JARVIS

Introdução à esta versão: Versão para .Net Framework 4.5.2 com reconhecimento de voz pelo Microsoft Speech Server 11 com suporte às vozes SAPI 5.

Aplicações Atuais -

Gravador de Voz:

Media Player:

Editor de Texto:

Compressor de Arquivos GZ:

Especificação mínima:

Processador Intel Celeron Dual Core 1.5 Ghz

Memória RAM 2 GB de 1333 Mhz

HD de 5400 RPM

Conexão de rede para outras aplicações inclusas.

Como funciona?

Através do reconhecimento de voz, entrada de texto ou um clique, o programa executa algum comando, por exemplo para falar que horas são ao usuário ou outro comando, o interessante é usar pelo reconhecimento de voz, mas para isso é preciso um microfone, de boa qualidade se possível, alterando os níveis de sensibilidade o reconhecedor fica mais sensível, apesar de que as chances de erros no reconhecimento aumentarem, mas é recomendável.

Pontos Fracos:

Os pontos fracos deste programa no momento são pela engenharia do código e pela falta de uma IA para “turbinar” o negócio, o Jarvis poderia usar uma IA ligada ao processamento de linguagem natural para executar um comando, responder à uma pergunta, ou simplesmente conversar com o usuário final, quanto ao desempenho do programa, quanto em execução não há tantos problemas, ao não ser quando iniciado, pois há certo tempo até o carregamento, algo que pode ser resolvido com mais Hardware.

Pontos Fortes:

O principal ponto forte é o suporte ao reconhecimento de voz, mesmo sendo apenas de comandos, mas mesmo assim pode ser modelado para obter-se um melhor resultado. Quando o programa puder reconhecer uma sequência de até 5 ou 8 palavras dentro de um conjunto de 50 mil, 100 mil palavras do cotidiano, podemos dizer que temos suporte ao reconhecimento de voz que queremos.

IA como se faz?

A maioria das pessoas que sabem o que é IA pelo cinema acham que ela é algo perigoso e que deve ser evitada, mas o cinema explora para ele mesmo qualquer lado de qualquer coisa que o seja lucrativo. A IA em alguns sentidos é perigosa, se fosse “ensinada” para tal, essa consciência randômica é algo improvável de acontecer.

Agora como se faz IA? IA é feita por um processo incialmente simples que foi proposto para fazer algo, que quando novamente solicitado fizesse algo, exemplificando IA é algo que aprende e responde ou faz algo quando é pedido à ela. Existem dois tipos dela, a simbólica que apenas parece ser inteligente, e a conexionista que consegue aprender algo de verdade.

Simbólica – Como exemplo de IA simbólica mais famosa temos a Eliza, um ChatBot que tenta por meios lógicos imitar a conversação humana, foi muito explorada no final do século XX, ganhou até alguns prêmios, mas é um modelo que não tem aplicação real de inteligência. Esse modelo pode ser usado para executar comandos simples, apenas.

Conexionista – IA de verdade! A IA conexionista é a que é mais próxima de algo inteligente, pois usa atributos mais genéricos e versáteis, é muito mais difícil implementar ela pelas razões óbvias, sua complexidade, descrever um problema para ela é algo desmansiado.

Paradigmas – Alguns paradigmas são Machine Learning(Aprendizado de Máquina) dentre dele temos Neural Networks(Redes Neurais), Self-Organizing Maps(Mapas auto organizáveis), SVM(Vetores de suporte de máquina), esse tipo de IA analisa algum padrão, e os usa para gerar hipóteses, explicar é algo meio trivial, mas vamos lá.

Imagine que eu queira saber qual é o melhor processador até 800 reais, eu posso aprender sobre cada processador, ou posso perguntar à algumas pessoas que sabem sobre processadores, dentre esses dois perguntar é uma caminho bem mais curto, do que ter que aprender o que faz com que um processador seja bom, isso se chama aprendizado supervisionado, alguém deu as respostas certas, o problema desse método é quando alguém dar as respostas erradas, mas vamos considerar apenas.

Tudo isso gira em cima de cálculos, matrizes de dados em intervalos de (0 à 1) ou (-1 à 1), podemos dar o valor 0 significado de “não” e 1 “sim”, o números de entradas seriam a série de verdades ou mentiras sobre algo.

A filosofia usa de lógica, programação bem mais

Thing thing = new Thing(“João”);

If(thing.IsMan==true)

{

Console.WriteLine(“João é homem, logo é mortal”);

}

Esse é um exemplo bem simples de lógica, vamos considerar algo mais legal.

Entrada Tem vida/Não tem Vida Saída Mortal/Imortal

Dog 1.0 1.0 mortal

Bolsa 0.0 0.0 imortal

João 1.0 1.0 mortal

Aqui basta saber se algo tem vida para deduzir se é mortal ou não, apesar de haver certo erro lógico, seria apenas preciso ter um fonte de dados que diz se um objeto tem vida ou não.

Obs:( “=>” é igual a saída)

Pernas | Vida | Tem lugar de sentar => cadeira ou banco

Cão 1.0 1.0 0.0 0.0

Sofá 1.0 0.0 1.0 0.8 // é quase isso

Gato 1.0 1.0 0.0 0.0

Cadeira 1.0 0.0 1.0 1.0

Aqui depois de dizemos se algo tem pernas ou patas, se tem vida ou não e se tem lugar de sentar, teremos o resultado se algo é uma cadeira, banco ou algo parecido, mas ainda há exceções

Um cavalo tem patas, vida e lugar de sentar(montar) logo é algo próximo de uma cadeira, esse erro é pode ser minimizado com mais entradas e a separação certa dos dados.