

CHATBOTS



WORKSHOP DE DESENVOLVIMENTO DE CHATBOTS

Prof. Dr. Saulo Popov Zambiasi

<http://saulopz.blogspot.com.br>

saulopz@gmail.com

Sumário

- Apresentação
- Definição
- Histórico
- Arisa
- O Teste de Turing
- A Sala Chinesa
- Exemplos e Aplicações
- AIML e Interpretadores
- Exemplo de implementação simplificada
- Desenvolvimento de Projeto

Apresentação

- Graduação: Ciência da Computação (UNOESC-Chapecó).
- Especialização: Ciência da Computação (UFSC).
- Mestrado: Ciência da Computação (UFSC) – Dissertação em Sistemas Multiagentes.
- Doutorado: Engenharia de Automação e Sistemas (UFSC) – Tese em Softwares Assistentes Pessoais
- Atual professor na UNISUL
- Grupos e Projetos de Pesquisa:
 - Subverse: Grupo de Pesquisa em Ciberarte
<http://subverseproject.blogspot.com.br/>
 - ARISA: Projeto de Pesquisa em Softwares Assistentes Pessoais.
<http://www.projetoarisa.com.br/>

Definição

- Programa de computador que simula a conversa em linguagem natural tal como se fosse uma pessoa.
- Tenta responder perguntas dando a impressão de que se está conversando com outra pessoa.
- Chatbot ou Chatterbot: *CHATTER* (pessoa que conversa) e *BOT* (robô de software ou *softbot*).
- A palavra foi inventada por Michael Mauldin (1994).
- Funcionamento: consulta uma base de conhecimento para tentar fornecer uma resposta condizente com a pergunta.

Histórico

- ELIZA – primeiro chatbot, desenvolvido por Joseph Weizenbaum (1996) no MIT.
- <http://www-ai.ijs.si/eliza/eliza.html>
- Por meio de uma sentença digitada pelo usuário, o chatbot identificava uma palavra chave importante ou um contexto mínimo.
- O processamento se dá pela manipulação de palavras e frases de modo a responder de forma “inteligente” as perguntas dos usuários.
- Muitos clones do ELIZA surgiram depois.

Histórico

- ALICE (*Artificial Linguistic Internet Computer Entity*) – evolução do ELIZA.
- Estrutura própria de codificação de premissas baseada em XML (AIML).
- AIML (*Artificial Intelligence Markup Language*) – padrão atual, baseado em contextos (licença GNU GPL).
- Base para que muitos *chatbots*.
- Cybele – Um dos primeiros *chatbots* brasileiros (1999) de Alex Primo.

ARISA

- **ARISA** – Possui um módulo de *chatbot* (serviço web) para interação com o usuário.
- Em funcionamento desde set de 2010 e integrado à ARISA em janeiro de 2011.
- **A.R.I.S.A.** *acrônimo para Assistant Representative: an Instance using Services Architecture.*
- Software Assistente Pessoal (SAP) baseado na Arquitetura Orientada a Serviços (SOA).
- Base de conhecimento gerenciada por banco de dados.
- Website: <http://www.projetoarisa.com.br>

ARISA – Projetos Atuais

- UFSC: Chatbot de auxílio aos alunos e colaboradores.
- P.E.M. (Persona Ex Machina): Robô interativo para espetáculos teatrais (ciberarte):
 - **Autonomia:** age por si, não controlada por humanos;
 - **Percepção:** percebe seu ambiente via sensores;
 - **Ação:** age em seu ambiente;
 - **Interação:** interage com os demais atores;
 - **Roteiro:** segue roteiro estipulado para o espetáculo teatral;
 - **Improvisação:** pode improvisar, se necessário;
 - **Interpretação:** pode seguir as mudanças de humor de seu personagem conforme o roteiro.

ARISA – Publicações Específicas

- Diálogos performáticos interativos para atores virtuais (JLAET'2013).
- Una propuesta de los principios de la performatividad de actores autónomos digitales con inteligencia artificial (III Congreso Internacional y V Congreso Nacional de Teatro'2014).
- Persona ex machina: o teatro dialogando com a inteligência artificial (JLAET'2014).
- O autômato e a inteligência artificial no teatro de animação: quem é o manipulador? (FITA'2015).
- O ator dialogando com a máquina: novos conflitos (JLAET'2015).

O Teste de Turing

- Alan Turing foi talvez o primeiro a estudar a IA!
- Computador é interrogado por um Humano!
- Será que o consegue enganar?
 - Sim? Então é Inteligente!
- Sem Interação Física (via teclado+monitor)
- Capacidades do Computador:
 - Processamento de Linguagem Natural, Representação do Conhecimento, Raciocínio Automático, Aprendizagem
- Teste Total de Turing: Físico!
 - Necessária Visão por Computador e Robótica
 - Ainda não foi conseguido?

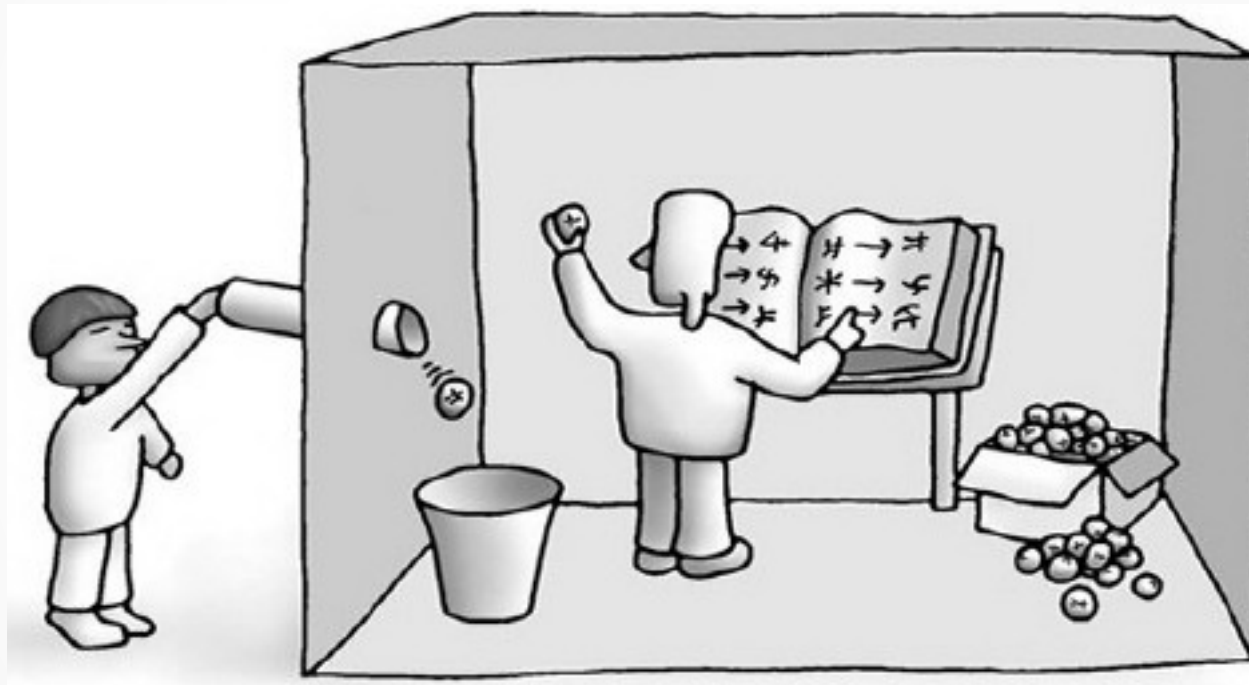
O Teste de Turing



A Sala Chinesa

- Procura rebater a validade do Teste de Turing;
- Questões:
 - O operador da sala entende chinês?
 - Os manuais entendem chinês?
 - O conjunto da sala é que entende chinês?
- Mostra que o Teste de Turing não prova de forma suficiente um sistema como sendo inteligente.

A Sala Chinesa



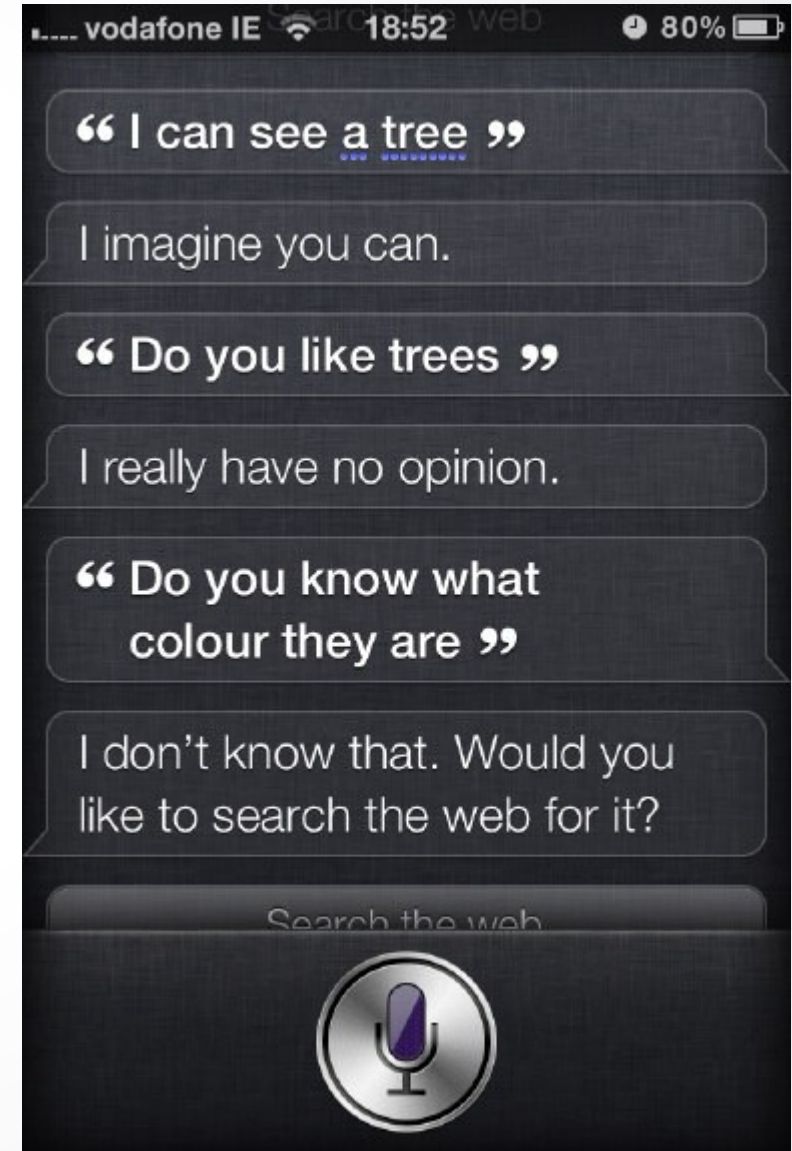
Exemplos e Aplicações

- **Robô Ed**
- Petrobras – Conpet: programa nacional de racionalização do uso dos derivados do petróleo e do gás natural.
- Chatbot para informações sobre preservação de energia e de outros recursos naturais.
- <http://www.ed.conpet.gov.br/br/converse.php>



Exemplos e Aplicações

- **SIRI**
- Assistente pessoal para iOS.
- Processamento de linguagem natural para responder perguntas, fazer recomendações, e executar ações.
- Adquirido pela Apple em de abril de 2010.



Exemplos e Aplicações



- **Servicebot-3:**
- Chatbot para virtualização de atendimento.
- Objetivos:
 - promover melhoria no nível de satisfação dos usuários;
 - aumento da acessibilidade dos serviços de atendimento;
 - padronização na prestação de serviços;
 - Desenvolver dialogos para poder realizar consultas, atualizar, validar e gravar dados em sistemas externos, executar transações e tomar decisões.
- <http://www.servicebot.com.br/index.php/demonstracao/>

Exemplos e Aplicações

- **GAL:** Assistente Virtual da Gol
 - <http://voegol.com.br>
- Desenvolvido pela Bluelab
 - <http://www.bluelab.com.br/>
- Agente que fornece auxílio no website da Gol tal como um assistente humano.



Exemplos e Aplicações

- Chatdip – Chatbot do Departamento de Integração Acadêmica e Profissional (UFSC)
- Interface web
- Atualmente em desenvolvimento.
- <http://gsigma.ufsc.br/chatdip/>



Exemplos e Aplicações

- A.R.I.S.A.
- Chat via Google Hangouts:
personal.assistant.arisa@gmail.com
- Chat via Twitter: https://twitter.com/arisa_ap
- Chat via SMS/Celular



AIML e Interpretadores

- O AIML (GNU GPL) é uma linguagem baseada em XML para chatbots desenvolvida em 1995 para o A.L.I.C.E. por Richard Wallace.
- Regras compostas de duas partes: **condição** e **ação**.
 - Se as condições forem satisfeitas, a ação é executada.
 - **Conflito**: quando mais de uma regra for satisfeita.
 - O interpretador precisa ter uma estratégia para resolver conflitos
- As **tags** mais importantes de um documento AIML são:
 - **<aiml>** inicia e termina um documento AIML;
 - **<category>** marca uma **unidade de conhecimento**;
 - **<pattern>** contem um padrão simples que pode fechar com o que o usuário fala ou digita;
 - **<template>** contem a resposta para uma entrada do usuário.

Interpretadores

- Intérpretes AIML são oferecidos sob uma licença livre ou de fonte aberta
- Muitos "clones Alicebot " tem sido criados baseados na implementação do programa original e na sua base de conhecimento

Interpretadores (alguns)

- RebeccaAIML:
 - C++, Java, .NET/C#, Python, Eclipse AIML editor plugin
 - <http://rebecca-aiml.sourceforge.net/>
- ChatterBean:
 - Java
 - <https://smallbusiness.yahoo.com/geocities>
- Program Q AIML:
 - C++, Qt
 - <http://sourceforge.net/projects/qaiml/>
- Program Y/PyAIML:
 - Python
 - <http://pyaiml.sourceforge.net/>

PyAIML

- Para usar o PyAIML:
 - Baixar e instalar o Python:
 - <https://www.python.org/>
 - Instalar o python-aiml (sudo apt-get install python-aiml)
 - Baixar o PyAIML
 - <http://sourceforge.net/projects/pyaiml/>
 - <http://pyaiml.soft112.com/>
 - Descompactar em um diretório
- Para editar os arquivos AIML com autocomplete e highlight:
 - Necessário ter o Eclipse
 - Baixar o plugin do RebeccaAIML para Eclipse
 - http://prdownloads.sourceforge.net/rebecca-aiml/RebeccaAIML_eclipse_plugin.zip?download
 - Descompactar o arquivo zipado
 - Copiar os arquivos JAR para a pasta plugin do eclipse.

PyAIML

```
# Programa chatbot.py
import sys
import aiml

kb = sys.argv[1]

k = aiml.Kernel()
k.learn(kb)
while True: print k.respond(raw_input("> "))
```

Utilização:

```
$ python chatbot.py arquivo.aiml
```


helloworld.aiml

```
<aiml version="1.0.1">  
  <category>  
    <pattern>Ol</pattern>  
    <template>Olá, como vai você?</template>  
  </category>  
</aiml>
```

-
- Utilizar em **<pattern>** caracteres em caixa alta, sem acentuação.
 - Demais exemplos são apresentados nos arquivos do conteúdo.