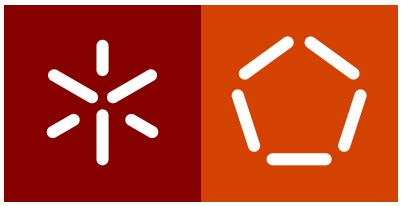
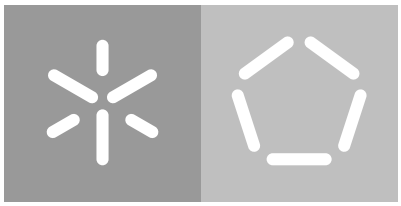


Universidade do Minho
Escola de Engenharia
Departamento de Informática

Nelson Manuel Figueiredo da Silva

Controlo de Periféricos por Voz

November 2019



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Nelson Manuel Figueiredo da Silva

Controlo de Periféricos por Voz

Master dissertation
Master Degree in Computer Science

Dissertation supervised by
Paulo Novais
Davide Carneiro

November 2019

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

ACKNOWLEDGEMENTS

Em primeiro lugar gostaria de agradecer ao meu orientador Paulo Novais e coorientador Davide Carneiro pelas sugestões e conselhos dados ao longo dos meses de desenvolvimento desta dissertação.

Gostaria de deixar uma mensagem de agradecimento ao grupo ISLab (Synthetic Intelligence Group), pelo apoio e sugestões dadas.

Queria igualmente deixar um agradecimento à minha família e amigos, pelo apoio dados ao longo destes últimos meses, o que em muito contribuiu para a qualidade do trabalho realizado.

ABSTRACT

Nowadays, mouse and keyboard are crucial perpherals for interacting with a computer. These peripherals allow computer users to navigate the windows, select items, type int text, among others. However, the use of such peripherals, which is considered elementary for most of us, is sometimes an obstacle to computer interaction, especially for users with physical limitations or fine motor skill problems. This dissertation presents a system that will allow the interaction with a computer using only the voice, without the need for a physical/mechanical interaction. This system also allows the user to, besides controlling the keyboard and the mouse cursor, carry out the most common tasks by using voice commands, such as creating a task for a date or conduct a web search. Lastly, this dissertation presents the efficiency tests carried out and the system acceptance in a real context.

RESUMO

Nos dias de hoje o controlo do rato e teclado apresentam-se como uma forma crucial de interação com um computador. É através do rato que podemos controlar o cursor que nos permite a navegação nas janelas, seleção de itens entre outros. A utilização de tais periféricos, como por exemplo o rato, que tantas vezes considerámos como elementares, é por vezes um obstáculo à interação com o computador, especialmente para utilizadores com limitações físicas ou de motricidade fina (capacidade da execução de movimento finos com controlo e destreza). Nesta dissertação será apresentado um sistema que permita o utilizador controlar um computador recorrendo apenas à voz, sem a necessidade de existir uma intervenção física/mecânica. Este sistema também permitirá que o utilizador, para além de controlar o teclado e o cursor do rato, consiga efetuar tarefas mais básicas, recorrendo a comandos de voz, como por exemplo criar uma tarefa para um dado dia ou uma simples pesquisa web. Por fim serão apresentados testes de eficiência e aceitação do sistema realizados em contexto real.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

A

AM Média Aritmética.

API Application Programming Interface.

C

CNN Convolution Neural Network.

D

DNN Deep Neural Network.

DNN-HMM Deep Neural Network Hidden Markov Model.

DTW Dynamic Time Warping.

F

FFT Fast Fourier Transform.

G

GM Média Geométrica.

GUI Graphical User Interface.

H

HMM Hidden Markov Models.

HTTPS Hyper Text Transfer Protocol Secure.

J

JSON JavaScript Object Notation.

L

LSTM Long Short Term Memory.

P

PDF Portable Document Format.

R

RNN Recurrent Neural Network.

S

SFM Spectral Flatness Measure.

SVM Support Vector Machine.

V

VQ Vector Quantization.

W

WER Word Error Rate.

1

INTRODUÇÃO

O conceito de utilizar a linguagem natural, como forma de permitir uma comunicação Homem-Máquina fluida, tem ganho cada vez mais interesse por parte de empresas como a *Google*, a *Microsoft*, o *Facebook*, ou a *Apple* que têm vindo a apostar, cada vez mais, no desenvolvimento de sistemas capazes de reconhecer e compreender linguagem natural.

Chatbot é o termo dado a um programa de computador que oferece a possibilidade de manter um sistema de conversação com um humano através da utilização de linguagem natural. O ponto crítico num chatbot é claramente o potencial que este contém em entender e interpretar a linguagem natural, que pode ser dividida em várias áreas incluindo: reconhecimento de voz, identificação de palavras, inteligência artificial entre outras (Abdul-Kader and Woods (2015)). Com a utilização de chatbot é possível desenvolver sistemas de conversação Homem-Máquina que permitem o utilizador controlar sistemas recorrendo a linguagem natural.

Pessoas com limitações físicas, como por exemplo problemas relacionados a mobilidade fina, vêm um computador como algo de difícil utilização. A dificuldade que estas pessoas apresentam na utilização de um computador resultam numa privação à sua utilização e consequentemente a uma diminuição na sua qualidade de vida (Armando B. Barreto and Adjouadi (2000)).

Uma forma alternativa, e não intrusiva para o utilizador, é a utilização de chatbots que recorrendo a técnicas de reconhecimento de voz permitam o utilizador, que era outrora privado da utilização de um computador, o possa agora fazer (Yaghoubzadeh et al. (2013)).

A utilização da voz no que toca a controlo do computador pode não ser das técnicas mais eficientes quando comparado com dispositivos físicos, mas claramente se apresenta como um método menos intrusivo e menos dispendioso, uma vez que atualmente qualquer computador contém uma placa de som e microfone.

1.1 CONTEXTO E MOTIVAÇÃO

O rato foi um dos dispositivos de entrada de computadores mais bem sucedidos e penetrantes desde o surgimento da interface gráfica do usuário na década de 1960. Junto com o teclado, moldou a maneira como as pessoas interagem com os computadores. Os atuais sistemas operativos de computadores, como *Windows*, *Mac OS* e muitas variantes do *Linux*, oferecem interfaces contendo diversos ícones, menus, entre outros. O sucesso do rato como um dispositivo de entrada e a *Graphical User Interface (GUI)* como a interface de computador predominante podem ser atribuídos principalmente ao mapeamento intuitivo e simples entre a manipulação necessária do dispositivo e o resultado resultante apresentado diretamente na interface gráfica. Apesar do sucesso das interfaces de computador centradas no rato de hoje, o pressuposto subjacente da disponibilidade de um dispositivo apontador no processo de projeto do sistema, impediu que tais interfaces fossem facilmente acessíveis por pessoas para quem o uso de um rato não é uma opção ou apresenta-se como um enorme desafio. Alguns dos motivos que impedem o uso do rato podem incluir vários tipos de deficiências motoras (por exemplo, artrite, distrofia muscular, lesões da medula espinhal, amputação e problemas de mobilidade fina), bem como deficiências situacionais (por exemplo, ambientes móveis e mãos ocupadas para outras tarefas). Neste sentido pretende-se então criar uma ferramenta que permita controlar o rato e o teclado através da fala, permitindo deste modo controlar todo o computador sem necessidade da existência de qualquer contacto físico entre o humano e um teclado ou rato.

1.2 TEMA E OBJETIVOS

Esta dissertação tem como foco principal a construção de uma interface de linguagem natural (chatbot) para controlo do rato e teclado através de comandos de voz. O chatbot deverá permitir, através da voz, desempenhar funções mais básicas da utilização destes periféricos: movimento em 2D, clique, duplo clique, scroll, clique com o botão direito e escrita.

Nos sistemas existentes no mercado, observa-se que estes são por vezes limitativos na forma que o utilizador pode interagir com o computador, limitando-se a apresentar uma série de comando pré definidos que o utilizador terá obrigatoriamente de utilizar. Os sistemas não apresentam uma variedade alargada de formas às quais o utilizador possa utilizar para o controlo do computador, ficando estes sistemas limitados aos algoritmos que foram implementados aquando do seu desenvolvimento.

Com o intuito de resolver o problema acima descrito, nesta dissertação é feita uma análise de diferentes algoritmos de controlo do rato e teclado através da voz com o intuito de perceber qual algoritmo se adapta mais a cada tipo de situação, seja em navegação web ou redigir um texto. Com esta análise previamente feita será possível dar ao sistema a capacidade de o mesmo se adaptar de acordo com a preferência do utilizador. Para além de existir esta componente de aprendizagem do sistema e sua adaptação a diferentes situações, irá ser possível também que o utilizador não esteja limitado apenas a comandos pré definidos de voz e sim os possa customizar. Por fim numa forma mais simplista o sistema irá permitir a programação de vários comandos que otimizem tarefas que realizamos frequentemente num computador como ver email ou marcar um evento no calendário permitindo deste modo que numa simples frase consiga expressar um leque alargado de ações.

O principal resultado deste trabalho será um protótipo de uma aplicação, que será submetida a validação de uso em contexto real, com a realização de tarefas comuns que normalmente realizamos num computador.

1.3 METODOLOGIA DE TRABALHO

Este projeto será desenvolvido segundo uma metodologia investigação-ação em que a ação e a investigação ocorrem ao mesmo tempo, utilizando um processo cíclico que alterna entre ação e reflexão critica. Pretende-se com este método que perante a presença de um dado desafio é estimulada uma hipótese de solução. Será efetuado uma compilação e organização de informação relevante para o problema e em seguida concebida uma proposta de solução. Na etapa final, serão formuladas as respetivas conclusões que permitem avaliar os resultados obtidos. Para seguir esta metodologia serão realizados os seguintes passos:

1. Especificação do problema e suas características
2. Atualização e revisão constante do estado de arte
3. Modelação e implementação do sistema
4. Análise de resultado e formulação de conclusões
5. Validação do sistema

1.4 ESTRUTURA DO DOCUMENTO

Este documento encontra-se estruturado da seguinte forma:

- No capítulo 2 é inicialmente apresentado a arquitetura base de um sistema de conversação por voz Homem-Máquina, expondo os componentes envolvidos neste tipo de sistemas. Juntamente com a apresentação destes sistemas, é feita uma análise e apresentação de desafios a que se encontra atualmente exposto os sistema de conversação Homem-Máquina, fazendo uma análise a cada um dos componentes envolvidos nestes sistemas, enumerando diversas ferramentas já existentes no mercado de reconhecimento de voz e processamento de linguagem natural. Por fim será feita uma apresentação de algumas técnicas para o controlo do cursor do rato utilizando comandos de voz.

- No capítulo 3 é feita uma análise a diversos sistemas existente de processamento de linguagem natural e de reconhecimento de voz que são à data os mais populares. Será feito um comparativo entre dois grupos de sistemas que atualmente são gratuitos com sistemas que contêm algum custo por forma a verificar qual melhor se adapta ao sistema que pretendemos desenvolver e de todos qual contem uma maior eficácia no que diz respeito a processamento de linguagem natural juntamente com reconhecimento de voz.

- No capítulo 4 inicialmente é feito um levantamento de desafios que os sistemas para controlo do computador com a voz estão expostos. Por fim é apresentada a proposta de solução ao qual o resultado deste protótipo pretende resolver juntamente com a apresentação de testes de eficiência e eficácia do sistema na sua utilização em contexto real.

- No capítulo 5 é apresentada alguma reflexões criticas ao trabalho realizado bem como apresentação de algumas ideias que seriam interessantes para serem abordadas num trabalho futuro.

2

LEVANTAMENTO DO ESTADO DA ARTE

Neste capítulo é feita uma apresentação da estrutura mais habitual de um sistema de conversação por voz Homem-Máquina, enumerando os seus diferentes componentes, e como os mesmos são amplamente usados em diversas áreas. De seguida serão mostrados alguns trabalhos realizados no que diz respeito à aplicação de conversação por voz que visam ajudar pessoas com limitações físicas ou cognitivas. Por fim serão apresentados alguns algoritmos utilizados em aplicações reais que têm como objetivo melhorar a performance relativamente à movimentação do cursor através de comandos de voz.

2.1 SISTEMAS DE CONVERSAÇÃO POR VOZ

Sistemas de conversação por voz, como assistentes virtuais, podem simplificar e mudar a forma como vivemos. Através destes com o poder de uma simples frase podemos executar ações sem a necessidade de existir a priori uma aprendizagem na utilização do sistema. Com o avanço dos sistemas de reconhecimento de voz é possível recorrendo apenas a linguagem natural realizar atividades como pedir alguma informação, gestão do calendário (Modi et al. (2005)) ou até mesmo realizar algo mais crítico como a realização de uma transação monetária.

A fala é uma das formas mais eficazes na comunicação entre humanos, devido a este facto, atualmente gigantes da tecnologia têm apostado forte na criação de assistentes virtuais, podemos verificar que por exemplo nos últimos anos empresas como *Microsoft* com o *Cortana*, *Apple* com a *Siri*, *Google* com o *Google Assistant* têm apostado forte neste conceito com o objetivo de apresentarem a melhor ferramenta possível.

Este tipo de sistema são bastante complexos (David Goddeau and Busayapongchaiy (1996), Glass (1999), Pierrick Milhorat and Sudparis) que incorporam uma variedade de tecnologias relacionada com o discurso e processamento de linguagem natural, como re-

conhecimento de voz, compreensão de linguagem natural, gestão do discurso, geração de linguagem natural e síntese da fala. Todas estas tecnologias para além de terem de funcionar de forma sincronizada devem também funcionar em tempo real, para permitir que o feedback dado ao utilizador seja o mais rápido possível.

Dos sistemas mais simples aos mais complexos todos estes apresentam algumas similaridades no que diz respeito à sua arquitetura. De uma forma mais simplista e conhecida de um sistema de conversação por voz, o sistema começa por aceitar uma expressão e depois de iterar sobre vários componentes do sistema produz um resultado que será reflexo da resposta dada ao utilizador, como ilustrado na figura 1 onde é possível visualizar os diferentes componentes de um típico sistema deste tipo. Como podemos observar estes sistemas são tipicamente constituídos por um módulo de *Speech Recognition* (reconhecimento de voz) que é responsável por traduzir um componente de áudio na respetiva representação textual. O módulo *Natural Language Understanding* (compreensão de linguagem natural) é responsável por interpretar o resultado dado pelo *Speech Recognition* e representá-lo de uma forma que seja possível ser interpretado pelo *Dialog Manager*. O módulo *Dialog Manager* (gestor do discurso) é responsável por controlar a interação com o utilizador, processando o contexto e o seu discurso e produzindo uma resposta com nível semântico. Por fim os dois últimos módulos, *Response Generation* que têm o papel de dada a informação produzida pelo *Dialog Manager* produzir um texto em linguagem natural e por fim o *Speech Synthesis* será responsável por pegar no texto produzido pelo *Response Generation* e converter o texto em linguagem natural em forma sonora que será a resposta fornecida pelo sistema ao utilizador. Muitos sistemas de conversação por voz típicos também contêm comunicação externa (Spiliotopoulos et al. (2001), Zue and Glass (2000)) através do *Dialog Manager*, por exemplo uma comunicação com uma base de dados externa para obtenção de alguma informação extra.

Nos subcapítulos seguintes será feito um breve estado da arte de cada uma das componentes descritas acima, que igualmente se encontrão representadas na figura 1, por forma a perceber melhor todos os componentes envolvidos nos sistema de conversação por voz e como cada um dos módulos contêm dificuldades e complexidades aos quais novos algoritmos têm surgido com o objetivo de melhorar tais sistemas.

2.1.1 *Reconhecimento de Voz*

A tarefa do módulo de reconhecimento de voz, envolvidos nos sistemas de conversação Homem-Máquina, é a tarefa de dado um sinal acústico contínuo o transformar numa sequência discreta de palavras. A importância de realizar esta tradução antes de execu-

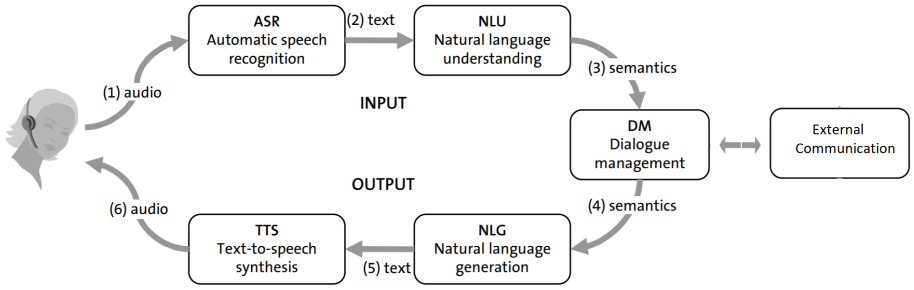


Figure 1: Arquitetura típica de assistentes virtuais

tar a ação deve-se ao facto de num sinal sonoro não existir qualquer tipo de informação que ao olho humano contenha conteúdo, contudo se o sinal sonoro poder ser traduzido para texto em linguagem natural poderemos desde logo inferir ações dessa mesma frase o que torna o modelo de conversação por voz bem mais simples de ser modulado.

O reconhecimento de voz contem uma série de desafios, sobretudo relativamente ao grau de variabilidade do sinal sonoro, por exemplo, diferentes pessoas ao discursarem emitem diferentes propriedades acústicas, o que se apresenta um enorme obstáculo no que toca ao reconhecimento de voz uma vez que estes tipos de sinais não são determinísticos. A variabilidade sonora pode existir de vários fatores, como por exemplo:

- **Variabilidade da Linguagem:** São efeitos sobre o sinal de voz causados por vários fenómenos linguísticos. Por exemplo, o mesmo fonema pode ter diferentes compreensão acústicas em diferentes contextos, determinados pelos fonemas anteriores e seguintes em questão.

- **Variabilidade do Orador:** Corresponde às diferenças entre oradores, relativas a fatores físicos tais como a forma do trato vocal, a idade, género, origem regional e variações por parte do mesmo orador, uma vez que as mesmas palavras faladas em diferentes ocasiões pelo mesmo orador podem ter diferenças em termo de propriedades acústicas. Fatores físicos como cansaço, mudança de humor, a localização de uma palavra dentro de uma frase e do grau de ênfase que é dada á palavra, têm influência direta sobre a forma que a palavra é prenunciada.

- **Variabilidade do Canal:** Este tipo de problemas diz respeito a ruído de fundo, que pode ser constante ou transitório, e ruído relativo ao canal de comunicação que pode ser por exemplo através da Internet ou de um microfone.

Um sistema de reconhecimento de voz típico de um sistema de conversação Homem-Máquina também deve ser capaz de lidar com os seguintes fatores adicionais (McTear (2002)):

- **Independente de Orador:** Para que o sistema seja possível de ser utilizado por uma variedade de utilizadores é importante que o módulo de reconhecimento de voz não fique treinado com um conjunto limitado a apenas um orador ou pequeno grupo, e sim permita ser independente, ou seja, dependendo do orador que utiliza o sistema este deve ser capaz de apresentar uma resposta válida. Apesar de parecer algo trivial este fator, como outros, representa uma dificuldade da conceção de tais sistemas uma vez que existe uma enorme variedade de possíveis acústicas. Os sistemas que tendem a ser o mais globais possíveis, devido exatamente à grande diferença entre oradores, tendem a ser mais sujeitos a erros comparativamente com sistema que são focados num certo grupo de utilizadores.

- **Tamanho do Vocabulário:** O tamanho do vocabulário varia com a aplicação e com a conceção particular do sistema de diálogo. Assim, um diálogo cuidadosamente controlado pode restringir o utilizador a um vocabulário limitado a alguma palavras que expressam as opções que estão disponíveis no sistema, enquanto num sistema com um vocabulário mais flexível pode aceitar um leque mais alargado de palavras.

- **Fala Contínua:** Os utilizadores de sistemas de conversação por voz esperam ser capazes de falar normalmente com o sistema. No entanto, é difícil de determinar um limite de palavras durante o decorrer da fala, uma vez que não existe uma separação física no tempo contínuo do sinal de voz. O que se faz mais habitualmente é considerar que a deteção de uma frase é resultado de reconhecer a separação entre discurso e silêncio. Poderemos por exemplo considerar num sistema que sempre que exista um discurso seguido de silêncio durante um período de tempo (por exemplo um threshold de 400ms) que o discurso é considerado uma frase que deve ser analisada, contudo esta separação nem sempre é simples. O sinal pode conter ruído, outro fator deve-se ao facto de calcular qual o threshold ”bom” pois para além dos oradores no decorrer do discurso fazerem pausas, um aumento no valor de threshold leva a que o sistema demore mais tempo a responder uma vez que o mesmo demora mais tempo para detetar que o silêncio atual é indicação de fim de frase.

- **Fala Espontânea:** Uma vez que a fala do utilizador é a entrada para o sistema de conversação por voz, ela, normalmente, é espontânea e não planeada. O que significa que o discurso pode conter defluências, como hesitações e enchimentos, como por exemplo ”hmmmmm” ou ”ahhh”, falsos inícios do discurso em que o orador começa o discurso e em seguida faz pausas e depois começa de novo, e fenómenos linguísticos

naturais, tais como tosse. Para tentar resolver este problema têm surgido soluções com o intuito de modular este tipo de acústicas de modo a poderem a serem detetadas e não constituírem de certo modo um falso positivo para o sistema de reconhecimento de voz (Zue and Glass (2000)). O reconhecedor de voz tem de ser capaz de extrair do discurso uma sequência de palavras para que a partir das quais o seu significado possa ser tratado.

De acordo com Valiyavalappil Haridas et al. (2018) o papel do reconhecimento de voz deve se preocupar com a inconsistência de fala e é responsável por aprender a associação entre os sons específicos e a palavra ou palavras relacionadas. Existem essencialmente três diferentes tipos de técnicas relacionadas com o reconhecimento de voz sendo elas:

1. **Abordagem acústica-fonética:** A técnica acústica-fonética (Likitsupin et al. (2016)) lida com a transmissão dos sons do orador para o ouvinte, esta abordagem oferece uma oportunidade para estudar a natureza do sinal da fala para diferentes sons, independentemente das características que representam esses sons. Não inclui técnicas de modelagem e métodos de extração de características, mas sim uma maneira de analisar e compreender a natureza de diferentes sons como vogais, semivogais, ditongos entre outros. A técnica de acústica-fonética, deste modo, concentra-se na compreensão das unidades fonéticas e as suas relações com diferentes contextos de uso (Parabattina and Das (2016)). O fundamento desta técnica baseia-se no fato de existirem uma série de fonemas finitos e exclusivos na linguagem falada, os quais podem ser amplamente representados por um grupo de traços acústicos que são exibidos no sinal da fala durante um período de tempo. Mesmo que as propriedades acústicas de unidades fonética sejam altamente variáveis, tanto devido ao orador como os sons que rodeiam os fonemas, esta técnica assume que as regras que governam esta variabilidade são diretas e podem ser aprendidas pela máquina. O primeiro passo na técnica de acústica-fonética é a análise do espectro da fala que combinado com a deteção de características, descreve as propriedades acústicas de diferentes unidades fonéticas. O passo seguinte é a segmentação e classificação da fala, onde o sinal é segmentado em regiões acústicas estáveis, seguido pela associação com a respetiva classificação a cada uma das regiões segmentadas, resultando num conjunto de fonemas. O último passo é determinar uma palavra válida a partir da classificação dada previamente.

2. **Abordagem de reconhecimento de padrões:** A técnica de reconhecimento de padrões não requer nenhuma consciência explícita sobre o discurso. Essa técnica prossegue por duas fases, o treino de padrões de fala por um conjunto de parâmetros espectrais comuns e a deteção de padrões por meio da análise de padrões. Esta abordagem oferece alta precisão e baixo custo computacional. Muitas técnicas de codificação de

imagens simples estão disponíveis na abordagem de reconhecimento de padrões. Esta abordagem fornece um método eficaz de combinar as contribuições de um número de medições do som da fala. A característica vital desta abordagem é que ela emprega uma estrutura matemática bem formulada e cria representações confiáveis de padrões da fala, para comparação consistente de padrões. Os métodos de reconhecimento de padrões mais conhecidos englobam as técnicas baseadas em templates e técnica estocástica. O modelo estocástico é o mais apropriado para o reconhecimento da fala, pois emprega técnicas probabilísticas para endereçar dados incertos ou imperfeitos. Existe uma enorme variedade de técnicas seguindo essa abordagem que incluem *Hidden Markov Models (HMM)*, *Support Vector Machine (SVM)*, *Dynamic Time Warping (DTW)*, *Vector Quantization (VQ)* e assim por diante. Entre as técnicas, destaca-se o HMM, que surgiu como a abordagem estocástica mais conhecida (Gales and Young (2007)).

a) **Abordagem baseada em templates:** A ideia subjacente ao método baseado em templates (De Wachter et al. (2007), Gbadamosi (2013)) é simples. Uma coleção de padrões de fala são armazenados como padrões de referência que representam o dicionário das palavras candidatas. O reconhecimento é então realizado através da correspondência de um enunciado falado desconhecido com cada um dos modelos de referência e selecionando a categoria que contém a melhor correspondência de padrões reconhecidos. Normalmente, modelos para palavras inteiras são construídos. Isto tem a vantagem de que erros devido à segmentação ou classificação de unidades acústicas menores, mais variáveis, como fonemas, podem ser evitados. Por sua vez, cada palavra deverá conter o seu próprio modelo de referência completo. A preparação e combinação de modelos se tornam proibitivamente caras ou impraticáveis à medida que o tamanho do vocabulário aumenta além de algumas centenas de palavras. Uma ideia-chave no método de templates é derivar uma sequência típica de quadros de fala para um padrão (uma palavra) através de algum procedimento de média, e confiar no uso de medidas de distância espectral local para comparar padrões. Outra ideia-chave é usar alguma forma de programação dinâmica para alinhar temporariamente os padrões para explicar as diferenças nas taxas de fala entre os participantes, bem como nas repetições da palavra pelo mesmo locutor. A abordagem baseada em modelos tem o benefício de que falhas devido à classificação ou segmentação de unidades acústicas menores mais variáveis, como por exemplo, fonemas, podem ser evitadas.

b) **Abordagem estocástica:** A modelagem estocástica (Dixit and Kaur (2013)) envolve a implementação de técnicas probabilísticas para lidar com dados vagos

ou imperfeitos. Na tarefa de deteção de fala, a ambiguidade e a imperfeição originam-se de uma variedade de fatores, como os sons confundíveis, a inconsistência do falante, os impactos contextuais e as palavras homófonas. A abordagem estocástica fornece mais informações sobre possíveis resultados futuros, levando, assim, à melhoria da tomada de decisões. Essa abordagem envolve o emprego de modelos probabilísticos para lidar com informações incompletas. Por isto, os modelos estocásticos são apropriados para o reconhecimento de voz. Atualmente, é utilizada a bem-sucedida técnica estocástica conhecida como o HMM, que representa um modelo de Markov de estado finito e um conjunto de distribuições de saída. As restrições de transição nos modelos da cadeia de Markov são a variabilidade temporal, enquanto a restrição nos modelos de distribuição de saída, e são a variabilidade espectral. Esses dois tipos de variabilidade são o núcleo da função de reconhecimento da fala. Quando comparada com a técnica baseada em modelos, o HMM é mais abrangente e possui uma forte base matemática. Considerando que, o modelo baseado em templates se refere a uma densidade ininterrupta HMM, com matrizes de covariância de identidade e uma topologia restrita por declive. Mesmo que os modelos possam ser treinados, em alguns casos, eles sofrem com a falta de formulação probabilística de HMM abrangentes e classicamente têm desempenho pior do que os HMM. Quando comparados com as técnicas baseadas em conhecimento, os HMM facilitam a incorporação sem esforço de fontes de conhecimento em uma arquitetura montada. Um efeito adverso disso é que os modelos de HMM não fornecem muita compreensão do processo de reconhecimento. Como resultado, é difícil a avaliação de erros de um sistema HMM com o intuito de aumentar sua eficiência. No entanto, uma cuidada assimilação de conhecimento tem consideravelmente aperfeiçoado os sistemas baseados em HMM (Rabiner (1989), Gales and Young (2007)).

3. **Abordagem com inteligência artificial:** A abordagem baseada em inteligência artificial (Hinton et al. (2012a)) pode ser entendida como uma técnica híbrida que explora os conceitos da abordagem acústica-fonética e da abordagem de reconhecimento de padrões, incorporando os conceitos e ideias de ambas as abordagens. O método baseado em inteligência artificial é baseado na utilização do conhecimento relativos à linguística, fonética e espectrograma do som. Alguns investigadores de fala desenvolveram um sistema de reconhecimento que utilizou o conhecimento fonético acústico para desenvolver regras de classificação para os sons da fala. Embora as abordagens baseadas em modelos tenham sido muito eficazes no desenvolvimento de uma variedade de sistemas de reconhecimento de voz, este tipo de modelo fornece pouco conhecimento sobre o processamento de fala humana, tornando difícil a análise de

erros e o aprimoramento do sistema baseado em inteligência artificial. Por outro lado, um grande corpo de literatura linguística e fonética forneceu a compreensão para o processamento da fala humana. Em sua forma pura, o projeto de engenharia do conhecimento envolve a incorporação direta e explícita de conhecimento de fala especializado em um sistema de reconhecimento. Este conhecimento é geralmente derivado do estudo cuidadoso de espectrogramas e é incorporado usando regras ou procedimentos. A engenharia pura do conhecimento também foi motivada pelo interesse e pesquisa em sistemas especialistas. No entanto, esta abordagem teve apenas sucesso limitado, em grande parte devido à dificuldade em quantificar o conhecimento especializado. Outro problema difícil é a integração de muitos níveis de conhecimento humano fonético, fonética, acesso lexical, sintaxe, semântica e pragmática. Alternativamente, a combinação de fontes de conhecimento independentes e assíncronas permanece otimamente um problema não resolvido. Em formas mais indiretas, o conhecimento também tem sido usado para guiar o design dos modelos e algoritmos de outras técnicas, como modelagem de modelos e correspondência coincidente. Esta forma de aplicação do conhecimento faz uma importante distinção entre conhecimento e algoritmos. Algoritmos permitem a resolução de problemas. O conhecimento permite que os algoritmos funcionem melhor. Essa forma de aprimoramento do sistema baseado em conhecimento contribuiu consideravelmente para o design de todas as estratégias bem-sucedidas relatadas. Ele desempenha um papel importante na seleção de uma representação de entrada adequada, na definição de unidades de fala ou no design do próprio algoritmo de reconhecimento.

Nos últimos anos inúmeros investigadores têm vindo a focar-se na utilização de algoritmos de *Deep Leaning* para estudo de técnicas de reconhecimento de voz. Em George E. Dahl and Acero (2012) é proposto um novo modelo dependente do contexto para o reconhecimento de voz em grandes vocabulários que aproveita os recentes avanços no uso de redes profundas para o reconhecimento de telefonemas. Utilizando para isso uma arquitetura híbrida de rede neural profunda pré-treinada, *Deep Neural Network Hidden Markov Model (DNN-HMM)* que treina a *Deep Neural Network (DNN)* para produzir uma distribuição sobre senones como sua saída. Para eficácia dos sistemas os fonemas são diferenciado pelo contexto de vizinhança fonética, utilizando "trifones". Um trifone representa um único fonema, dados os fonemas anteriores e o fonema posterior. O senone representa a divisão do trifone em três sub estados, denominado por "senones". O primeiro estado corresponde ao inicio de fonema, o segundo corresponde a parte intermédia do fonema e o terceiro estado corresponde á parte final do fonema.

Em Xiong et al. (2017) é apresentado pela *Microsoft* o sistema de reconhecimento de voz onde é utilizado uma solução que usa redes DNN juntamente com rede bidirecionais

Long Short Term Memory (LSTM) para o modelo acústicos e uma rede LSTM para o modelo linguístico.

Em Graves et al. (2013) com a utilização de redes LSTM em *Recurrent Neural Network (RNN)* bidirecionais profundos com treinamento de ponta a ponta e com utilização de ruído Gaussiano para os pesos. De acordo com os autores o ruído Gaussiano permite simplificar a rede reduzindo a quantidade de informação necessária para a aprendizagem resultando numa maior generalização da rede. Os resultados obtidos com a utilização do dataset TIMIT mostraram uma taxa de erro para fonemas de 17,7%.

Muitas outras variantes têm vindo a imergir com a utilização de diferentes modelos de *Deep Learning* com a utilização de redes *Convolution Neural Network (CNN)* (Hinton et al. (2012b), Abdel-Hamid et al. (2014)) e variantes com a sua complexidades (Xiong et al. (2017)). Também com a utilização de DNN (Xiong et al. (2017)) com e muitas variantes por forma a apresentarem o melhor sistema possível (Chiu et al. (2017), Chorowski et al. (2015)) e conseguirem ultrapassar todos os desafio que são apresentados (Valiyavalappil Haridas et al. (2018)).

2.1.2 *Compreensão de Linguagem Natural*

O módulo de compreensão de linguagem natural é responsável por pegar no output dado pelo reconhecedor de voz e produzir uma relação semântica entre as palavras. O principal objetivo deste módulo é retirar o significado de uma expressão. Mas não é óbvio como isto deve ser entendido no contexto de sistemas de conversação por voz. Em Bangalore et al. (2006) é defendido como "uma representação que pode ser executada pelo um interpretador com o objetivo de mudar o estado do sistema".

A compreensão da linguagem envolve uma análise sintática, para determinar uma estrutura constitutiva da cadeia reconhecida por um grupo de palavras juntas, e análise semântica, para determinar os significados dos elementos. Outras abordagens para uma compreensão da linguagem podem envolver pouco ou nenhum processamento sintático e gerar uma representação semântica diretamente a partir da cadeia reconhecida.

Para sistemas de diálogo mais simples, onde a análise semântica não é importante, uma lista de chave-valor pode ser optada. Por exemplo, num sistema de preenchimento de formulário de viagens o que interessa saber é o destino e em que dia, uma simples frase "Marcar viagem para Portugal" neste tipo de sistema uma simples análise seria associar a

chave ”destino” com ”Portugal”. Apesar de serem sistemas mais simples restringem o que o utilizador pode ou não dizer.

Em sistemas em que e necessário um diálogo mais fluido será necessário optar por incluir neste modulo a capacidade de relacionamento semântico entre palavras dentro de uma expressão (Pierrick Milhorat and Sudparis, Hirschberg and Manning (2015)) por forma a permitir o sistema de prever intenções e conteúdo dentro da frase como nome de pessoas, cidades ou outro valor especifico. Nas ultimas décadas algoritmos de machine learning têm vindo a ser utilizados em problemas de processamento de linguagem natural (ex. *Supported Vector Machine* e *Logistic Regression*). Redes neuronais baseados em *Dense Vector Representation* impulsionados pelo sucesso dos modelos de *Deep Learning* e *Word embedding* têm vindo a apresentar resultados superiores em problemas de processamento de linguagem natural (Young et al. (2018)). Um dos impulsionadores principais foi a possibilidade da representação de palavra em um conjunto de vetores nos quais as redes de *Deep Learning* fossem capazes de perceber e interpretar o resultado, esta representação foi impulsionada pelo Mikolov et al. (2013a) e Mikolov et al. (2013b) onde propuseram os modelos CBOW e skip-gram. O CBOW calcula a probabilidade condicional de uma palavra alvo, dadas as palavras de contexto que a rodeiam, através de uma janela de tamanho k. Por outro lado, o modelo skip-gram faz exatamente o oposto do modelo CBOW, prevendo as palavras de contexto adjacentes dada a palavra alvo central. A representação de palavra em vetores é interessante na medida em que consegue captar relações semânticas e sintáticas entre palavras e são, por isso, capazes de exprimir similaridades entre diferentes palavras. Deste modo a similaridade entre palavras é possível calcular utilizando por exemplo o valor do cosseno entre vetores, onde palavras mais similares tendem a ter um ângulo entre vetores mais curto como ilustrado na figura 2, outro facto interessante é permitir que através de cálculo de vectores, vec(”rei”) - vec(”homem”) + vec(”mulher”), traduz num vetor que é próximo do vector vec(”rainha”).

Seguindo a popularização do *Word embedding* e sua capacidade de representação das palavras em espaços distribuídos, existe a necessidade de uma forma funcional e efetiva para extração de atributos de alto nível. Sendo esses recursos variados, utilizados no processamento de linguagem natural, como análise de sentimentos, resumo, tradução automática e resposta a perguntas. O uso de RNN têm mais se popularizado no processamento de linguagem natural por modular unidades em sequencia, tem a capacidade de capturar a natureza sequencial inerente na linguagem, onde as unidades podem ser caracteres, palavras ou até mesmo sentenças. Palavras em uma língua desenvolvem seu significado semântico com base nas palavras anteriores da sentença (Mesnil et al. (2015)). Um exemplo simples afirmando isso seria a diferença de significado entre *”dog”* e *”hot dog”*.

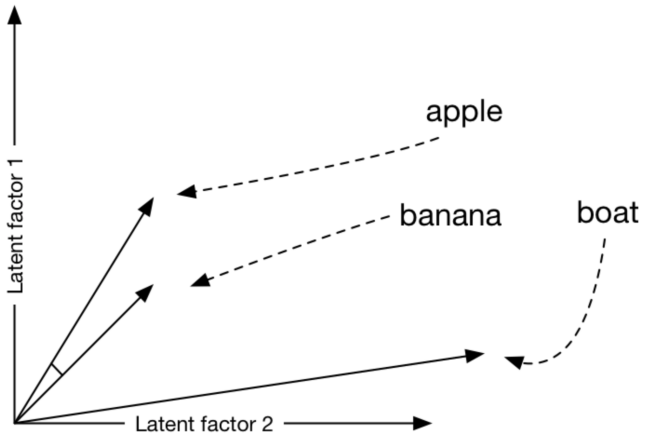


Figure 2: Representação visual entre palavra baseado na métrica do cosseno [1]

Os algoritmos de *Deep Learning* que utilizam *Word embedding* são capazes de capturar informações sintáticas e semânticas, mas para tarefas como marcar partes do texto e *Name entity recognition*, informações morfológicas também podem ser muito úteis. Existem três tarefas essenciais quando se pretende processar o que foi proferido pelo utilizador: deteção do domínio, determinação da intenção e *slot filling*. Em Dos Santos and Guimarães (2015) sugerem uma abordagem utilizando DNN (chamada de CharWNN) que utiliza reprentações ao nível do carácter, junto com *Word embedding* para deteção de *Name entity recognition* independente da linguagem. Em Liu and Lane (2016) propuseram uma rede RNN bidirecional com o intuito de juntar os dois módulos de deteção de intenções e *slot filling* permitindo os dois modelos serem treinado ao mesmo tempo e no final concluíram que a junção melhorou os resultados tanto da deteção de intenções como *slot filling* em comparação com os dois modelos em separado. Em Yang et al. (2016) apresenta um modelo também baseado em RNN por forma a fazer junção do processamento de linguagem natural com o gestor do diálogo.

2.1.3 *Gestão do Diálogo*

O módulo de gestão do diálogo tem a responsabilidade de controlar a interação do utilizador com o sistema de conversação por voz. Como se encontra expresso na figura 1, este módulo é o componente central de sistemas de conversação Homem-Máquina. A partir de uma representação semanticamente anotada e analisada, resultado do módulo de com-

1 Imagem adaptada de https://erikbern.com/assets/2015/09/vector-model1.png

preensão de linguagem natural que representa a entrada do discurso do utilizador, gera a ação apropriada no sistema.

O controlo do diálogo é feito com base em diferentes regras que o gestor de diálogo deve seguir de modo que o sistema funcione adequadamente. Numa fase mais inicial do sistema, o controlador poderá, por exemplo, ter a função de recolher alguma informação relativa ao utilizador do sistema, para possivelmente esclarecer conteúdo ambíguo ou terminar uma query que o utilizador se encontra ou pretenda fazer no futuro. Este componente, deve portanto, ser capaz de captar situações ambíguas (Zue and Glass (2000)) como por exemplo *"Tu disseste Braga ou Lisboa?"* ou especificações incompletas (*"Em que dia gostaria de viajar?"*), que necessitam de mais informação para permitir que o sistema execute a ação desejada, para além de aspetos mais complexos como a recuperação de erros resultado do reconhecimento e compreensão do discurso de entrada.

Com o objetivo de conseguir cumprir as tarefas anteriormente referidas o controlador do discurso deve rastrear o histórico do diálogo e atualizar alguma representação do estado atual do diálogo. Para além disso, o controlador de diálogo necessita de uma estratégia de diálogo que defina o comportamento conversacional do sistema, como por exemplo, quando este deve tomar a iniciativa do diálogo. O desempenho geral destes sistemas de conversação Homem-Máquina está altamente dependentes da estratégia adotada. A seguir são apresentados alguns exemplos de estratégias de diálogo, uma vez que não existe uma estratégia de diálogo perfeita. Tal como acontece com o modulo de reconhecimento de voz a variabilidade da população faz com que as estratégias adotadas por sistemas de conversação Homem-Máquina tenham de ser variadas, uma vez que, por exemplo duas pessoas diferentes conseguem produzir um discurso diferentes para um mesmo objetivo.

Assim sendo, o controlador de diálogo pode conter uma estratégia de iniciativa, na qual o sistema tem controlo total sobre o diálogo fazendo perguntas especificas ao utilizador, por exemplo, *"Qual o destino da viagem"*. Nestes sistemas o utilizador é normalmente sempre interrogado de modo a que o sistema consiga obter a informação necessária para executar determinada ação. Estes tipos de sistemas são usados em aplicações comerciais uma vez que a dimensão do vocabulário é muito bem controlado e o erro produzido pelo reconhecimento é baixo (Schatzmann et al. (2006)).

Uma alternativa é o tipo de sistema de iniciativa mista, onde o sistema partilha o controlo do diálogo com o utilizador. Este tipo de estratégia é mais complexo uma vez que permite que o utilizador possa realizar respostas com mais informações. O sistema não se limita a reconhecer respostas diretas a perguntas que o mesmo colocou, em vez disso dada uma pergunta do sistema o utilizador poderá para além de responder a pergunta objetivamente inserir mais alguma informação que seja relevante. Sistemas mais recentes com a utilização

de redes neuronais (Schatzmann et al. (2006)) têm vindo a surgir de modo a conseguir criar uma forma prática de estratégia de diálogo.

Em McTear (2002) são apresentados três tipos de sistemas que contêm três métodos diferentes de controlo do dialogo com o utilizador:

- **Finite-state (or graph) based systems:** Este tipo de sistemas consistem numa sequencia pré definida de estados. O diálogo é realizado seguindo uma série de estados com transições que são reflexo de várias alternativas no grafo previamente construído. O controlo do diálogo é feito através do sistema, produzindo perguntas em cada estado do diálogo e produzindo ações baseadas nas respostas dos utilizadores. Estes tipos de sistemas estão muito presentes em atendimentos telefónicos, onde o atendimento passa por perguntar ao utilizador, por exemplo, qual a opção desejada e aguardar a devida resposta. A maior vantagem deste tipo de sistemas é que para cada estado a gramática pode ser especificada primeiramente, resultando num menor domínio do reconhecimento de voz e processamento de linguagem natural. A maior desvantagem é o facto de deste tipo de sistemas limitarem o input que o utilizador pode submeter no sistema.

- **Frame based systems:** Neste tipo de sistema são realizadas perguntas ao utilizador para o preenchimento de formulários, por exemplo numa pesquisa de voos. A grande diferença deste sistema para o *Finite-state (or graph) based systems* é conter algum tratamento de linguagem natural, dando uma maior flexibilidade a capacidade de resposta por parte do utilizador.

  Exemplo de um tipo deste diálogo:

  - Primeiro exemplo
    - Sistema: Qual o seu destino?
    - Utilizador: Portugal
    - Sistema: Em que dia pretende viajar?
    - Utilizador: Próxima quinta feira.
  - Segundo exemplo
    - Sistema: Qual o seu destino?
    - Utilizador: Portugal na próxima quinta feira.

No exemplo de diálogo acima como podemos observar o utilizador para a mesma pergunta inicial do sistema teve a possibilidade de responder de duas formas diferentes, sendo esta a maior diferença entre este tipo de sistema e o sistemas *Finite-state (or graph) based systems*.

- **Agent-based systems:** Este tipo de sistema são baseados em inteligência artificial para permitir diálogos complexos entre utilizador e o sistema (Schatzmann et al. (2006)). Exemplo de um diálogo deste tipo:

  * Utilizador: Procuro algum lugar para férias. Alguma sugestão?

  * Sistema: Portugal, mas durante a próxima semana irá estar muito calor. É do seu interesse?

  Pegando no exemplo acima o sistema em resposta a uma pergunta do utilizador não tende a fazer uma resposta simples, como "sim" ou "não", mas sim uma resposta mais construtiva de modo a preencher as necessidades do utilizador.

  O modelo de diálogo nestes sistemas tem em conta o contexto anterior com o resultado de que a conversação evolui dinamicamente como uma sequência de etapas relacionadas que se desenvolvem umas nas outras. Estes sistemas tendem a serem mistos no que toca a iniciativa, isto é, o utilizador pode tomar conta do diálogo introduzindo novos tópicos que podem não estar relacionadas com os contextos anteriores. Por estas razões estes sistemas não podem ser modelados de forma determinística com um conjunto de palavras, frases ou conceitos e na maior parte deste tipo de sistemas é necessário compreensão de linguagem natural ao nível semântico.

  Em suma estes sistemas permitem uma utilização de uma linguagem mais natural e mais idêntica a uma linguagem entre humanos e permite uma maior complexidade do âmbito do diálogo. Em contrapartida, estes sistemas são mais complexos de construir.

2.1.4 *Geração de Linguagem Natural*

A geração de linguagem natural serve dois papéis importantes. Em primeiro lugar, fornece uma resposta verbal às consultas do usuário, o que é essencial nas aplicações onde as respostas visuais não estão disponíveis. Além disso, ele pode fornecer feedback ao usuário na forma de uma expressão, confirmando a compreensão adequada do sistema da consulta de entrada. Embora tenha havido muita pesquisa sobre a geração de linguagem natural, lidando com a criação de parágrafos coerentes, o componente de geração de linguagem de

um sistema conversacional geralmente produz a resposta uma frase por vez, sem planeamento ao nível do parágrafo. Em casos mais simples estes sistemas limitam-se muitas vezes a recorrer a frases de saída que são simplesmente cadeias de palavras, em formato acústico de texto ou pré-gravado, que são invocadas quando apropriado. Em alguns casos, as frases são geradas pela utilização de templates aos quais é feito o preenchimento de slots. Apesar de serem mais simples os sistemas baseados em templates, estes são difíceis de manter e reutilizar, e as frases que produzem não possuem a variabilidade e a robustez necessárias aos sistemas de conversação. Em Klarner and Ludwig (2004) para colmatar o problema baseado em templates é proposto um sistema híbrido em que são utilizadas expressões fixas juntamente com texto gerado livremente. E mais recentemente têm vindo a surgir métodos utilizando redes neuronais.

Os autores em Tang et al. (2016) propuseram duas novas abordagens que codificam os contextos em uma representação semântica contínua e descodificam a representação semântica em sequências de texto com a utilização de RNN. Os autores propuseram duas abordagem para geração de texto de acordo com o contexto. O primeiro modelo *C2S* codifica um conjunto de contextos numa representação continua e depois descodifica para sequências de texto através de RNN com unidade LSTM. O primeiro modelo é capaz de gerar sequências de texto coerentes, no entanto, os autores referem que para casos mais longos a informação do contexto tende a desaparecer. Uma solução direta seria fazer uma dependência entre cada palavra com o contexto no entanto isto levanta um outro problema onde certas palavras não dependem do contexto mas apenas da(s) palavras anteriores. A solução para o problema encontrada pelos autores foi a utilização de uma *gating function* que dependendo do corrente *hidden state* faz a decisão de incluir ou não informação sobre o contexto. Os testes realizados foram feito num sistema de avaliação da *Amazon* onde o texto gerado foi 50% mal avaliado por júris e 90% mal avaliado por sistemas de deteção de avaliações falsas.

Em Wen et al. (2015) é apresentado uma proposta de uma rede neuronal recorrente (RNN) capaz de gerar variadas resposta numa profunda e semanticamente controlada arquitetura LSTM. O gerador pode ser treinado com dados não alinhados juntando o planeamento da sentença e realização superficial usando o critério *cross entropy*.

2.1.5 *Texto para Som*

O módulo de transformação de texto para som consiste em dar uma forma audível ao conteúdo que foi previamente gerado pelo módulo de geração de linguagem natural no contexto de sistemas de conversação por voz. Para sistemas mais simples poderá ser uti-

lizado falas pré gravadas onde o som final produzido é resultado do preenchimentos de espaços. Este método funciona bem para casos onde as resposta exigidas são estáticas, no entanto em muito dos casos necessitamos de gerar respostas mais complexas o que faz com que o texto a ser audível é variável e imprevisível, daí a necessidade de criar uma forma eficaz de gerar som através de texto.

O processo de transformar texto para som pode ser composto em dois problemas:

1. Análise do Texto
2. Síntese de Voz

A análise do texto envolve dar uma representação linguística ao texto de entrada para que o mesmo possa ser sintetizado em um som audível representando o conteúdo do texto de entrada. Esta tarefa envolve essencialmente mais quatro sub tarefas, segmentação e normalização do texto, análise morfológica, análise sintática e modelação de efeitos contínuos que ocorrem na fala (McTear (2002)). A segmentação tem o papel de separar o texto em unidade menores, tais como parágrafos, palavras, caracteres. A normalização envolve a interpretação de abreviações e outras formas normais de representações de datas, para que símbolo presentes no texto de entrada possam serem convertidos para uma representação que possa ser prenunciada. A análise morfológica para lidar com a pronúncia de um número alargado de palavras que são morfologicamente variantes de outras. A análise sintática ajuda o sistema a perceber a correta pronúncia de determinada palavras dentro de uma frase, uma vez que existem palavras em que a sua pronúncia é dependente da sintaxe da palavras (se é verbo, pronome entre outras). A modelação de fala continua deve-se ao facto de existirem palavras próximas onde ocorre a oclusão de certas fonemas inerentemente.

A síntese de voz corresponde à criação da componente audível resultante deste módulo. Este deve ser capaz de gerar uma forma audível ao texto dado como input de uma forma mais natural possível, isto envolve o correto processamento do texto de entrada e também uma correta sequências de unidades fonéticas (Mache et al. (2015)). Existem diferentes técnicas que têm sido desenvolvidas ao longo dos anos (Balyan et al. (2013), Kayte (2015)), podendo se destacar as seguintes: *Articulatory Synthesis*, *Formant Synthesis*, *Concatenative Synthesis* e *Hidden Markov Models based Synthesis*.

Articulatory Synthesis é das técnicas de som mais complexas. Esta técnica é baseada em modelos da produção de sons por parte do ser humano. Envolve a simulação de acústicas do trato vocal e suas movimentações dinâmicas. A *Articulatory model* reconstitui a forma do trato vocal como uma função de posições do órgãos fonatórios (lábios, maxilar, língua).

O sinal é calculado com simulações matemáticas do fluxo de ar através do trato vocal. O problemas encontrado nesta técnica é a complexidade de obter um modelo tridimensional preciso que represente o trato vocal.

A técnica *Formant Synthesis* modela o trato vocal simulando os frequências formantes (frequências ressonantes). Formantes podem ser definidos como picos de energia numa região do espectro sonoro. Este método usa o modelo de filtro-fonte (*source-filter*) de produção de fala, o que significa que a ideia é a geração de fontes de sinal periódicos e não periódicos e alimenta-los através de um circuito ressonante (filtro) que modela o trato vocal. Cada fonema contem um grupo de picos de energia Este método apresenta-se bastante flexível necessitando da utilização de poucos recurso para a produção do som. No entanto o discurso produzido não é muito natural.

A técnica *Concatenative Synthesis* (A (2016)) é das técnicas menos complicadas para a síntese de voz. O processo dá-se através da concatenação de diferentes falas pré gravadas e guardas em base de dados, que podem ser sentenças, palavras, silabas, trifone. No entanto as diferenças entre a variação natural da fala e a natureza de técnicas de segmentação do sinal pode resultar em erros audíveis. Existem essencialmente alguns sub tipos de *Concatenative Synthesis* podendo se destacar: *Unit Selection Synthesis*, *Di-Phone Based Speech Synthesis* e *Domain Specific Synthesis*. *Unit Selection Synthesis* usa uma grande base de dados. Durante a criação do banco de dados, cada enunciado gravado é segmentado em alguns fonemas individuais, sílabas, morfemas, palavras, frases e sentenças. A organização dos dados na base de dados é então feito com base na segmentação e nos parâmetros acústicos, como frequência fundamental, altura, duração, status da sílaba e fonemas anteriores e próximos. Esse método fornece naturalidade na fala de saída em comparação com outras técnicas. Em *Diphone synthesis* usa uma base de dados limitada de transições de fonema, e a fala sintetizada é concatenada diretamente desta base de dados. Os recursos prosódicos são então adicionados em um pós-processamento do sinal. Por fim *Domain Specific Synthesis* concatena palavras e frases pré-gravadas para criar enunciações completas. A síntese por domínio específico é aplicada quando a variedade de textos que o sistema produzirá é limitada a um domínio específico de negócio. O nível de naturalidade destes sistemas pode ser muito alto porque a variedade de tipos de sentença é limitada.

O método estocástico *Hidden Markov Models based Synthesis* (Kayte et al. (2015)) A abordagem HMM (Ze et al. (2013)) vem resolver alguns problemas encontrados no método baseado na concatenação: a criação de banco de dados demorada e a impossibilidade de criar algo além do que foi gravado. Usando uma abordagem estocástica, duas vantagens aparecem: menos memória para armazenar o parâmetro do modelo e mais variações permitidas.

Mais recentemente algoritmos de *Deep Leaning* têm vindo mostrado uma enorme capacidade de aprendizagem baseada nos dados, conseguindo captar características mais facilmente do que o humano. Deste modo os métodos de *Deep Leaning* não necessitam a modelagem de características por parte do humano tendo a capacidade, no contexto de síntese de voz, de aprender mais facilmente a associação entre um texto de entrada e o som representativo desse mesmo texto. O modelo *Char2Wav* apresentado por Sotelo et al. (2017) é composto por um *reader* e *neural vocoder*. O *reader* utiliza redes bidirecionais RNN que aceita texto como input e RNN para gerar propriedades acústicas. Por fim uma extensão de um rede *SampleRNN* é utilizada para a geração do áudio final.

Em Wang et al. (2017) é apresentado um sistema em que dada um sequência de caracteres é dado como output do sistema um áudio correspondente a transcrição do texto para linguagem audível. Nestes sistemas existe a grande vantagem de não necessitarem de dados complexos e a rede apresentada pode ser treinada dado uma sequência de caracteres e o respetivos ficheiro de áudio.

Outro trabalho é apresentado em O. Arik et al. (2017) é exposta uma rede neuronal que é uma adaptação da *WaveNet* (rede neuronal profunda para a geração de áudio).

2.1.6 *Sistemas Externos*

Na maior parte das vezes os sistemas de conversação por voz necessitam de acesso a dados exteriores para ajuda ao sistema numa situação inesperadas seja para busca de informação pedida pelo utilizador ou tentativa de resolução de algum erro. Este tipo de acesso a componentes externas do sistema é realizado através do modulo de gestor do diálogo que pode ser dos seguintes tipos (McTear (2002)):

- **Comunicação com a base de dados:** Tomando como exemplo um cenário em que o utilizador queira saber informação relativa a um voo, como a data de partida e chegada, o sistema deve ser capaz de procurar essa informação numa base de dados por forma a satisfazer as necessidades do utilizador.
- **Comunicação com uma base de conhecimento:** Este tipo de sistema são usado na resolução de problemas aos quais o utilizador necessita de algum apoio especializado.
- **Comunicação com um sistema de planeamento:** Este tipo de sistema também eles podem ser para resolver problemas baseado em objetivos, planos e ações.

2.2 CONTROLO DO COMPUTADOR USANDO A VOZ

Ao longo dos últimos anos foram propostos vários sistemas, sendo alguns de pesquisa e alguns produtos comerciais, para permitir controlar o cursor do rato recorrendo à voz humana.

Em Igarashi and Hughes (2001), os autores propuseram várias técnicas para o uso de funções de voz não-verbais. Recorrendo há duração da expressão, por exemplo, quando o utilizador pronúncia *"Volume mais alto, ahhhh"* o volume irá aumentar enquanto o utilizador continue a pronunciar *"ahhhh"*. Também é proposto recorrendo ao tom, por exemplo aplicando esta técnica na navegação de uma página web quando o utilizador pretende fazer *scroll* este prenuncia *"para baixo, ahhhh"* e consoante o tom do *"ahhhh"* a página deslocaria-se mais rápida ou mais lenta. Por fim é apresentada uma técnica recorrendo à frequência do som discreta, permitindo deste modo, por exemplo, na mudança de canal de uma televisão pode-se proferir *"Canal seguinte, ta ta ta"* sendo que o sistema neste caso mudaria para o canal seguinte até que o utilizador terminasse de produzir o som *"ta"*, o diferencial neste caso era que o utilizador não precisava de pronunciar o sinal através da voz, este funcionaria de igual forma se o mesmo fosse um estalar de dedos por exemplo.

Em Yoshiyuki Mihara and Shibayama (2004) é proposto um sistema em que sempre que o utilizador pretenda mover o cursor no ecrã é apresentado múltiplos cursores fantasma alinhados verticalmente e horizontalmente. Na figura 3 está representado o sistema num contexto de utilização onde observamos que quando o utilizador pretende mover o cursor até à bola amarela este começa por dizer *"move right"*. De seguida são apresentados vários cursores fantasma, sendo que o utilizador terá agora de especificar um número correspondente ao cursor fantasma destino. Por fim como o mesmo pretende mover para cima (destino é a bola amarela) este acaba pode dizer *"move up three, click"* e o cursor sobe três posições na vertical e por fim clica na bola amarela. O mesmo resultado que é apresentado na figura 3 poderia ser simplificado bastando para isso que o utilizador especificasse as coordenas destino neste caso, sete na horizontal e três na vertical.

Uma das técnicas mais usadas é apresentado em Haque et al. (2013) que se apresenta como uma melhoria ao método baseado em grid tradicional, que tipicamente é constituído por uma grid de 3×3. Em um sistema grid típico de 3×3 como ilustrado na figura 4 a sua utilização é bastante simples e bastante intuitiva. Partindo de que o utilizador pretende deslocar o cursor do rato para a posição marcada com uma bola vermelha na figura 4, o utilizador teria de dizer posição três, posição onde é gerada uma nova grid recursivamente, depois posição cinco e por fim clique. O proposto em Haque et al. (2013) é

2 Imagem adaptada de Yoshiyuki Mihara and Shibayama (2004)

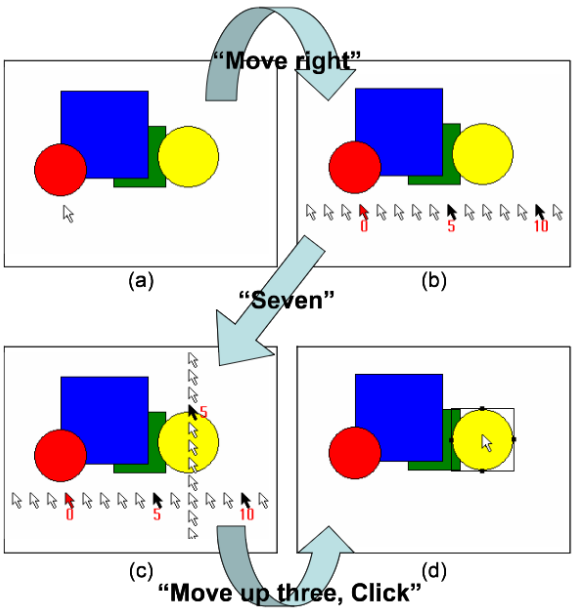


Figure 3: Exemplo de comando e movimentação do cursor [2]

adotar uma grid dinâmica em que o utilizador poderia definir o tamanho da grid consoante a sua preferência, sendo possível ter grid 3×3 ou 4×4. Contudo, os autores do artigo Haque et al. (2013) fazem referência para o facto de apesar de ser possível redimensionar o tamanho da grid é necessário ter em consideração um fator importante relativo a sistemas de conversação Homem-Máquina referente ao reconhecimento de voz. Considerando um sistema modelado por uma grid 4×4 as palavras *"seis"* e *"dezasseis* podem muito bem serem confundidos no reconhecimento de voz e deste modo o sistema fica impreciso.

Uma outra forma de controlo do cursor utilizando a voz é apresentado em Harada et al. (2006), que usa diferentes sons de vogais associados a cada uma das direções do movimento do cursor. Neste sistema, o usuário pronuncia um som de vogal correspondente ao desejado das quatro direções (”a” para cima, ”e” para direita, ”i” para baixo e ”o” para esquerda) e o cursor começa movendo-se na direção prenunciada. Uma vez que o cursor dá inicio ao seu movimento, ele é governado pelo ”movimento da inércia” e o usuário não necessita continuar vocalizando. A velocidade do cursor começa inicialmente lenta e acelera gradualmente com o tempo. O cursor é interrompido ao prover novamente o mesmo som da vogal e o clicar é executado proferindo um comando de duas vogais (”a-e”). Neste sistema tentam limitar o problema de delay que é imposto quando utilizamos o reconhecimento de voz, que se deve ao facto da ação apenas pode ser dada quando o reconhecimento de voz termina a sua tarefa. Este sistema para evitar tal delay, este sempre que o utilizador

3 Imagem adaptada de Haque et al. (2013)

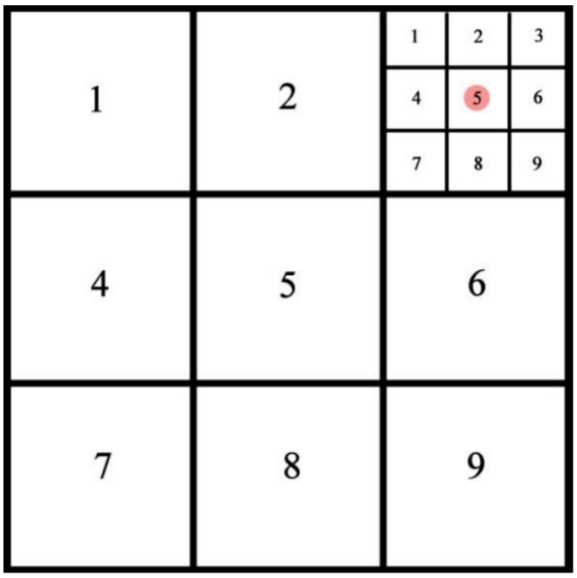


Figure 4: Modelo grid 3×3 [3]

começa a produzir o som de uma vogal o cursor começa imediatamente o seu movimento. Neste sistema é possível controlar a velocidade de deslocamento do cursor.

Em B. Manaris and MacGyvers (2001) é apresentada uma interface, *SUITEKeys*, que permite que o utilizador consiga controlar totalmente o computador através da voz, isto implica controlar o rato e teclado. Para a movimentação do rato o utilizador teria de dizer por exemplo *"move para a direita"* e o rato moveria nesse sentido com uma velocidade constante. Quando o rato estava na posição desejada pelo utilizador para que o fizesse para bastava apenas dizer por exemplo *"stop"*. O sistema também dividia o ecrã por cem unidades tanto na vertical como na horizontal (formando uma matriz 2D) para permitir que o utilizador pudesse especificar comando do tipo *"move para a direita 10 unidades"*. Para além da movimentação do cursor no computador este sistema possui a capacidade de executar ações de clique no botão do lado direito do rato e arrastar objetos, tudo isto recorrendo apenas a reconhecimento de voz. Neste sistema também inclui o controlo do teclado, sendo que neste caso apenas o utilizador falava a letra do teclado que queria que fosse pressionado e o sistema processa a voz e executa a ação. Para além de tudo isto este sistema quando por exemplo cria um documento no *Word* faz previsão das palavras que o utilizador pretende dizer com base num histórico que é feito pelo próprio *software*.

Um dos software mais usados no reconhecimento de voz e controlo do computador por voz é o *Dragon NaturallySpeaking* que oferecem vário métodos diferentes de controlo do cursor através da voz. Um desses métodos é o *MouseGrid*, e um outro baseado em um cursor

de velocidade constante semelhante ao *SUITEKeys*. Por exemplo, o usuário diria ”mover o rato para cima” e o cursor começaria a mover-se para cima em uma velocidade fixa (o padrão é aproximadamente 4 pixels por segundo). O usuário pode então emitir comandos para alterar a velocidade (por exemplo, ”muito mais rápido”), direção (por exemplo, ”esquerda”), para parar o movimento (”parar”) ou para clicar no botão do rato (”clique”). Existem três níveis de comandos para alterar a velocidade do cursor (”mais rápido”, ”muito rápido” e ”muito mais rápido”, e os correspondentes para desaceleração). O movimento do cursor também é brusco, atualizando sua posição aproximadamente quatro vezes por segundo, ignorando assim um número de pixels quando a velocidade é maior do que o padrão.

2.3 SUMÁRIO

Ao longo dos anos tem vindo a existir um interesse crescente no desenvolvimento e aperfeiçoamento de assistentes virtuais. Correntemente vemos este tipo de sistema nos automóveis, telemóveis e computadores. Os assistentes virtuais como se verficou engloba um conjunto de conceitos com as sua dificuldades e complexidades, desde o reconhecimento de voz, processamento de linguagem natural, geração de linguagem naural e sintese de fala para além da parte central destes sistemas, o gestor do discurso.

Atualmente o uso de linguagem natural como uma forma de controlar todos os aspectos do computador tem vindo a ser desenvolvida, apesar de que a um ritmo lento. Temos vindo a ver o *Siri* no *Mac os* e o *Cortana* da *Microsoft* que apesar de não ser focado no controlo do computador em si, já apresentam funcionalidades que no futuro puderam se expandir, por forma a que o utilizador não necessite de periféricos para controlar o seu sistema. No entanto atualmente existe falta de ferramentas universais, que independente do sistema operativo consigam controlar todos os aspectos do mesmo.

3

PLATAFORMAS DE PROCESSAMENTO DE LINGUAGEM NATURAL

Durante a realização do protótipo da aplicação foi necessário fazer uma análise prévia a diferentes soluções que incorporassem a componente de reconhecimento de voz e também o processamento de linguagem natural. No secção 3.1 é realizada uma levantamento de alguns conceitos necessário para a estruturação de um chatbot nos devidos sistemas de processamento de linguagem natural. Na secção 3.2 é apresentado algumas plataformas de processamento de linguagem natural e reconhecimentos de voz. E por fim, na secção 3.3 e secção 3.4 é exposto alguns testes de performance do módulo de processamento de linguagem natural e do módulo de reconhecimento de voz respetivamente.

3.1 CONCEITOS PARA A CONSTRUÇÃO DE UM CHATBOT

Como foi referido anteriormente, um chatbot é um sistema conversacional, que através da utilização de linguagem natural, é capaz de interpretar o que foi dito pelo utilizador e retirar daí a suposta intenção que o utilizador pretende com a sua sentença. Com este objetivo é necessário fornecer previamente conhecimento no sistema, sendo para isso necessária a introdução de frases exemplo no sistema e respetivas ações. A partir deste conhecimento introdutório é possível o chatbot aprender, permitindo interpretar não só frases como as que foram previmanete introduzidas no sistema como suas variantes.

A estruturação do conhecimento num sistema de chatbot é fundamentado numa séria de conceitos, podendo se destacar o conceito de *"Expression"*, *"Intents"* e *"Entities*. Uma *"Expression"* é uma frase do orador submetida ao sistema em linguagem natural. A *"Intents"* são propósitos ou objetivos expressos pelo orador para alcançar determinado objetivo. Por fim temos as *"Entitites"* que se traduz em algum valor que está associado com alguma *"Intent"*. Essencialmente através dos conceitos apresentados acima é possível fornecer ao

sistema de chatbot a possibilidade de entender e interpretar qual a intenção do utilizador e a partir da análise determinar a ação que o utilizador deseja.

Para uma melhor perceção destes conceitos tomemos como exemplo a seguinte *"Expression" "Apagar a lâmpada"*. Uma correta definição dos dados num simples sistema de chatbot seria definir a *"Intent"* como sendo a *"lâmpada"* e as *"Entitites"* associada ao *"Intent" "lâmpada"* poderiam ser *"Acender"* ou *"Apagar"*. Deste modo o sistema de chatbot dada a expressão *"Apagar lâmpada"* ou *"Acender lâmpada"*, e suas derivadas, seria capaz de perceber que a ação que o utilizador pretender exprimir é de apagar ou acender a lâmpada.

3.2 Análise de plataformas para construção de assistentes virtuais

Para proceder a construção da aplicação foi necessário realizar uma pesquisa para perceber que plataformas hoje se encontram disponível e quais as que apresentam melhores resultados quando pretendemos proceder a construção de um chatbot. Com este objetivo em mente, um aspeto importante seria que a plataforma que apresentasse uma elevada eficácia juntamente com uma baixa latência e que apresentasse um custo reduzido por forma a ser uma plataforma acessível para utilização.

Atualmente existem essencialmente quatro plataforma que apresentam uma maior popularidade no contexto de assistentes virtuais e criação de chatbot, sendo estas: *Wit.ai*, *Dialogflow*, *IBM Watson* e *LUIS Microsoft*.

A plataforma *Wit.ai* pertence ao *Facebook* e apresenta um plano gratuito[1] para a sua utilização, contendo para além do processamento de linguagem natural também a componente de reconhecimento de voz. O acesso a esta plataforma é oferecido através da *Application Programming Interface (API)*[2] disponibilizada. As ligações a esta API é feita utilizando o protocolo *Hyper Text Transfer Protocol Secure (HTTPS)*, onde é possível o envio de ficheiros de áudio ou simples mensagens de texto para que o mesmo seja analisado e processado. Depois de feito o processamento a plataforma retorna o resultado em formato *JavaScript Object Notation (JSON)* contento a *"Intent"* juntamente com as *"Entities"*, caso existam, e o texto sobre o qual foi realizado o processamento. Este texto como referido anteriormente tanto poderá corresponder a texto introduzido ou a uma transcrição do áudio que foi enviado para o sistema.

1 https://wit.ai/faq
2 https://wit.ai/docs/http/

A plataforma *Dialogflow* atualmente apresenta um plano também ele gratuito[3] com a limitação para chamadas com querys de texto de 180 pedidos por minuto e a componente de reconhecimento de voz com no máximo um minuto de tamanho de áudio e limitado a 15000 pedidos de áudio por mês. O *Dialogflow* é propriedade da *Google* fazendo por isso utilização dos serviços da mesma como o *Google Cloud* e o *Google Cloud Speech-to-Text* para o reconhecimento de voz. O funcionamento desta plataforma é muito idêntico à plataforma do *Wit.ai* onde para o seu acesso é feito através da API[4] disponibilizada.

A plataforma *IBM Watson* também apresenta um plano gratuito mensal de 10000 mensagens de texto[5] por mês e 100 minutos por mês para pedidos de reconhecimento de voz[6]. Esta plataforma é uma das mais conceituadas plataforma para o processamento de linguagem natural, contudo o aspeto mais negativo é o reconhecimento de voz apenas ser possivél para a lingua português do Brasil. O pedido a esta plataforma segue a mesma ideia das anteriores onde os pedidos são realizadas através da API[7] disponibilizada.

A plataforma *LUIS Microsoft* também tal coma as anteriores apresenta no seu catálogo um plano gratuito mensal[8]. Para o processamento de linguagem natural a plataforma encontra-se limitada a 10000 pedidos de texto, não podendo ser realizadas mais do que 5 transações por segundo. No que toca ao reconhecimento de voz o serviço oferece cinco horas de áudio gratuitamente por mês[9]. Tal como acontece com as restantes plataforma apresentadas, os pedidos são realizados através da utilização da API[10] disponibilizada.

3.3 Testes de Eficácia das Plataforma de Processamento de Linguagem Natural

Depois de identificadas as plataforma que atualmente são mais populares para construção de chatbots, foi necessário fazer uma série de testes para verificar os seus níveis de performance. Para a medição da performance foi então necessário a construção de um pequeno dataset com algumas frases divididas em diferentes categorias, sendo que cada categoria representa um possível ação no sistema. A este dataset foram também adicionadas algumas

3 https://cloud.google.com/dialogflow-enterprise/quotas
4 https://dialogflow.com/docs/reference/api-v2/rest
5 https://www.ibm.com/cloud/watson-assistant/pricing/
6 https://www.ibm.com/cloud/watson-speech-to-text/pricing
7 https://cloud.ibm.com/apidocs/assistant-v2
8 https://azure.microsoft.com/en-us/pricing/details/cognitive-services/language-understanding-intelligent-services/
9 https://azure.microsoft.com/en-us/pricing/details/cognitive-services/speech-services/
10 https://westus.dev.cognitive.microsoft.com/docs/services/5890b47c39e2bb17b84a55ff/operations/5890b47c39e2bb052c5b9c2f

frases aleatórias que não tivessem um representação de ação no computador. Foram então feitos três testes com diferentes níveis no número que era utilizado para os casos de treino como para casos de teste. Para todos os testes foram realizadas três medições e escolhida a média dessa medições. Para o cálculo da eficácia do sistema apenas foi considerado os valores da *"Intent"*, por se apresentar o aspeto da resposta mais crítico, uma vez que é o responsável por indicar ao sistema qual ação a realizar. O cálculo da eficácia do sistema é dado pela fórmula 1 onde $TotalHits$ é um valor que é somado sempre que a plataforma devolva a *"Intent"* esperada e a sua confiança seja maior do que o nível mínimo estipulado, que será apresentado mais à frente. O $TotalSamples$ é o numero total de casos de testes que foram enviados para a plataforma de processamento de linguagem natural. Para além da eficácia do sistema, também é importante saber a percentagem de falsos positivos, ou seja, a percentagem que o sistema devolve uma ação com uma confiança maior que o limite mínimo, mas na qual não representa a ação desejada do utilizador. A fórmula utilizada para o cálculo deste valor é a representada igualmente pela fórmula 1 onde neste caso o $TotalHits$ é somado apenas quando o valor da *"Intent"* devolvida pelo sistema não é igual à esperada, mas apresenta uma confiança maior do que o valor mínimo estipulado. Igualmente como acontece com o teste de eficácia o valor do $TotalSamples$ representa o número total de samples que foram enviadas para o sistema.

O primeiro teste foi realizado com a utilização de 80% de casos para teste e 20% de casos para treino e os valores obtidos são o que se encontram representados na figura 5. O segundo teste foi realizado com a utilização de 50% de casos para teste e 50% de casos para treino e os valores obtidos são os que se encontram na figura 7. O terceiro e último teste foi realizado com a utilização de 20% de casos para teste e 80% de casos para treino e os valores obtidos são os que se encontram na figura 9.

Os testes permitem desde logo perceber algo que já se estaria à espera, em todas as plataformas à medida que aumentamos o número de casos de treino todas as plataformas analisadas tendem a aumentar a sua eficácia. Na transição do segundo teste para o terceiro a plataforma do *Wit.ai* tem uma eficácia mais baixa quando aceitamos testes com no mínimo 70% de confiança, este facto pode ser explicado essencialmente as frases que foram selecionada para o teste que possivelmente fizeram com que o *Wit.ai* não fizesse uma aprendizagem tão eficaz o que resultou numa pequena baixa na sua taxa de acerto, mas no entanto aumentou a percentagem para os casos onde se exige uma confiança maior.

Para o casos de falsos positivos, as samples usadas foram as mesmas aquando do calculo da eficácia. O primeiro teste foi então realizado com 80% de casos para teste e 20% de casos para treino sendo o valores obtido representados na figura 6. O segundo teste foi feito com a utilização de 50% de casos de teste e 50% de casos de treino e os valores obtidos são

os que se encontram representados na figura 8. O terceiro e último teste foi feito com a utilização de 20% para casos de teste e 80% para casos de treino e os valores obtidos são os que se encontram representados na figura 10. Este testes permitem observar um aspeto diferente do valor da eficácia onde à medida que aumentamos o número de casos de treino este valor também tende aumentar apesar de não ser muito significativamente. Este facto pode ser explicado uma vez que à medida que aumentamos o número de casos de treino o sistema apreende a fazer um maior número de conexões, ou seja, para cada sentença tende sempre a dar uma resposta. Também pode ser explicado pelo facto no dataset existirem frases similares como por exemplo: ”mover o rato para cima” e ”mover rato para baixo”. Estes exemplos são divididos em duas categorias distintas, movimentação do cursor para cima e para baixo respetivamente. Uma vez que a escolha das samples para treino e testes das diferentes categorias é feita de forma aleatória, pode acontecer que o sistema numa categoria selecionar a frase ”mover o rato para cima” para treino e a frase ”mover rato para baixo” como teste. Quando isto acontece a plataforma tenta arranjar uma conexão e oferece um resultado errado dando que no teste com a frase ”mover rato para baixo” o sistema interpreta a ação de mover o rato para cima.

O principal objetivo da utilização de diferentes taxas para casos de uso, ajudou a perceber para cada sistema qual um número mínimo que seria necessário para que o sistema começasse a responder de forma satisfatória. Este dado é importante pois, no desenrolar no protótipo será dada a possibilidade do utilizador criar as sua próprias ações, isto implica o mesmo referir um conjunto de frases para essas ações por forma a treinar o modelo de linguagem natural com mais informação. Caso optasse-mos por um sistema onde seria necessário uma enorme quantidade de dados o utilizador obrigatoriamente teria de oferecer ao sistema de processamento de linguagem natural uma quantidade com a mesma proporção, o que não seria aceitável.

Em suma, observamos que a plataforma da *IBM Watson* é que para um mesmo número samples apresenta uma maior taxa de eficácia, contudo encontra-se praticamente ao nível da plataform *Wit.ai*. Para além disto uma facto importante é a rapidez com que a plataforma apresenta os resultados e neste aspecto o *Wit.ai* claramente leva a melhor apresentando uma valor bem mais baixo do que as suas concorrentes (dados na tabela 1). Um facto interessante é o da plataforma *Dialogflow* que apesar de em todos os seus teste não apresentar qualquer tipo de falsos positivos tende a necessitar de um elevado número de casos para que comece a apresentar resultados.

$$Am = \frac{TotalHits}{TotalSamples} \tag{1}$$

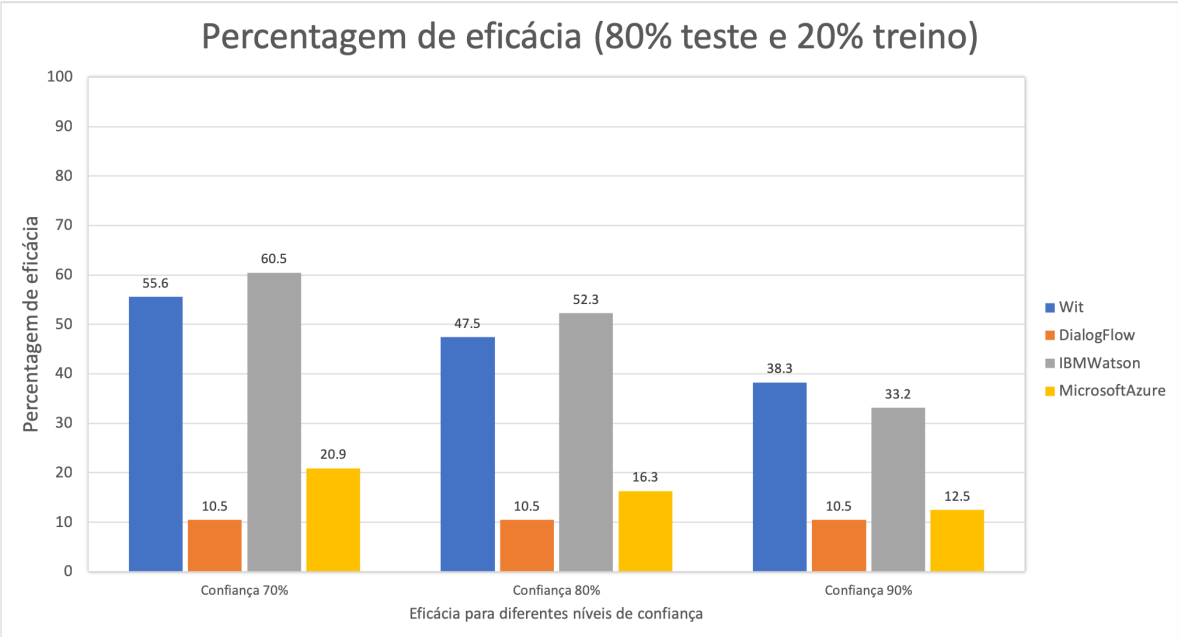


Figure 5: Teste de eficácia com 80% dos dados para teste e 20% para treino

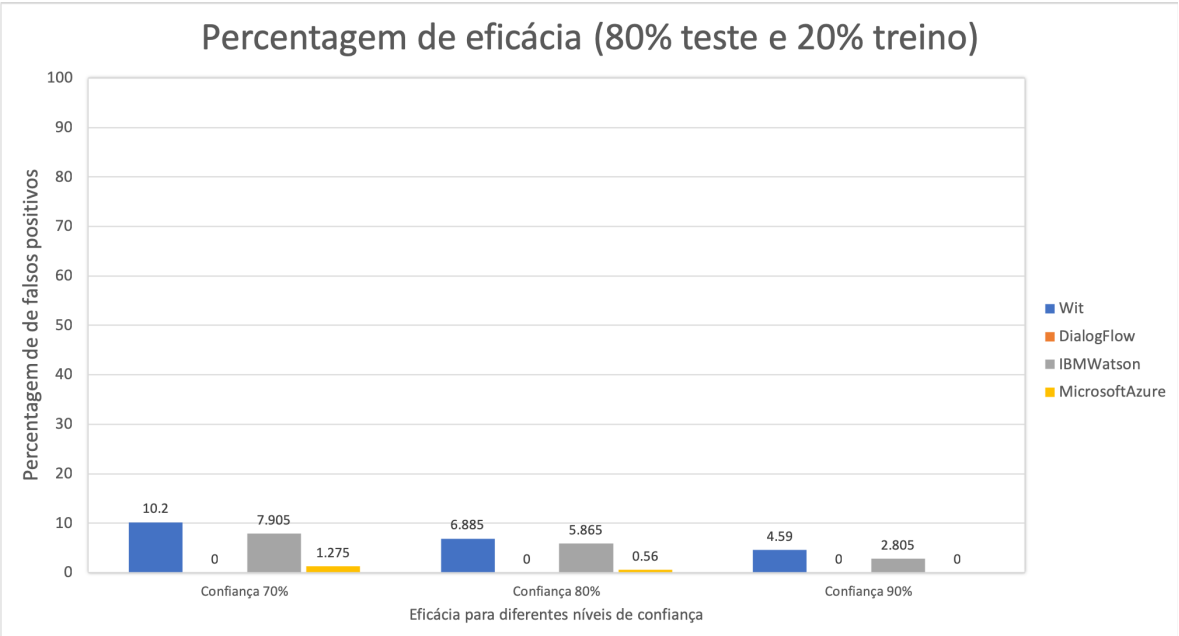


Figure 6: Teste de falsos positivos com 80% dos dados para teste e 20% para treino

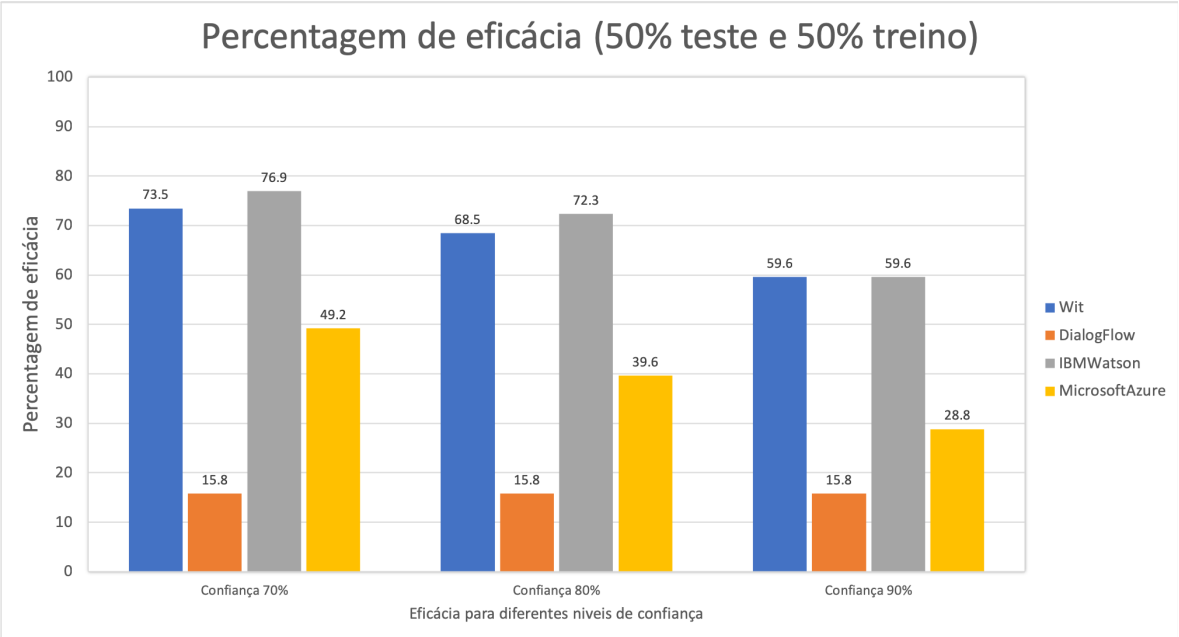


Figure 7: Teste de eficácia com 50% dos dados para teste e 50% para treino

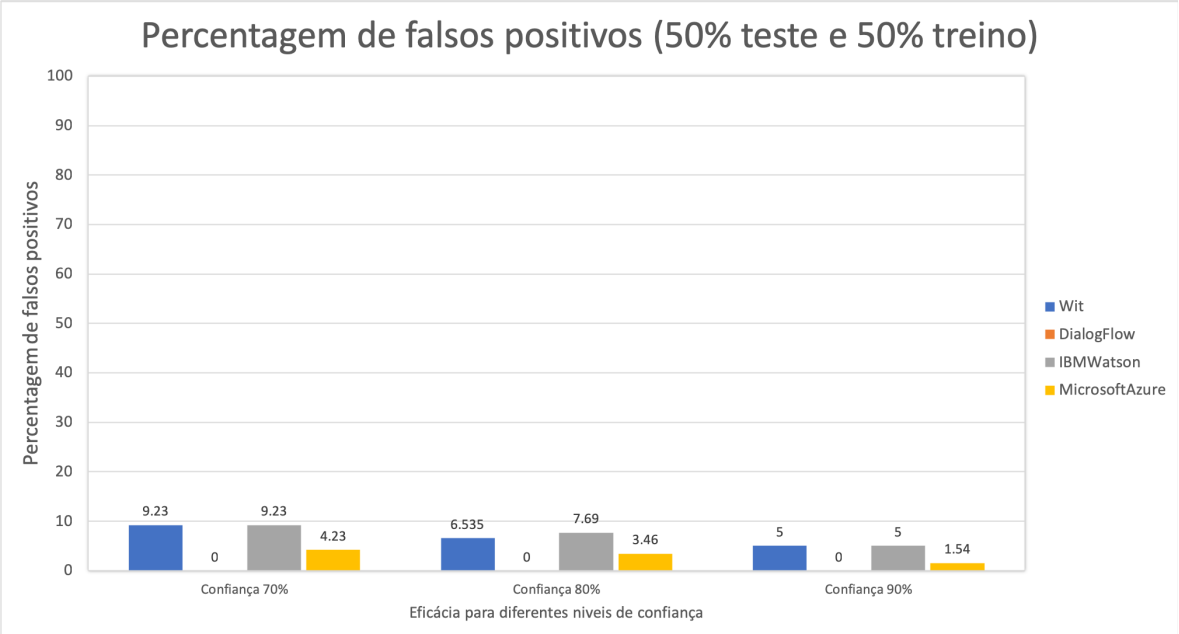


Figure 8: Teste de falsos positivos com 50% dos dados para teste e 50% para treino

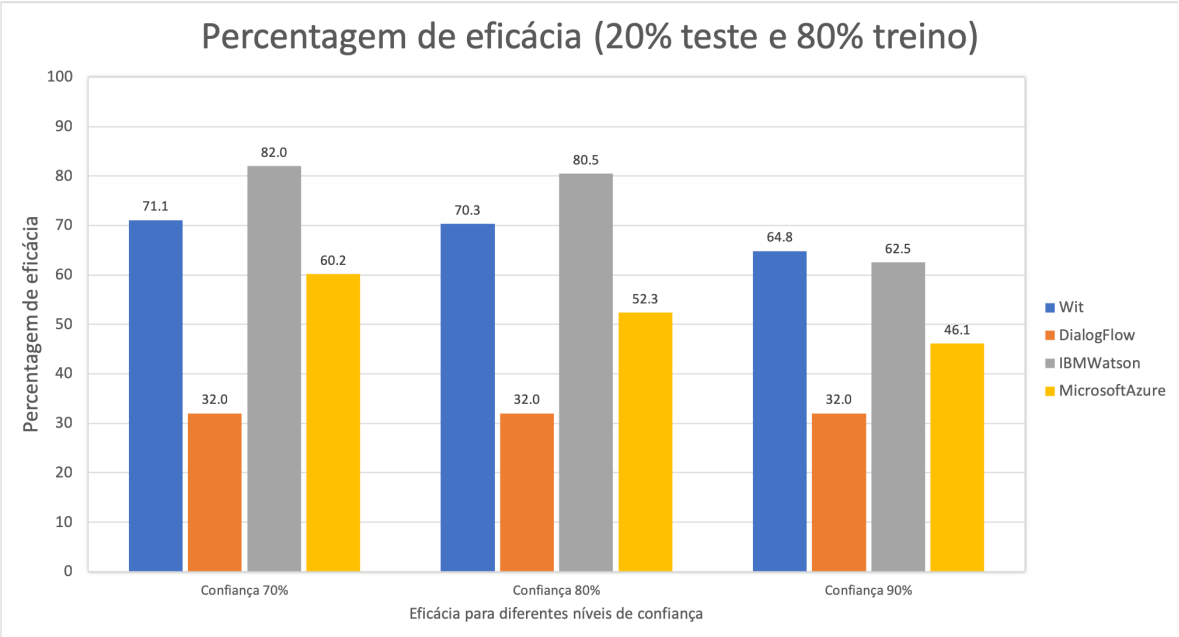


Figure 9: Teste de eficácia com 20% dos dados para teste e 80% para treino

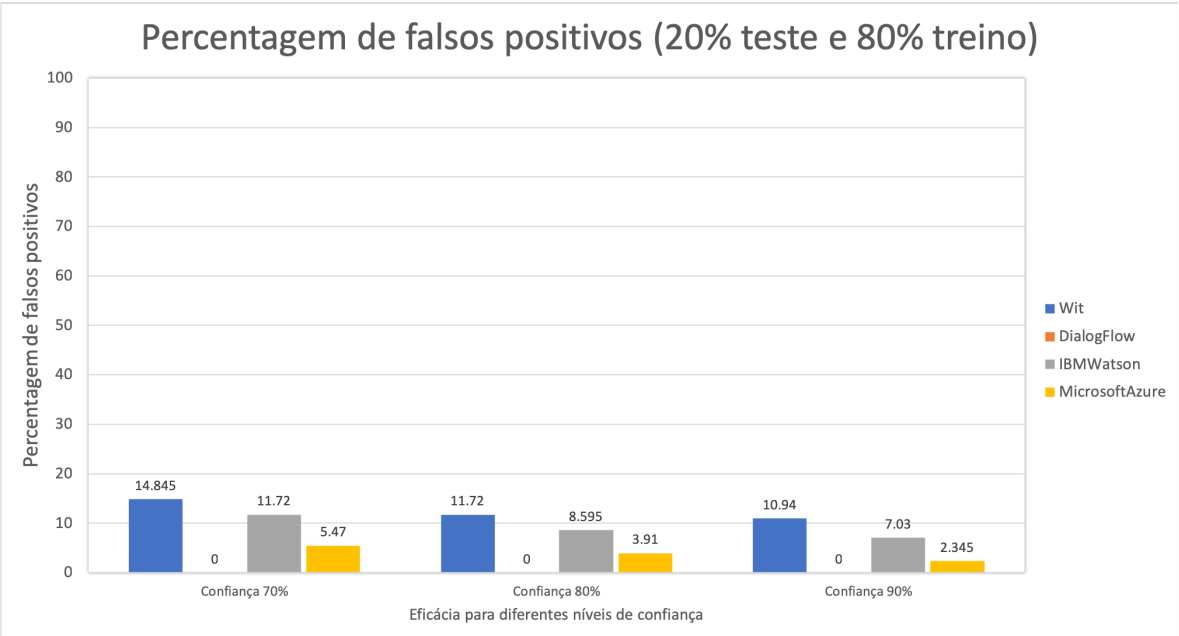


Figure 10: Teste de falsos positivos com 20% dos dados para teste e 80% para treino

Sistema	Tempo (milissegundos)
Wit.ai	160,77
Dialogflow	192,45
IBM Watson	217,20
LUIS Microsoft	189,94

Table 1: Tempo de execução das plataformas de processamento de linguagem natural

3.4 TESTES AO RECONHECIMENTO DE VOZ

Para a análise desta componente do sistema foi criada uma pequena base de dados com frase típicas para o controlo do computador. Para o teste foi criado um script que envia um ficheiro de áudio para o servidor e com o resultado obtido da plataforma que se encontrava a ser testada calcula da taxa de erro da palavra (*Word Error Rate (WER)*), que é a métrica tipicamente usada para avaliação da precisão de sistemas de reconhecimento de voz que é dada pela fórmula apresentada em 2 onde S é o número de substituições, D o número de eliminações, I número de inserções e N o número de palavras na frase.

$$WER = \frac{S + D + I}{N} \tag{2}$$

Para o teste avaliou-se qual a distância entre a frase que é dita no ficheiro de áudio com o que foi reconhecido. Este teste foi feito com áudio stereo a um sample rate de 16Khz. Das plataformas anunciadas acima apenas três delas apresentam a possibilidade de reconhecimento de voz na linguagem portuguesa que é o *Wit.ai*, *Dialogflow* e *Microsoft Azure* as restantes, *IBM Watson*, apresentam apenas a possibilidade de tradução para Português do Brasil pelo que a análise apenas recairá sobre a plataforma *Wit.ai*, *Dialogflow* e *Microsoft Azure*. Na tabela 2 é apresentado para cada plataforma qual o WER e o tempo que de processamento da transcrição da voz para texto, na medida em que é crucial tempos de execução neste tipo de sistemas.

Sistema	Word Error Rate	Tempo (milissegundos)
Wit.ai	14,48	3029,67
Dialogflow	15,03	1932,83
IBM Watson	52,08	2446,26
LUIS Microsoft	45,38	2231,03

Table 2: Taxa de erro da palavra mais tempos de resposta

3.5 SUMÁRIO

Como se pode verificar das plataformas analisada, o *Wit.ai* é a que atualmente apresenta um melhor custo benifício oferecendo uma boa performance comparativamente às sua concorrentes. É a plataforma que com uma menor taxa de casos de teste, consegue obter uma melhor aprendizagem. Quanto ao reconhecimento de voz verificamos que tanto o oferecido pelo *Wit.ai* e *Dialogflow* serem bastante próximos em termos de eficácia, o *Wit.ai* é demasiado lento, em termos de processamento, quando pretendemos usá-lo como forma de controlar o cursor com voz, pelo que de todos os sistemas analisados o *Dialogflow* é o mais rápido e o segundo que apresenta uma taxa de eficácia mais elevada.

4

CRIAÇÃO DA APLICAÇÃO *NATURALASSISTANT*

Neste capítulo é feita uma análise ao sistema que foi construido com o propósito de atender a todos os requisitos inicialmente proposto, que como referido anteriormente consiste em através da utilização de linguagem natural, permitir a utilização de um computador sem a necessidade de interação física/mecânica entre o utilizador e o computador. Inicialmente na secção 4.1 é feito um levantamento dos desafios a que sistemas deste tipo encontram-se expostos. De seguida na secção 4.2 é apresentada a arquitectura global do sistema desenvolvido. E por fim, na secção 4.3 é feita uma explicação global do sistema bem como todas as ferramentas que foram utilizadas para a construção do mesmo passando para as sub secções seguintes com explicações mais detalhadas de todos os módulos que fazem parte do sistema e que permitiram a construção do mesmo, bem como justificação das escolhas que foram feitas ao longo do processo.

4.1 DESAFIOS

O maior desafio que se encontra na realização deste sistema é a movimentação do cursor do rato de tal forma que a diferença para quem usa da forma convencional seja a mínima possível. Relativamente ao controlo do teclado essa é considerada uma tarefa mais simples, supondo que o reconhecimento é bastante preciso, uma vez que não é necessário existir uma precisão tão fina, pois é algo determinístico, ou seja, caso o utilizador pretenda clicar na tecla ”N” apenas basta que o mesmo diga tecla ”N” e o mesmo se aplica a atalhos de teclado ou escrita de texto.

Deste modo ao longo deste capítulo será feita uma análise dos problemas e tentativas de soluções a esses mesmo problemas.

Para além dos problemas relatados com o desenvolvimento de um sistema de diálogo por voz, no que toca ao controlo do cursor do rato através da voz temos de ter em consideração problemas relacionados com latência. A aplicação a desenvolver tem, para além de conseguir controlar todos os aspetos do sistema, de funcionar de forma rápida e o mais próximo do tempo real. É importante que o utilizador se sinta confortável e satisfeito na utilização do sistema para que o mesmo seja aceite, caso contrário o mesmo acaba por se sentir insatisfeito e não utilizará o sistema, sendo assim um requisito fundamental que o sistema funcione de forma bastante eficiente.

Em Sporka et al. (2006) é feita uma análise entre sistemas de input baseados em discurso e sons aplicados em vídeo jogos. A principal diferença entre estes dois sistemas deve-se ao facto de no caso de sistemas baseados no discurso o sistema apenas poderá executar a ação quando o utilizador termina a sua frase e seguidamente o reconhecedor de voz reconheça o respetivo conteúdo, ao contrário do que sucede com sistema baseados em som. Nestes sistemas à medida que o som vai sendo pronunciado a própria ação vai sendo executada ao mesmo tempo. Como é mostrado na figura 11 em sistemas baseado em reconhecimento de voz, a expressão de inicio e fim da ação desejada deve ser completada antes do inicio ou fim da ação desejada. Em sistemas baseados em sons a resposta desejada pode ser mapeada diretamente a um tom e ser executada enquanto o tom é mantido. No final concluirão que através do sistema baseado em sons os participantes, que não eram pessoas com dificuldades motoras, foram 2.5 vezes mais precisos usando o sistema baseado em sons comparativamente ao sistema baseado em reconhecimento de voz. Também em Harada et al. (2011) aplicando o mesmo principio que foi aplicado em Sporka et al. (2006), também concluirão que os sistemas baseados em sons foram mais rápidos no controlo do personagem do jogo do *PacMan* relativamente ao sistema baseado em reconhecimento de voz. Estes dois estudos feitos acima contêm dados relevantes para o controlo do cursor do rato uma vez que tal como acontece nos vídeos jogos o aspeto da precisão deve ser tomado em grande consideração por forma a que o sistema se apresente o mais preciso e rápido possível e que a diferença para quem usa um rato convencional seja o mínima possível.

Outro aspeto, mas que carece um certa atenção, é a variedade de sistemas operativos e a forma como através do sistema conseguimos executar ações no mesmo. Certos sistemas operativo são mais fechados e não permitem um alargada execução de ações que poderiam, eventualmente, serem úteis quando pretendemos controlar o cursor, por exemplo, obtenção de todos os pontos de ação num ecrã que engloba botões, menus, pastas entre outros. Caso obtivéssemos essa informação poderíamos por exemplo permitir uma antecipação a posição do rato baseado na direção inicial que o utilizador escolhera, ficando com a possibilidade de não guiar com a voz o cursor desde o ponto inicial até ao seu ponto de interesse, ficando apenas a necessidade de confirmação.

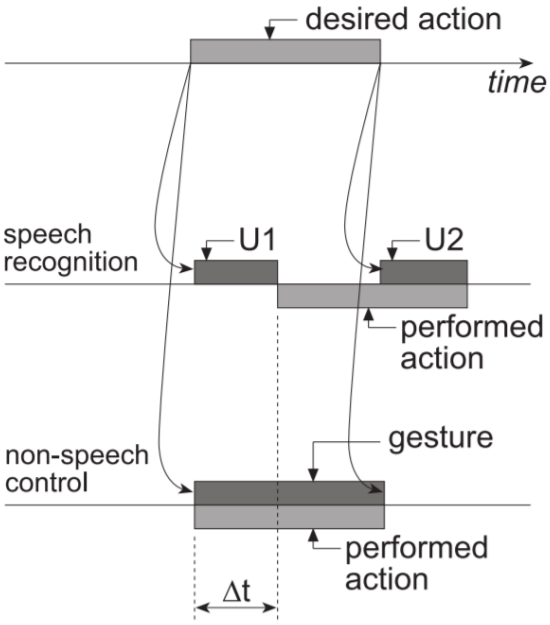


Figure 11: Diferença teórica entre um sistema baseado em discursos e baseado em sons [1]

Sendo o objetivo principal desta dissertação a utilização de linguagem natural e apesar de o sistema de sons dar a possibilidade de execução em paralelo da ação à medida que o som vai sendo pronunciado, implica que o utilizador tenha de manter o seu tom de voz enquanto da duração da ação ou ficar restrito a determinado som representativo da ação especifica, o que não se torna prático se pretendemos utilizar o sistema por muito tempo. Também é objetivo utilizar linguagem natural, isto consequentemente traz vantagens e desvantagens no contexto do controlo do computador. Primeiro temos a latência do reconhecimento de voz e em seguida a latência do processamento de linguagem natural que dará as devidas ações que o utilizador pretende. Também associado à linguagem natural a parte do gestor do diálogo deve ser suficientemente robusto para guardar alguma informação do utilizador como por exemplo quais as aplicações preferidas para determinado tipo de tarefa, por exemplo ver um documento *Portable Document Format (PDF)* ou visitar algum website.

4.2 ARQUITETURA

A arquitetura global do sistema *NaturalAssistant* é apresentada na figura 12. Esta encontra-se essencialmente dividida em dois módulo principais. O módulo para reconhecimento de voz e processamento de linguagem natural e o módulo responsável por produzir a ação

1 Imagem adaptada de Sporka et al. (2006)

desejada do utilizador baseado na resposta dada pelo processamento de linguagem natural. O módulo ”Controlador/Executar da Ação” é responsável por determinar quando o áudio deve ser enviado para o reconhecimento de voz. Também é responsável por pegar na informação dada como resposta do processamento de linguagem natural e executar a respectiva ação. Por forma a que o sistema não ficasse apenas limitado ao controlo do cursor e teclado, o módulo ”Controlador/Executor da Ação” é suficientemente módular para que seja possível desenvolver controladores para diferentes periféricos. Para melhor compreender a estrutura do sistema tomamos como exemplo a ação de movimentar o cursor para cima. O ”Controlador/Executar da Ação” ao receber esta intenção do processamento de linguagem natural fará o devido processamento e posteriormente enviará a indicação ao ”Controlador do cursor”, com os devido parâmetros necessário para a execução da ação, para iniciar a movimentação do cursor. Neste sentido o ”Controlador/Executar da Ação” comunica com cada um dos módulo independentes, para a execução de determinada tarefa.

Para a realização do sistema foi utilizada a linguagem de programação JAVA e C/C++ que permite construir um sistema de forma rápida e que ao mesmo tempo seja independente da plataforma. Neste sentido a arquitetura global do sistema consiste numa parte em JAVA que corresponde a maioria do programa onde é feito o processamento e o módulo em C/C++ que contem toda a lógica para acesso a funções de baixo nível do sistema operativo de modo a permitir conseguir controlar o mesmo dando instruções.

A linguagem adotada no sistema foi a língua portuguesa. Existe uma carência de sistemas que permitem o controlo de computador utilizando a linguagem natural, principalmente utilizando a língua portuguesa. Aliado com a escolha da língua portuguesa as aplicação que permitem o controlo do computador utilizando linguagem natural são praticamente nulas. A solução apresentada é a utilização de linguagem natural para controlo do computador. Deste modo como apresentado anteriormente o sistema conterá com delay, que será igual á variação de tempo que ocorre a fala do utilizador com o tempo que o mesmo vê a ação a ser executada no computador. Para a resolução deste problema implicaria que o reconhecimento de voz mais o processamento de linguagem natural funcionassem ambos em tempo real o que atualmente não é possível. Outra opção, que foi adotada, é minimizar o numero de interações necessárias para a execução de determinada tarefa, ou seja, para abrir determinada aplicação não é necessário o deslocamento do cursor até a devida aplicação e depois proceder ao clique, em contrapartida dar a ordem de abertura da aplicação diretamente.

Apesar dos sistemas baseados na utilização de linguagem natural não permitirem uma visualização da ação imediatamente durante a pronuncia da mesma, acontece que estes sistemas são bastante úteis para tarefas complexas. Neste caso não é necessário o utilizador

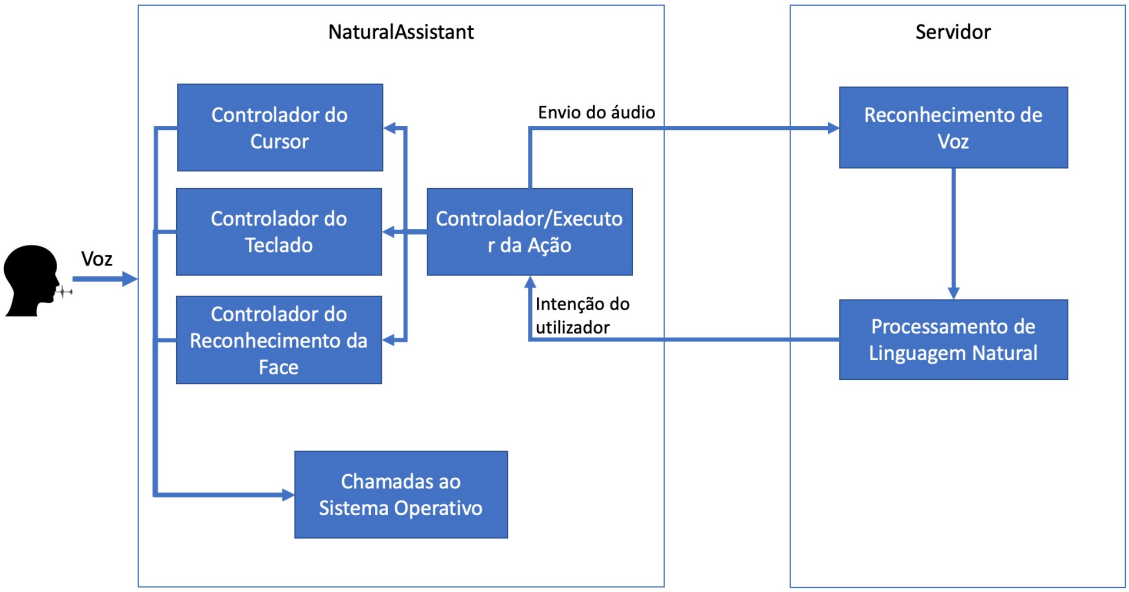


Figure 12: Arquitetura global do sistema

navegar num conjunto de janelas e menus para realizar determinada ação, bastando este apenas exprimir a ação pretendida e o sistema arranja uma forma de executá-la. Comparativamente com sistemas que definem um certo conjunto de comando pré definidos que o utilizador possa dizer, faz com que o sistema fique bastante restrito a forma que o utilizador possa exprimir determinada ação e consequentemente faz com o utilizador necessite de saber todos os comandos necessários do sistema. A linguagem natural permite desta forma eliminar a necessidade de o utilizador ter a priori um conhecimento sobre o sistema e como deverá o mesmo prenunciar determinada ação.

Em suma, os sistemas baseados em linguagem natural são mais lentos quando pretendemos controlar apenas periféricos básicos como rato e teclado, mas permitem que o sistema aceite, através da forma mais natural para o humano, a expressão de um conjunto de ações que o computador executará sem o utilizar se preocupar com todas as etapas envolvidas.

4.3 Módulos Desenvolvidos

Nesta secção é apresentado todos os módulos desenvolvidos para o sistema para que fosse possivél cobrir os objetivos inicialmente propostos.

4.3.1 *Deteção de Actividade de Voz*

Para o sistema de controlo de computador utilizando a voz é necessário a existência de um detetor de atividade de voz por forma a permitir ao sistema saber quando deve escutar e quando o utilizar terminou a expressão da ação. O sistema de reconhecimento de atividade de voz que para além de funcionar de forma correta terá de ter a capacidade de execução em tempo real e que também fosse capaz da eliminação de falsos positivos como o caso do ruído.

O sistema escolhido é uma adaptação do sistema descrito em Moattar and Homayounpour (2009) onde utiliza características tais como *Spectral Flatness Measure (SFM)* e *Short-term Energy* com um conjunto de parâmetros para perceber se o utilizador se encontra a discursar ou não.

De uma forma resumida o algoritmo funciona da seguinte forma, por cada frame de áudio que o algoritmo recebe este calcula três características do áudio recebido, para determinada se o frame de entrada corresponde a uma nova entrada de áudio no sistema ou não.

O calculo da energia do sinal é feita baseado na fórmula 3:

$$Energy = \frac{1}{n}\sum_{n=0}^{n} x^2 \tag{3}$$

Outro parâmetro utilizado é a SFM que se calcula de acordo com a fórmula 4:

$$SFM = 10\log_{10}\frac{Gm}{Am} \tag{4}$$

Onde *Média Geométrica (Gm)* é calculado da seguinte com a fórmula 5:

$$Gm = (\prod_{n=0}^{n} |x|)^{\frac{1}{n}} \tag{5}$$

E *Média Aritmética (Am)* é calculado da de acordo com a fórmula 6:

$$Am = \frac{1}{n}\sum_{n=0}^{n} x \tag{6}$$

Um outro conceito que é utilizado no algoritmo é o *Fast Fourier Transform (FFT)* que no fundo pega num sinal no domínio do tempo e coloca-o no domínio das frequências e vice-versa. A transformada rápida de Fourier calcula rapidamente essas transformações fatorizando a matriz da transformada discreta de Fourier em um produto de fatores esparsos. Como resultado, consegue-se reduzir a complexidade de calcular a transformada discreta de Fourier de $O(n^2)$, ou seja na ordem de n elevado ao quadrado, que surge quando se aplica simplesmente a definição de transformada discreta de Fourier, a $O(n \log n)$, onde n representa o tamanho dos dados.

Em conjunto com os parâmetros descritos acima o algoritmo sempre que recebe um frame de áudio processa o frame da seguinte forma:

1. Defini-se um threshold primário para a energia do sinal, um valor para a frequência predominante e um outro para o SFM

2. O algoritmo inicialmente começa por admitir que os primeiros frames são silêncio de modo a permitir definir as variáveis para definir no atual ambiente qual o mínimo valor da energia, frequência dominante e SFM. Estes valores são apenas para a parte inicial, uma vez que os mesmo vão sendo atualizados com o tempo.

3. Uma vez realizada a configuração inicial o algoritmo sempre que recebe um novo frame para validar se é som ou silêncio começa por calcular o threshold para a energia, para a frequência dominante e par o SFM da seguinte forma:

   a) $Threshold\_E = Primary\_Threshold\_E * \log Min\_E$

   b) $Threshold\_F = Primary\_Threshold\_F$

   c) $Threshold\_SFM = Primary\_Threshold\_SFM$

4. Uma nova variável é criada com valor igual a zero ($Contador = 0$), no qual é somado quando:

   a) *se* $(Energy - Min\_E) > Threshold\_E$ então *Contador* soma 1

   b) *se* $(F - Min\_F) > Threshold\_F$ então *Contador* soma 1

5. Se *Contador* for maior que um, o frame que se encontra a ser analisado é considerado como som caso contrário é selecionado como silencio.

6. Se o frame que se encontra a ser analisado é marcado como silencio a variavél que contem o valor da energia minima é atualizado juntamente com o valor do threshold da energia, dado pelas seguintes formulas:

$$Min\_E = \frac{(Silence\_Count * Min\_E) + Energy}{Silence\_Count + 1} \tag{7}$$

$$Threshold\_E = Primary\_Threshold\_E * \log Min\_E \tag{8}$$

Com a utilização deste algoritmo foi possível obter para primeiro protótipo de aplicação uma boa performance, de acordo com os autores o algoritmo foi testado com o dataset TIMIT, Farsdat e TPersianDat com uma eficácia de 94.63% para áudio sem ruido, 87.76% para áudio com ruído branco, 79.42% para áudio com *Babble noise*, 81.11% para áudio com *pink noise*, 82.01% para áudio com *factory noise* e 78.51% para áudio com *Volvo noise*.

4.3.2 *Reconhecimento de Voz*

Para perceber a intenção do utilizador é necessário existir um reconhecimento de voz por forma a ser possível identificar o texto e posterior análise para identificação da ação a ser executada.

Para o sistema de reconhecimento de voz teríamos de ter atenção que o mesmo deveria ter um baixo valor para o WER e que fosse o mais rápido possível a processar o áudio que era enviado. Com base na análise realizada, expressa na secção 3.4, verifica-se que o valor WER do *Wit.ia* e *Dialogflow* são bastante próximos, no entanto o tempo de execução é substancialmente mais baixo no serviço do *Dialogflow* pelo que a escolha passou por adotar este sistema para todas as tarefas de reconhecimento de voz.

4.3.3 *Compreensão de Linguagem Natural*

Para entender a intenção que será executada no sistema é necessário que o que foi dado como output do sistema de reconhecimento de voz seja agora processado pelo módulo de processamento de linguagem natural.

Com base nos testes realizados, expressos no capítulo 3.3, a escolha da plataforma responsável pelo reconhecimento de linguagem natural recaiu sobre o *Wit.ai*. Um dos fatores

que levou a esta escolha foi do facto da plataforma ser totalmente gratuita e apresentar, em média, valores de eficácia muito próximos da plataforma *Watson IBM* (que representa a plataforma que no geral contem um valor de eficácia maior nos três testes realizados), chegando mesmo a ter maiores valores de eficácia quando filtramos as resposta com níveis de eficácia maiores do que 90%. Outro factor é o facto da plataforma do *Wit.ai* apresentar um tempo de processamento bem mais baixo do que as sua concorrente.

4.3.4 *Cursor com Linguagem Natural*

A movimentação do cursor num computador é essencial para permitir a interação com o mesmo, é através deste, que conseguimos ter acesso a todos os menus e executar as diferentes ações. Claramente a movimentação do cursor é o aspeto mais importante nestes sistema, por isso, é importante que o sistema permita o utilizador mover o rato num computador seja rápido, simples e eficaz pois é a componente que mais se utiliza.

O objetivo principal é conseguir com que o utilizador consiga utilizar livremente o cursor no ecrã recorrendo à utilização de linguagem natural, permitindo deste modo, que o mesmo não necessariamente dependa de um conhecimento sobre comandos pré definidos, mas possa desta forma, usar a sua própria maneira para referenciar a respetiva ação.

Deste modo temos antes de mais ter em atenção que ao contrário do que sucede quando utilizamos um dispositivo fixo para controlo do cursor este oferece um infinidade de direções possíveis e tem a habilidade de ser rápido, o que se mostra bastante eficaz, mas para pessoas com problemas motores o mesmo não sucede. Por este motivo existe a necessidade de dar a possibilidade de controlar o cursor utilizando a própria voz do utilizador na medida em que é um processo praticamente de borla, uma vez que atualmente qualquer computador contem uma placa de som e microfone permitindo deste modo a capatação da voz do utilizador. Por outro lado temos de perceber que o sistemas para controlar o cursor com a voz tem vários input lag, ou seja, a execução da ação em maneira nenhuma será executada durante o instante que o utilizador esteja a enunciar a própria ação.

No sistema proposto temos dois tipos de input lag, um primeiro que é o envio e receção da informação do gestor da aplicação para com o servidor, mais o tempo que o servidor levará a processar o ficheiro de som passando-o para texto e posterior processamento da linguagem sobre esse mesmo texto.

Para esta componente do sistema o primeiro passo foi transmitir ao *Wit.ai* um conjunto de frases que se pensa serem frases típicas para a execução de determinada ação, por

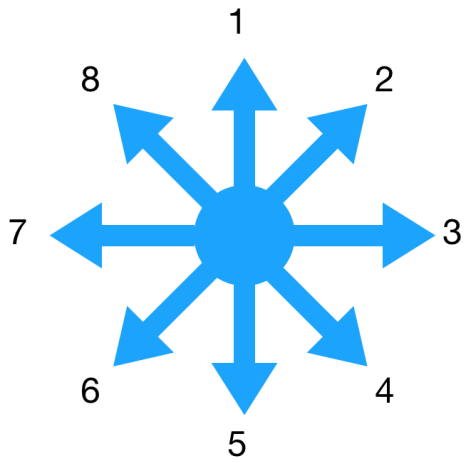


Figure 13: Direções do cursor

exemplo, ”mover o rato para cima” e ”clicar”. No sistema proposto é dado a possibilidade do utilizador realizar movimentações do cursor em 8 direções, representadas na figura 13.

Outra técnica implementada foi a utilização de uma grid como a representada na figura 4 com tamanho variável. A utilização da linguagem natural neste sistema permite deste modo que a grid contenha qualquer tamanho que o utilizador final pretender. Assim é possível com a utilização de dois comandos chegar à maior parte do ecrã uma vez que uma grid com determinado tamanho englobará muito do conteúdo ao qual o utilizador poderá interagir.

Na figura 14 é mostrado um exemplo de utilização deste sistema, onde o objetivo é mover o cursor até à pasta. Na figura encontra-se representada a série de comandos que foi necessário para chegar a pasta, no entanto os comandos apresentados não são limitativos ao que o utilizador pode dizer e sim são uma representação, na medida em que a utilização de linguagem natural permite a não restrição do que o utilizador pode ou não dizer.

Também com o objetivo de permitir uma movimentação do cursor mais livre com infinitas direções possíveis, optou-se pela utilização da biblioteca *Dlib* para deteção de pontos faciais do uma imagem, como representados na figura 15. Esta biblioteca disponibiliza um ficheiro, denominado por *Shape Prediction*, que foi obtido através do método *Histogram Oriented Gradients*, combinado com um classificador linear, uma pirâmide de imagem e um esquema de deteção de janela deslizante. Este ficheiro corresponde a uma rede neuronal já treinada que estará encarregue de detetar as coordenadas dos pontos de interesse. Para a deteção dos pontos de interesse numa imagem é necessário em primeiro lugar realizar a

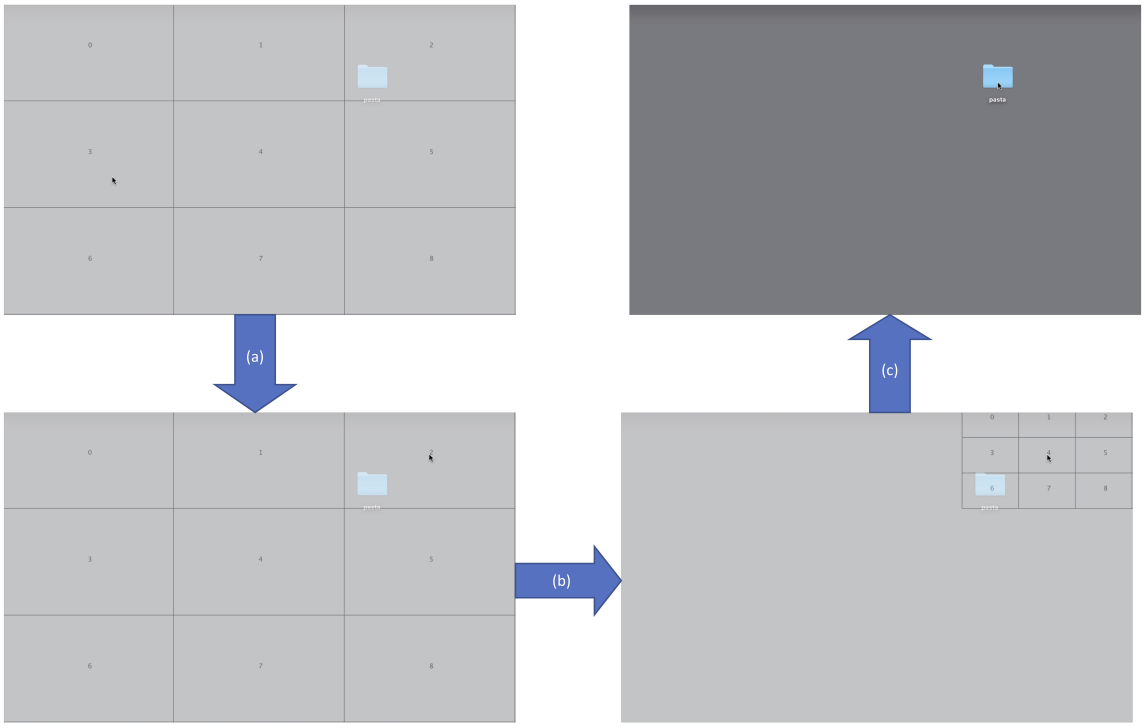


Figure 14: Exemplo de utilização do sistema de grid; (a) ”2”; (b) ”clicar”; (c) ”6 e fechar”

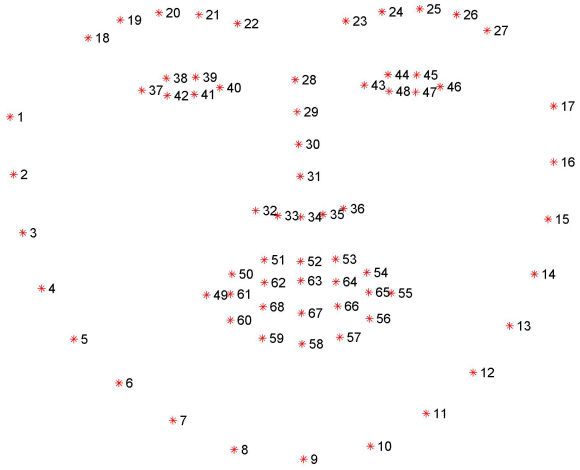


Figure 15: Dlib landmarks [2]

deteção da face numa imagem que neste caso será dada pela webcam do computador do utilizador, e para permitir que funcione em tempo real utilizou-se o OpenCV com o método *Haar Cascades* para a deteção da face.

Obtendo as landmarks da face na cara os primeiros frames será para marcar a posição central do utilizador na imagem por forma a permitir calcular os vetores orientadores que serão os guias para a movimentação do cursor no ecrã. Neste caso a movimentação passa a ter praticamente direções infinitas bastando o utilizador mover ligeiramente a cara na direção que pretender mover o cursor. A linguagem natural ajudará o utilizador para informar se pretende clicar, parar a movimentação, aumentar a velocidade de movimentação, entre outros diversos comandos. O maior problema deste tipo de sistema são as condições sobre as quais o utilizador se encontra a utilizar o computador, dado que em ambiente muito escuros ou muito claros a deteção da cara tende a não ser extremamente precisa e consequentemente os pontos faciais não são corretamente detetados. Este sistema tanto é utilizável em movimentação do cursor, como no scroll durante a visualização de um documento onde o utilizador poderá sempre que pretender pausar a movimentação. Apesar de estar sujeito as condições onde o utilizador se encontra a utilizar o computador mostra-se uma boa alternativa ao método de grid pois permite uma movimentação rato mais suave aproximando-se com a sensação da utilização do hardware (rato). Na figura 16 encontra-se representado uma exemplo onde o ponto a verde representa um ponto de referência que foi calculado durante a calibração e o ponto a azul representa o ponto dado pela biblioteca *Dlib* que corresponde ao ponto central da face (nariz). A linha a laranjada representa o vetor orientador do movimento do rato. A distancia entre os dois pontos referidos serve neste

2 Imagem obtida de https://www.pyimagesearch.com/wp-content/uploads/2017/04/facial_landmarks_68markup.jpg

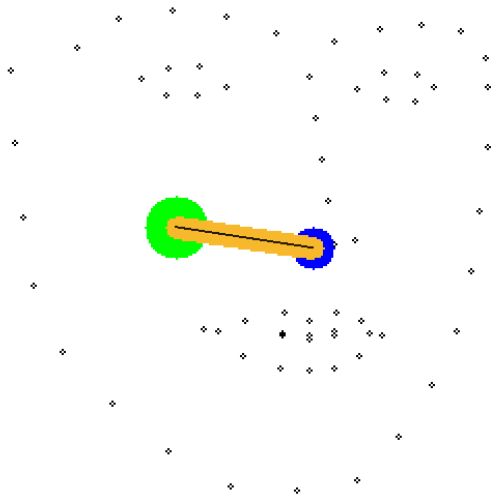


Figure 16: Exemplo da utilização do rato com a face

caso apenas para informar o sistema se deve parar ou começar a movimentação do cursor. Este sistema não se pretende que seja um substituto á utilização de linguagem natural mas um auxiliar.

Por fim, fugindo um pouco ao objetivo central desta dissertação, recorreu-se ao uso de comandos de voz para tentar minimizar tanto erros que advêm do reconhecimento de voz como problemas relacionado com a latência de um comando e a sua execução. Este sistema serviu somente para testar tempos relativamente a performance comparativamente ao mesmo utilizando a linguagem natural. Por este motivo neste sistema apenas foi dada a possibilidade de o utilizador apenas controlar todos os aspetos do rato, onde cada comando corresponderia a uma ação. Também é possível ao utilizador alterar os comandos que foram previamente definidos.

4.3.5 *Teclado com Linguagem Natural*

O teclado representa uma forma eficiente de introdução de conteúdo num computador. Existem duas importantes partes no teclado, a primeira deve-se ao facto e que o teclado permite a digitação de texto e a segunda representa uma possibilidade de execução de ações por intermediário da utilização de atalhos.

No que toca á escrita propriamente dita de documento esta é feito utilizado o sistema de reconhecimento de voz onde o utilizador primeiramente terá de dar ordem de inicio do estado de escrita, o sistema neste estado vai continuamente enviando o áudio para o reconhecimento de voz e consequente escrita do documento. O processo de escrita fica dependente da eficácia do reconhecimento de voz, que sendo ele preciso a pessoa não tem qualquer dificuldade na escrita, aliás a sua escrita é bem mais rápida comparativamente á utilização de um comum teclado. Caso não seja de interesse do utilizador uma continua transcrição da sua voz o mesmo poderá solicitar apenas para escrever determinado texo.

Outra parte importante é a utilização de atalhos. A utilização de atalhos permitiu automatizar tarefas, que caso contrário haveria a necessidade de movimentação do curso, por exemplo num documento se o utilizador pretender dar a indicação que quer um porção do texto a negrito. No sistema esta tarefa consistiria apenas na utilização de uma frase referir o texto que pretende que esteja a negrito, não havendo a necessidade de movimentar o rato até ao texto, fazer a sua seleção e posterior colocação a negrito. Ainda tendo este exemplo da colocação de palavra a negrito é de ressaltar que o sistema porventura existirem mais do que duas mesmas palavras o sistema manterá o estado do que foi feito anteriormente e consequentemente se a primeira palavra não for a que o utilizador pretende ele poderá apenas dizer para passar à seguinte ou referir um numero total de palavra à frente da qual a primeira palavra se encontra oferecendo um salto direto entre palavras permitindo fazer um salto de um estado incorreto para um correto com a palavra a negrito que o utilizador realmente pretende.

4.3.6 *Interação com Aplicações*

Para interação com as aplicações estas muitas vezes disponibilizam alguns atalhos que são úteis para interagir com as aplicação, de facto apresenta-se claramente como um ótimos substituo da movimentação do rato. Pegando como exemplo fechar uma janela o utilizador não necessita de necessariamente fechar a janela recorrendo á movimentação do rato uma vez que essa mesma ação se poderá traduzir num atalho. Este tipo de atalhos poderão ser definidos pelo utilizador, ou seja, para além do que é dado no sistema o utilizador poderá definir novos comandos, que bem pretender, que executem diferentes tipo de ações. Para a definição de novas ações o utilizador terá primeiramente de dar indicação ao sistema que pretender criar nova ação, o sistema automaticamente perguntará para o utilizador dizer três frases exemplo, para qual a ação terá de ficar associada e de seguida o utilizador executa um conjunto de ações aos quais pretende que as frases estejam associadas. No final o utilizador dá a intenção de terminar a gravação das ações, e a ação é posteriormente

gravada no sistema. Deste modo quando o utilizador pretender executar um conjunto de ações que gravou basta apenas dizer alguma frase, ou derivadas, das quais ficaram associadas com a ação. Este tipo de possibilidade permite que certo tipo de ações que o sistema de controlo do computador não entenda seja possível de ser configurado. Supondo por exemplo que o utilizador pretende todos os dias abrir o navegador web e pesquisar as noticias do dia, em alternativa à movimentação do rato ou dar indicação para abrir aplicação e posteriormente informar o sistema que pretende ver as noticias do dia, apenas com o comando, e derivados, que definiu previamente, poderá com uma simples frase realizar a execução de um conjunto de tarefas que o mesmo programou.

4.3.7 *Definição de Ações em Janelas*

Algo que se percebeu na realização do projeto foi que maioritariamente as interfaces gráfica das aplicações não mudam, ou seja, imaginando a aplicação *Word* o botão onde alinha o texto normalmente mantêm-se sempre numa mesma posição baseado no tamanho da janela. Com base neste conceito implementou-se a possibilidade do próprio utilizador definir as sua próprias ações dentro do contexto da aplicação corrente. Deste modo comandos relativos a determinada aplicação que o sistema a priori não contêm qualquer informação poderá ser programa pelo utilizador. Suponhamos que o utilizador que utiliza o sistema utiliza uma aplicação de criação de documento onde não existe o atalho de colocação de palavras a negrito. Neste caso o utilizador na plataforma poderá mover o cursor utilizando a voz para executar a respetiva ação e o sistema vai registando esse conjunto de ações no contexto da aplicação. Estas ações ficam guardas no sistema e apenas são executadas dentro da aplicação à qual ficou associada. Também é dada a possibilidade de o mesmo guardar, associado à aplicação, um conjunto de diversas outras ações que não só a movimentação do cursor.

4.3.8 *Múltiplos comandos numa frase*

A linguagem natural é claramente um processo fortíssimo quando pretendemos executar diversas ações baseado na linguagem em que utilizamos no dia a dia. Pelo que foi dito anteriormente o utilizador poderia proferir uma simples frase que se traduziria em determinada ação, sendo que as frases submetidas no sistema serão validadas pelo mesmo e o sistema será responsável pela execução da respetiva ação desejada.

Contudo pretende-se também fazer com que o sistema não execute apenas e somente simples ações e sim numa só frase retirar um conjunto de ações. Contudo estas ações devem ser de certo modo distintas, ou seja, suponhamos que pretendemos ver o email e pesquisar alguma coisa no *Google*. Em sistema comuns o utilizador teria de dizer "Ver email" e uma vez executada a ação ai estava liberado para o mesmo dizer "Pesquisar no *Google*". Contudo no contexto de linguagem natural isto é possível ser feito em apenas uma frase, como por exemplo a frase "Abrir o email e fazer uma pesquisa no Google", isto levará a que o utilizador não tenha de dizer uma frase a seguir a outra para executar duas ações distintas mas sim a possibilidade de numa mesma frase executar diversas ações, fazendo com que não seja necessário uma interação proporcional ao número de ações desejadas.

4.4 VALIDAÇÃO DO SISTEMA

Para a validação do sistema foi reunido um conjunto de seis pessoas que não contêm problemas físicos. O teste consiste em dar aos utilizadores um conjunto de tarefas que deverão igualmente serem executadas utilizando um rato e teclado e também realizar a mesma tarefa utilizando para isso linguagem natural. Os parâmetros utilizados para validação serão o tempo médio entre os seis participantes na execução das seguintes tarefas:

1. Deslocamento do cursor do rato para posição especifica no ecrã utilizando linguagem natural
2. Deslocamento do cursor do rato para posição especifica no ecrã utilizando a face
3. Deslocamento do cursor do rato para posição especifica no ecrã utilizando a grid
4. Deslocamento do cursor do rato para posição especifica no ecrã utilizando comandos de voz
5. Realizar uma pesquisa web
6. Escrever uma porção de texto
7. Colocar uma palavra a negrito
8. Abrir duas aplicações distintas

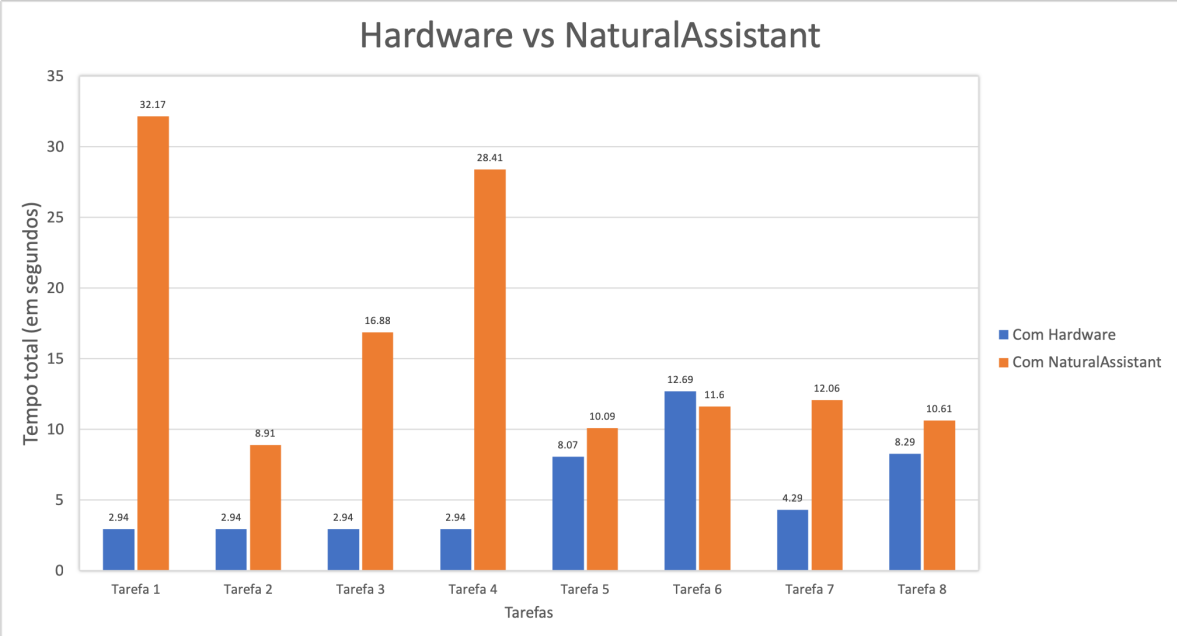


Figure 17: Gráfico dos tempos entre a utilização de hardware vs NaturalAssistant

	Utilizando o Teclado e Rato	**Utilizando o sistema apresentado**
Tarefa		
1	2,94	32,17
2	2,94	8,91
3	2,94	16,88
4	2,94	28,41
5	8,07	10,09
6	12,69	11,60
7	4,29	12,06
8	8,29	10,61

Table 3: Tempos (em segundos) da realização de tarefas utilizando o hardware habitual (teclado e rato) comparativamente com o software apresentado nesta dissertação

Apesar dos testes realizados não terem sido utilizados por pessoas com dificuldades motores, reflete as vantagens e desvantagens de utilizar um sistema baseado em linguagem natural comparativamente com a utilização de algum tipo de periférico especifico para o controlo do cursor e teclado. As primeiras quatro tarefas consistiam em avaliar de que forma variava o tempo com diferentes métodos no quesito do controlo do cursor. Para isso foi escolhido um ponto aleatório no ecrã tanto para a posição inicial do cursor como para o ponto final, ponto o qual o utilizador teria de mover o cursor. A distância entre estes dois pontos foi de aproximadamente 1407 pixeis. Este calculo foi feito utilizado a fórmula

9 onde (X_0, Y_0) é o ponto final e (X_1, Y_1) é o ponto inicial. Pelos testes realizados é notória a diferença de tempos entre a utilização do periféricos especifico para controlo do cursor comparativamente a utilização de linguagem natura. Como dito anteriormente o sistemas para controlar o cursor tem de em primeiro perceber que o utilizador terminou a introdução da sua ação no sistema, deteção de fim de discurso, e consequente reconhecer e compreender o que foi reconhecido o que leva a que a ação demore mais tempo quando comparado com o hardware do rato. Outro aspecto a ter em conta é a variação de velocidade que um dispositivo hardware (rato) apresenta onde fácilmente podemos aumentar ou diminuir a velocidade do movimento. Quanto a utilização do sistema grid este mostrou-se bastante próxima uma vez que uma grid $10x10$ cobria a posição final desejada. Por fim temos que a utilização da face como forma de controlar o cursor torna o sistema mais eficiente, pois neste caso tempos com deteção de final de discurso e reconhecimento de voz não são um problema.

Os testes também mostraram algo que já foi referido anteriormente, o tempo quando pretendemos realizar uma tarefa especifica encontra-se praticamente ao mesmo nível da utilização de um periférico para o controlo do cursor. Pelo que podemos concluir que à medida que os sistemas de reconhecimento de voz se tornem mais rápidos e eficientes irão certamente permitirem um rápido processamento do sinal e claramente a utilização de sistemas com a utilização da voz vão ser mais preferíveis do que a utilização de periféricos. Para tarefas de escrita de texto o sistema apresentado conseguiu obter melhores resultados comparativamente com a utilização de um teclado.

$$D = \sqrt{(X_0 - X_1)^2 + (Y_0 - Y_1)^2} \tag{9}$$

5

CONCLUSÃO

Nesta secção, são apresentadas todas as conclusões e reflexões tiradas ao longo desta dissertação. Para além disso, haverá também um levantamento de ideias para uma possível melhora da aplicação.

5.1 CONCLUSÕES DO TRABALHO REALIZADO

Como se pode constar, a utilização de linguagem natural como forma de controlar um computador mostra-se uma forma viável em alternativa a utilização de periféricos comuns, como a utilização do rato e teclado. No entanto este tipo de sistemas carecem de um conjunto de fatores para a sua construção.

Percebemos que a construção de um chatbot com o objetivo de utilizar um computador usando a voz, envolve uma série de tecnologias, que não são à data perfeitas, o que leva que o chatbot para controlo do computador não seja também ele perfeito. Percebemos que uma das componentes mais fundamentais nestes sistema é o módulo de reconhecimento de voz. Preferencialmente este deverá ser em tempo real, o que atualmente isso não acontece. O tempo de reconhecimento das ações por parte do módulo de reconhecimento de voz, introduz um certo delay no sistema que é percetível quando comparado com a utilização de um rato e teclado. Também observamos que o módulo de processamento de linguagem natural é bastante eficiente na sua tarefa, produzindo bem as sua ações a serem executadas. Apesar disto, verifica-se que a utilização de linguagem natural como forma de expressar ações complexas torna-se um processo mais rápido quando comparado com a utilização de uma rato e teclado que deverá navegar num conjunto de janelas para a elaboração da mesma ação. Outra vantagem neste tipo de sistemas é a escrita, onde o reconhecimento de voz não falhando ou falhando pouco torna o processo de escrita de documento num processo bem mais rápido e mais fácil para o utilizador, onde este não necessita de premir

um conjunto de teclas mas sim expressar-se como habitualmente faria quando comunica com outros humanos.

Espera-se que com o avanço da tecnologia de reconhecimento de voz e processamento de linguagem natural, estes acabem por funcionar em tempo real, dando uma resposta imediatamente à pergunta expressa pelo orador. No futuro, a alternativa de controlar um computador utilizando linguagem natural não será uma hipótese e sim uma realidade e isso pode ser visto com o número de assistentes virtuais que têm nos últimos anos vindo a surgir. Esta popularização deve-se ao facto de para o humano não ficar retida a uma forma especifica de expressar a ação e sim poder expressar como bem entender, não necessitando o mesmo de aprender de antemão a utilização de determinado sistema.

Em suma, espera-se que à medida que com a introdução de novas tecnologia no futuro e o aumento do poder de processamento dos computadores seja possível ao computador reconhecer e interpretar o que o utilizador deseja e executar a ação diretamente não precisando este de interagir num conjunto de menus ou apreender como o sistema funciona.

5.2 PERSPETIVAS DE TRABALHO FUTURO

Um dos maiores problemas que é encontrado neste tipo de sistema é latência inerente da utilização do reconhecimento de voz mais o processamento de linguagem natural. Talvez por este motivo a tecnologia de utilização de linguagem natural como forma alternativa aos métodos tradicionais de utilização do rato e teclado, não tenha sido alvo de grande pesquisa por parte da comunidade. Uma possível melhora no sistema seria o estudo e criação de duas componentes fundamentais neste sistema: o reconhecimento de voz e o processamento de linguagem natural. Em primeiro lugar é necessário colocar o reconhecimento de voz e processamento de linguagem natural em modo offline, para que todo o sistema não fique limitado a uma ligação web para ser utilizado.

Outro aspeto que infelizmente não foi possível ser abordado, foi o teste da aplicação junto de pessoas com problemas motores/físicos por forma a que o sistema fosse testado num contexto no qual a aplicação encontra-se mais focalizado.

Por fim seria interessante criar uma plataforma centralizada para que à medida que a comunidade criasse novos comandos estes pudessem ser validades e consequentemente disponibilizados para os restantes utilizadores. Outro aspeto que depende em grande parte do sistema operativo era estudar a possibilidade, que independente da plataforma, permitisse a deteção de pontos de interesse no ecrã para que fosse possível a previsão da

movimentação do cursor. Deste modo, poderíamos inferir que uma posição do cursor no ecrã baseado no movimentação corrente seria um ponto de ação possível (como pastas, botões entre outros).

BIBLIOGRAPHY

Rubeena A. Concatenative speech synthesis: A review. *International Journal of Computer Applications*, 136:1–6, 02 2016. doi: 10.5120/ijca2016907992.

O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1533–1545, Oct 2014. ISSN 2329-9290. doi: 10.1109/TASLP.2014.2339736.

Sameera A. Abdul-Kader and Dr. John Woods. Survey on chatbot design techniques in speech conversation systems. *IJACSA*, 2015.

Scott D. Scargle Armando B. Barreto and Malek Adjouadi. A practical emg-based human-computer interface for users with motor disabilities. *Journal of Rehabilitation Research and Development*, 2000.

R. McCauley B. Manaris and V. MacGyvers. An intelligent interface for keyboard and mouse control - providing full access to pc functionality via speech. In *Proceedings of 14th International Florida AI Research Symposium (FLAIRS-01*, pages 182–188. AAAI Press, 2001.

Archana Balyan, S. S. Agrawal, and Amita Dev. Speech synthesis: A review. 2013.

Srinivas Bangalore, Dilek Hakkani-Tur, and Gokhan Tur. Introduction to the special issue on spoken language understanding in conversational systems. 48:233–238, 03 2006.

Chung-Cheng Chiu, Tara Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Katya Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. State-of-the-art speech recognition with sequence-to-sequence models. 12 2017.

Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Y Bengio. Attention-based models for speech recognition. 06 2015.

Joe Polifroni Stephanie Seneff David Goddeau, Helen Meng and Senis Busayapongchaiy. A form-based dialogue manager for spoken language applications. In *In Proc. ICSLP*, pages 701–704, 1996.

Mathias De Wachter, Mike Matton, Kris Demuynck, Patrick Wambacq, Ronald Cools, and Dirk Van Compernolle. Template-based continuous speech recognition. *Audio, Speech,*

and Language Processing, IEEE Transactions on, 15:1377 – 1390, 06 2007. doi: 10.1109/TASL.2007.894524.

R Dixit and Navdeep Kaur. Speech recognition using stochastic approach : A review. 2013.

Cicero Dos Santos and Victor Guimarães. Boosting named entity recognition with neural character embeddings. 05 2015. doi: 10.18653/v1/W15-3904.

Mark Gales and Steve Young. The application of hidden markov models in speech recognition. *Found. Trends Signal Process.*, 1(3):195–304, January 2007. ISSN 1932-8346. doi: 10.1561/2000000004. URL http://dx.doi.org/10.1561/2000000004.

Luqman Gbadamosi. Voice recognition system using template matching. *International Journal of Research in Computer Science*, 3:13–17, 09 2013. doi: 10.7815/ijorcs.35.2013.070.

Li Deng George E. Dahl, Dong Yu and Alex Acero. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. In *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 2012.

James R. Glass. Challenges for spoken dialogue systems. 10 1999.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. 38, 03 2013.

Tarif Haque, Emily Liang, and Jeff Gray. The adjustable grid: A grid-based cursor control solution using speech recognition. In *Proceedings of the 51st ACM Southeast Conference*, ACMSE '13, pages 36:1–36:6, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1901-0. doi: 10.1145/2498328.2500084.

Susumu Harada, James A. Landay, Jonathan Malkin, Xiao Li, and Jeff A. Bilmes. The vocal joystick:: Evaluation of voice-based cursor control techniques. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 197–204, New York, NY, USA, 2006. ACM. ISBN 1-59593-290-9. doi: 10.1145/1168987.1169021.

Susumu Harada, Jacob O. Wobbrock, and James A. Landay. *Voice Games: Investigation Into the Use of Non-speech Voice Input for Making Computer Games More Accessible*, pages 11–29. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-23774-4. doi: 10.1007/978-3-642-23774-4_4.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov 2012a. ISSN 1053-5888. doi: 10.1109/MSP.2012.2205597.

Geoffrey Hinton, li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Phuongtrang Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. 29:82–97, 11 2012b.

Julia Hirschberg and Christopher D. Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015. ISSN 0036-8075. doi: 10.1126/science.aaa8685.

Takeo Igarashi and John F. Hughes. Voice as sound: Using non-verbal voice input for interactive control. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 155–156, New York, NY, USA, 2001. ACM. ISBN 1-58113-438-X. doi: 10.1145/502348.502372.

Sangramsing Kayte. Marathi speech synthesis: A review. *International Journal of Computers, Communications and Control (IJCCC)*, 3, 08 2015.

Sangramsing Kayte, Monica Mundada, and Jayesh Gujrathi. Hidden markov model based speech synthesis: A review. *International Journal of Computer Applications*, 130:975–8887, 12 2015. doi: 10.5120/ijca2015906965.

Martin Klarner and Bernd Ludwig. *Hybrid Natural Language Generation in a Spoken Language Dialog System*, pages 97–111. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-30221-6. doi: 10.1007/978-3-540-30221-6_9.

Krerksak Likitsupin, Proadpran Punyabukkana, Chai Wutiwiwatchai, and Atiwong Suchato. Acoustic-phonetic approaches for improving segment-based speech recognition for large vocabulary continuous speech. *Engineering Journal*, 20:179–197, 05 2016. doi: 10.4186/ej.2016.20.2.179.

Bing Liu and Ian Lane. Attention-based recurrent neural network models for joint intent detection and slot filling. pages 685–689, 09 2016. doi: 10.21437/Interspeech.2016-1352.

Suhas Mache, Manasi R Baheti, C Mahender, and Asst Professor. Review on text-to-speech synthesizer. *International Journal of Advanced Research in Computer and Communication Engineering*, 4:54–59, 09 2015. doi: 10.17148/IJARCCE.2015.4812.

Michael F. McTear. Spoken dialogue technology: Enabling the conversational user interface. *ACM Comput. Surv.*, 34(1):90–169, March 2002. ISSN 0360-0300. doi: 10.1145/505282.505285.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Y Bengio, li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig. Using recurrent neural networks for slot filling in spoken language understanding. *Audio,*

Speech, and Language Processing, IEEE/ACM Transactions on, 23:530–539, 03 2015. doi: 10.1109/TASLP.2014.2383614.

Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013, 01 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, G.s Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 10 2013b.

M. H. Moattar and M. M. Homayounpour. A simple but efficient real-time voice activity detection algorithm. 2009.

Pragnesh Jay Modi, Manuela Veloso, Stephen F. Smith, and Jean Oh. *CMRadar: A Personal Assistant Agent for Calendar Management*, pages 169–181. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-31946-7. doi: 10.1007/11426714_12.

Sercan O. Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. Deep voice: Real-time neural text-to-speech. 02 2017.

Bhagath Parabattina and Pradip Das. Acoustic phonetic approach for speech recognition: A review. 11 2016.

Gérard Chollet Jérôme Boudy Pierrick Milhorat, Stephan Schlögl and Télécom Sudparis. Multi-step natural language understanding.

L.R. Rabiner. A tutorial on hidden markov models and selected applications on speech recognition. 77:257 – 286, 03 1989.

Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowl. Eng. Rev.*, 21(2):97–126, June 2006. ISSN 0269-8889. doi: 10.1017/S0269888906000944.

Jose Sotelo, Soroush Mehri, Kundan Kumar, João Felipe Santos, Kyle Kastner, and Aaron Courville. Char2wav: End-to-end speech synthesis. 2017.

Dimitris Spiliotopoulos, Ion Androutsopoulos, and Constantine D. Spyropoulos. Human-robot interaction based on spoken natural language dialogue. In *in: Proceedings of the European Workshop on Service and Humanoid Robots*, pages 25–27, 2001.

Adam J. Sporka, Sri H. Kurniawan, Murni Mahmud, and Pavel Slavík. Non-speech input and speech recognition for real-time control of computer games. In *Proceedings of the 8th*

International ACM SIGACCESS Conference on Computers and Accessibility, pages 213–220, New York, NY, USA, 2006. ACM. ISBN 1-59593-290-9. doi: 10.1145/1168987.1169023.

Jian Tang, Yifan Yang, Samuel Carton, Ming Zhang, and Qiaozhu Mei. Context-aware natural language generation with recurrent neural networks. *CoRR*, abs/1611.09900, 2016.

Arul Valiyavalappil Haridas, Ramalatha Marimuthu, and Vaazi Gangadharan Sivakumar. A critical review and analysis on techniques of speech recognition: The road ahead. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 22:39–57, 03 2018. doi: 10.3233/KES-180374.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: A fully end-to-end text-to-speech synthesis model. 03 2017.

Tsung Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. 08 2015. doi: 10.18653/v1/D15-1199.

Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. The microsoft 2017 conversational speech recognition system. Technical report, August 2017. URL https://www.microsoft.com/en-us/research/publication/microsoft-2017-conversational-speech-recognition-system/.

Ramin Yaghoubzadeh, Marcel Kramer, Karola Pitsch, and Stefan Kopp. *Virtual Agents as Daily Assistants for Elderly or Cognitively Impaired People*, pages 79–91. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-40415-3. doi: 10.1007/978-3-642-40415-3_7. URL https://doi.org/10.1007/978-3-642-40415-3_7.

Xuesong Yang, Yun-Nung Chen, Dilek Hakkani-Tur, Paul Crook, Xiujun Li, Jianfeng Gao, and li Deng. End-to-end joint learning of natural language understanding and dialogue manager. 12 2016.

Shin Takahashi Yoshiyuki Mihara and Etsuya Shibayama. Wataridori: Multiple ghost cursorsfor speech-based cursor movement. 2004.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13:55–75, 08 2018. doi: 10.1109/MCI.2018.2840738.

H. Ze, A. Senior, and M. Schuster. Statistical parametric speech synthesis using deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7962–7966, May 2013. doi: 10.1109/ICASSP.2013.6639215.

V. W. Zue and J. R. Glass. Conversational interfaces: advances and challenges. *Proceedings of the IEEE*, 88(8):1166–1180, Aug 2000. ISSN 0018-9219. doi: 10.1109/5.880078.