon, guilesgarcia, carolinarutililima, tbrollo, tioguerra}@gmail.com, {aalmeida, librelotto}@inf.ufsm.br

Abstract. *Humanoid robots are robots that imitate the human appearance and actions, implying a greater empathy. An essential element to the human-robot relation is the speech recognition and interpretation. Evidencing that there are no solutions in Brazilian Portuguese for this purpose, we developed a system capable to recognize and interpret the speech using APIs. This system is capable of performing the transcription of the audio to text and extract the intent and information of phrases. The present article shows the system development and reports its application in a humanoid robot.*

Resumo. *Os robôs humanoides são robôs que imitam a aparência e ações dos humanos, gerando assim uma maior empatia. Um elemento essencial para a relação humano-robô é o reconhecimento e interpretação de fala. Evidenciando que não há soluções em português do Brasil para este fim, desenvolvemos um sistema capaz de reconhecer e interpretar a fala utilizando APIs. Este sistema é capaz de realizar a transcrição do áudio em texto e extrair a intenção e informações de frases. O presente artigo apresenta o desenvolvimento do sistema e relata a aplicação do mesmo em um robô humanoide.*

1. Introdução

Atualmente os robôs são utilizados para diferentes fins, como cuidado às pessoas idosas e crianças com autismo, educação e terapia [Broadbent, 2017]. Um dos tipos de robôs são os robôs humanoides, os quais são geralmente definidos como máquinas programáveis que imitam as ações e a aparência dos humanos [Graefe and Bischoff, 2003]. Estes robôs

são equipados com sensores capazes de perceber o ambiente para executar ações. Podem expressar emoções através dos olhos e bocas e executar diferentes tarefas como os humanos usando mãos e pernas [Ting et al, 2014].

O robô ser capaz de reconhecer e interpretar o que é falado é um elemento essencial para a interação entre robôs humanóides e os humanos. O processamento de linguagem natural é uma área que estuda esse reconhecimento e interpretação, na qual é definido como técnicas computacionais motivadas por teorias para a análise e a representação automáticas da linguagem humana [Cambria, White, 2014].

Os *softwares* de reconhecimento de fala hoje atingem taxas de acerto muito próximas das dos próprios humanos [9to5google, 2017], segundo Sundar Pichai, CEO da *Google*, e são cerca de três vezes mais rápidos que a digitação [Ruan et al., 2016]. Esses avanços foram, em grande parte, obtidos pela evolução de inteligência artificial e *softwares* de predição, que se beneficiam de treinamentos em grandes bancos de dados e técnicas de *deep learning* [Hinton et al., 2012]. Estes softwares são aplicados cotidianamente no controle de utensílios domésticos, de *smartphones* e de eletrônicos através de 'ajudantes', como Siri [Gruber, 2009] e Cortana [Canbek, 2016].

Ao utilizarmos ferramentas para o reconhecimento de fala, como o *Google Cloud Speech* [Google Cloud Speech API, 2017], obtemos o texto do que foi dito por uma pessoa. Neste contexto, a interpretação do que foi falado é de suma importância para a relação humano-robô. Através de uma pesquisa na literatura e ferramentas disponíveis no mercado, não encontramos nenhuma ferramenta disponível atualmente que faça o reconhecimento e interpretação da fala em português do Brasil. Por esse motivo, desenvolvemos um sistema que une o reconhecimento de fala e a interpretação da mesma. Para isso, em conjunto ao *Google Cloud Speech*, utilizamos o Wit.ai, aplicação que identifica a intenção da fala e extrai segmentos importantes das frases.

O presente artigo apresenta o desenvolvimento deste sistema e o relato de sua aplicação em um robô humanoide, desenvolvido pela Qiron Robotics, empresa especializada em robótica. O artigo é divido da seguinte forma: a seção 2 apresenta os materiais e métodos utilizados no projeto. A seção 3 apresenta os resultados e as conclusões, e os trabalhos futuros são apresentados na seção 4.

2. Materiais e Métodos

O sistema foi desenvolvido na linguagem de programação Python, seguindo o paradigma de orientação a objetos, prática de criação que possibilita flexibilidade através de um design modular [Smith, 2015]. O código foi dividido em classes, conjunto de objetos e ferramenta de programação que associa atributos e métodos numa só estrutura. Há classes responsáveis pelo controle das principais funcionalidades do projeto: o Google Cloud Speech, o Wit.ai e a resposta do robô, estas são descritas na subseção 2.3.

2.1. Interfaces de programação de aplicações (APIs) utilizadas

Interface de Programação de Aplicações (API) é um *software* intermediário que permite que duas aplicações se comuniquem. Essa comunicação consiste em enviar dados para um servidor, que os interpreta, realiza as ações necessárias e envia as informações requisitadas de volta para quem as requisitou [Stowe, 2015]. Nosso sistema utiliza duas APIs, ambas descritas abaixo, uma para a transcrição do áudio em texto e outra para a interpretação das frases de tal texto.

2.1.1. Google Cloud Speech

A API *Google Cloud Speech* [Google Cloud Speech API, 2017] é responsável pela transcrição do áudio em texto. Essa API é capaz de converter áudio em texto em cerca de 80 idiomas e está em contínua expansão de funcionalidades. Utiliza o conceito de reconhecimento de fala automático, além de possuir suporte à áudios pré-gravados ou em tempo real e lidar com possíveis ruídos e barulhos inadequados.

2.1.2. Wit.ai

A segunda API é a *Wit.ai* [Wit.ai API, 2013], uma plataforma para linguagem natural, que possibilita que desenvolvedores criem *softwares* capazes de comunicação com os humanos, e a transformação da palavra em ação. Esta API funciona com os conceitos de *stories*, *intents* e *entities* (histórias, intenções e entidades).

O exemplo apresentado na Figura 1 mostra o funcionamento destes três elementos. *Story* é o cenário a ser programado, a requisição de informações ou de ações do robô. Neste exemplo, é o pedido da hora atual em um local determinado. *Entities* são fragmentos do texto que carregam significados importantes, como a da localização: "Santa Maria". Por fim, as *intents* apresentam as intenções ou funções da história em curso, os seus objetivos.

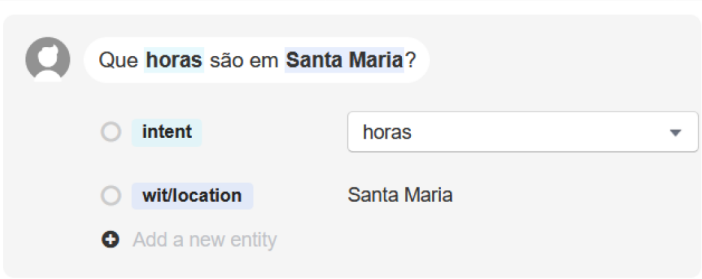

Figura 1. Exemplo de criação de uma *story* no wit.ai.

2.2. Organização do código

O sistema é composto por seis classes, conforme é mostrado na Figura 2. As classes responsáveis pela conexão com a API do Google são duas: *SpeechRec* e *SpeechRecognize.* A *SpeechRec* é responsável por capturar o áudio no microfone e enviá-lo ao Google para receber a transcrição do texto. E a classe *SpeechRecognize* executa a classe *SpeechRec* e recebe a transcrição do texto.

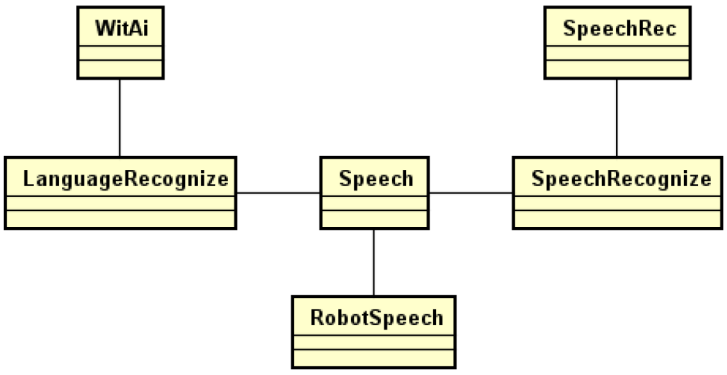

Figura 2. Diagrama de classes do software desenvolvido.

As duas classes que fazem a conexão com a API do wit.ai são: *WitAi* e *LanguageRecognize*. A primeira envia a frase transcrita em texto e recebe o arquivo JSON (Notação de Objetos JavaScript) com as interpretações geradas pelo Wit.ai no processamento da linguagem natural. Já a *LanguageRecognize* processa o JSON gerado pela classe WitAi e retorna a intenção da frase em texto.

A classe responsável pela fala do robô é a *RobotSpeech*. Nessa classe estão as respostas do robô de acordo com a intenção da fala capturada anteriormente, tanto em texto para exibição na tela, quanto em comandos de áudio (fala do robô). Por exemplo, a intenção 'horas' possui frases de resposta, com ou sem comandos adicionais, pré-definidas escritas pelos desenvolvedores. Por fim, existe a classe *Speech*, responsável por instanciar as classes *RobotSpeech*, *LanguageRecognize* e *SpeechRecognize* e controlar a execução do sistema. Nela estão funções que gerenciam o estado do mesmo: ligado ou desligado.

O fluxo de dados do sistema pode ser visualizado na Figura 3. Após ser iniciado, o sistema aguarda o recebimento do áudio, através do microfone, para transcrição. O texto gerado é impresso na tela e enviado para o Wit.ai para obtenção da intenção e entidades. Por fim, a resposta do robô é recuperada de acordo com os dados obtidos no cenário em questão.

A porção do código responsável pela geração do áudio foi desenvolvida pela empresa *Qiron Robotics,* também na linguagem de programação Python. A resposta do robô, em texto, é enviada como parâmetro para geração do áudio com a biblioteca *gTTS (Google Text to Speech)*, que além de criar o arquivo, ajusta a frequência do áudio para a voz característica do robô. O arquivo gerado, então, é executado e transmitido pela caixa de som presente no corpo do mesmo.

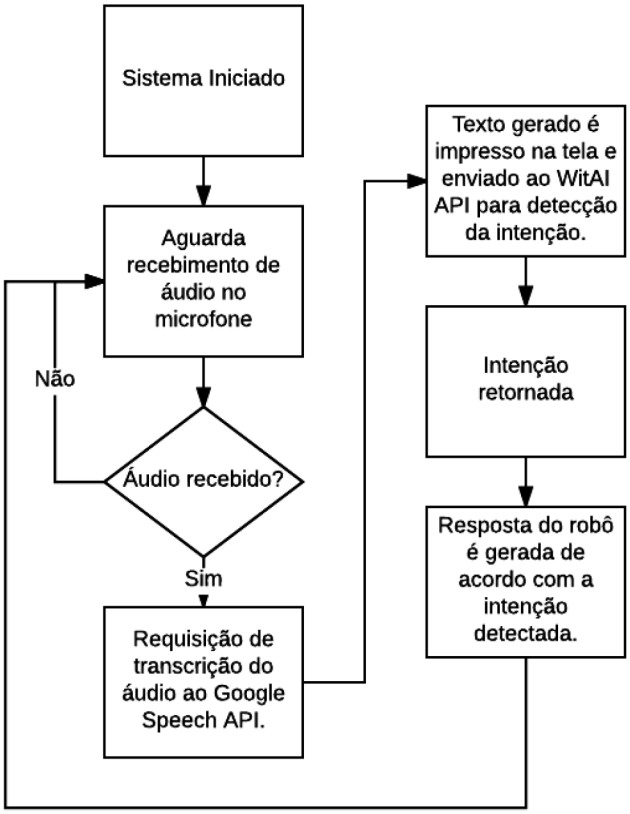

Figura 3. Fluxo de dados no software.

3. Resultados

O sistema foi aplicado no robô humanoide Beo, desenvolvido pela empresa especializada em robótica: *Qiron Robotics*. Este robô possui 42cm de altura e é equipado com sensores ultrassônico e de toque, motores *Dynamixel* e módulos de microfone e câmera [Qiron Robotics, 2016].

Figura 4. Robô humanoide Beo da Qiron Robotics.

Os cenários para reconhecimento de fala em fase de implementação no código podem ser observados na Tabela 1. Obteve-se sucesso para todas as frases apresentadas na tabela. O teste foi realizado da seguinte maneira: uma pessoa se aproxima em um metro do robô e fala uma frase, por exemplo, “Que horas são?”. O robô responde outra frase, por exemplo, “Agora são 2 horas e 19 minutos”, esta resposta também é exibida no terminal de execução, conforme mostra a Figura 5.

Intenção	Frase dita pelo usuário	Resposta do robô
Horas	Que horas são?	Agora são X horas e Y minutos
Dia	Que dia é hoje?	Hoje é dia X do mês Y do ano de Z.
Cumprimentar	Olá! Qual seu nome?	Olá, meu nome é Beo!
Despedida	Tchau, até mais!	Tchau! Até mais.

Tabela 1. Lista de cenários em fase de implementação e testes.

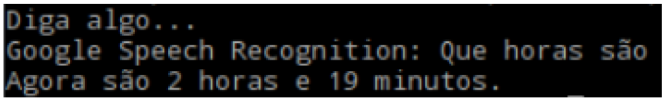

Figura 5. Execução do software no terminal de comando.

4. Conclusões e Trabalhos Futuros

Neste artigo foi apresentado o desenvolvimento de um sistema de reconhecimento e interpretação de fala através da utilização de *API* e os resultados de sua aplicação em um robô humanoide. Este sistema será utilizado como uma das funcionalidades do robô Beo, que passa a ter o reconhecimento e interpretação de fala com um dos seus recursos. Assim possibilitando uma interação através da fala entre humanos e este robô.

Como trabalho futuro, espera-se acrescentar funções de detecção de intenções e entidades no *Wit.ai*, a fim de torná-lo um *software* robusto e com ‘personalidade’. A criação de *stories* e respostas do robô são feitas manualmente e requerem tempo para serem desenvolvidas, portanto serão acrescentadas ao longo dos próximos meses.

Referências

BROADBENT, Elizabeth. Interactions With Robots: The Truths We Reveal About Ourselves. Annual review of psychology, v. 68, p. 627-652, 2017.

Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language processing research. IEEE Computational intelligence magazine, 9(2), 48-57.

Canbek, N. G., & Mutlu, M. E. (2016). On the track of Artificial Intelligence: Learning with intelligent personal assistants. Journal of Human Sciences, 13(1), 592-601.

Google Cloud Speech API. Disponível em: <https://cloud.google.com/speech/>. Acesso em: 12 ago. 2017.

Google’s speech recognition is now almost as accurate as humans. Disponível em: <https://9to5google.com/2017/06/01/google-speech-recognition-humans/>. Acesso em: 12 ago. 2017.

GRAEFE, Volker; BISCHOFF, Rainer. Past, present and future of intelligent robots. In: Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International Symposium on. IEEE, 2003. p. 801-810.

Gruber, T. R. (2009). Siri, a Virtual Personal Assistant—Bringing Intelligence to the Interface.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82-97.

Qiron Robotics. Disponível em: <http://webpage.qironrobotics.com/>. Acesso em: 13 ago. 2017.

Ruan, S., Wobbrock, J. O., Liou, K., Ng, A., & Landay, J. (2016). Speech Is 3x Faster than Typing for English and Mandarin Text Entry on Mobile Devices. *arXiv preprint arXiv:1608.07323*.

Smith, B. (2015). Object-oriented programming. In *Advanced ActionScript 3*. Apress, p. 1-3.

Speech Recognition 3.7.1. Disponível em: <https://pypi.python.org/>. Acesso em: 13 ago. 2017.

Stowe, M. (2015). *Undisturbed REST: A guide to designing the perfect API*. Lulu. com. p. 1-3.

TING, Chen-Hunt et al. Humanoid robot: A review of the architecture, applications and future trend. Res. J. Appl. Sci. Eng. Technol, v. 7, p. 1364-1369, 2014.

Wit.ai API (2013). Disponível em: <https://wit.ai/>. Acesso em: 12 ago. 2017.