Universidade de Aveiro
2004
Departamento de Engenharia Electrónica e Telecomunicações

Mário Jorge Ferreira Rodrigues

Interacção Humano-Robô ao Nível da Linguagem

Universidade de Aveiro
2004
Departamento de Engenharia Electrónica e Telecomunicações

Mário Jorge Ferreira Rodrigues

Interacção Humano-Robô ao Nível da Linguagem

dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor António Joaquim da Silva Teixeira, Professor do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro e do Doutor Luís Filipe de Seabra Lopes, Professor do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro

o júri

presidente

Francisco António Cardoso Vaz
Professor Catedrático da Universidade de Aveiro

António Joaquim da Silva Teixeira
Professor Auxiliar da Universidade de Aveiro (orientador)

Luís Filipe de Seabra Lopes
Professor Auxiliar da Universidade de Aveiro (co-orientador)

João Paulo Silva Neto
Professor Auxiliar do Instituto Superior Técnico da Universidade Técnica de Lisboa

agradecimentos

Quero agradecer a todos aqueles que me apoiaram no decurso deste trabalho: à Luísa, aos meus pais e aos meus orientadores.

resumo

Sistemas capazes de interagir com seres humanos através de linguagem natural falada têm sido alvo de muitos estudos e projectos. Estes sistemas são aliciantes, pois permitem ao Homem comunicar com a máquina de uma forma simples e intuitiva.
O trabalho apresentado tem por objectivo tornar a interacção entre um humano e um robô, utilizando linguagem natural falada, mais eficiente e robusta, permitindo a sua utilização em ambientes mais adversos. O trabalho efectuado consistiu, essencialmente, na adaptação dos modelos acústicos de um reconhecedor de fala às condições de utilização e na concepção e desenvolvimento de um sistema de compreensão de linguagem natural, o mais possível imune a eventuais erros cometidos pelo sistema de reconhecimento.
Conclui-se este trabalho avaliando, comparativamente, o sistema desenvolvido e o sistema em utilização no robô Carl.
Os resultados obtidos apontam para uma melhoria do desempenho com o treino dos modelos acústicos. Também se verifica que o módulo de compreensão de linguagem natural é capaz de processar todas as frases aceites pelo sistema em utilização no Carl, extraindo informação de um número significativo de frases adicionais.

abstract

Systems capable of interacting with humans using spoken natural language have been the subject of many studies and projects. These systems are useful because they allow Man-machine communication in a simple and intuitive way. The main goal of the work presented is the development of a more efficient and robust human-robot interaction using spoken natural language, allowing its use in more adverse environments. The work focused on the adaptation of the speech recognizer acoustic models and development of a natural language understanding system capable of handling errors from the speech recognizer. The work is concluded with the comparative evaluation of the new system and the one currently in use by the robot Carl.

Results point to an improved performance of the new trained speech recognition acoustic models. It is also verified that the new natural language understanding module is capable of processing all the utterances accepted by Carl current system, extracting meaningful information from a significant additional number of utterances.

Índice

1 Introdução **1**

1.1 Comunicação Humano-Robô . . . 2
1.2 CARL . . . 2
1.3 Evolução da Interface de Fala do Carl . . . 4
1.4 Outros Robôs . . . 7
1.4.1 Godot . . . 8
1.4.2 IBL: Instruction-Based Learning for Mobile Robots . . . 9
1.4.3 JIJO-2 . . . 9
1.4.4 Philips: Futebol Robótico . . . 10
1.4.5 Robita . . . 11
1.5 Objectivos . . . 11
1.6 Estrutura da Dissertação . . . 13
1.7 Resultados Publicados . . . 13

2 Processamento de Linguagem Natural Falada **15**

2.1 Reconhecimento de Fala . . . 16
2.1.1 Classificação dos Reconhecedores de Fala . . . 17
2.1.2 Fundamentos do Reconhecimento de Fala Contínua . . . 18
2.1.3 Extracção das Características Acústicas do Sinal de Fala . . . 19
2.1.4 Modelos de Markov . . . 20
2.1.5 Descodificação . . . 23
2.1.6 Dicionário de Pronúncia . . . 24
2.1.7 Modelos de Linguagem . . . 25
2.2 Compreensão de Linguagem Natural . . . 25
2.2.1 Abordagens Existentes . . . 26
2.2.2 Anotação Morfossintáctica . . . 27
2.2.3 *Parsing* . . . 30
2.2.4 *Parsing* Robusto . . . 30

3 Sistema Híbrido de Processamento de Linguagem Natural Falada **35**

3.1 Abordagem Escolhida . . . 35
3.2 Descrição Detalhada . . . 36
3.2.1 Pré-Processamento . . . 37

3.2.2 Análise Sintáctica . . . 39
3.2.3 Análise Semântica . . . 41
3.2.4 Exemplo de Análise . . . 42
3.3 Ferramentas utilizadas . . . 44
3.3.1 Nuance 8.0 . . . 44
3.3.2 Sphinx 2 . . . 44
3.3.3 Sphinx 3 . . . 44
3.3.4 Brill Tagger . . . 45
3.3.5 LTPOS . . . 47
3.3.6 LCFlex . . . 47
3.3.7 TiMBL . . . 48

4 Resultados **51**
4.1 Teste ao Reconhecimento de Fala . . . 52
4.1.1 Metodologia do Teste . . . 53
4.1.2 Resultados . . . 54
4.2 Teste ao Sistema de Compreensão . . . 58
4.2.1 Metodologia do Teste . . . 59
4.2.2 Resultados . . . 59
4.3 Discussão . . . 70

5 Conclusões **73**
5.1 Resumo do Trabalho Apresentado . . . 73
5.2 Resultados Mais Importantes . . . 74
5.3 Trabalho Futuro . . . 75

A Aprendizagem Automática **83**
A.1 Aprendizagem Baseada em Memórias . . . 83
A.2 Aprendizagem Baseada em Transformações . . . 84

B Informação Utilizada no LCFlex **87**
B.1 Léxico . . . 87
B.2 Gramática . . . 90

Lista de Figuras

1.1 Fotografia do robô Carl. 3
1.2 Fotografia do robô Godot e um utilizador. Fonte Godot (2003). 8
1.3 Fotografia do robô utilizado no projecto IBL. Fonte: IBL (2004). 9
1.4 Fotografia do robô Jijo-2 e um utilizador. Fonte Matsui *et al.* (1999). 10
1.5 Fotografia de um robô da Philips com uma bola. Fonte Philips (2002). 11
1.6 Fotografia do robô Robita. Fonte Robita (2002). 12

2.1 Arquitectura geral de um sistema de processamento de linguagem natural falada. 15
2.2 Arquitectura geral de um sistema de reconhecimento de fala. 16
2.3 Esquema de como as unidades linguísticas básicas, representadas neste caso por HMMs com 5 estados, são encadeadas de modo a formarem palavras e frases. Fonte Teixeira e Vaz (2004). 21
2.4 Esquema de um modelo de Markov com cinco estados numa topologia esquerda-direita. Fonte Meinedo (2000). 22
2.5 Malha de pesquisa de hipóteses de frase. 23
2.6 Esquema de um módulo de compreensão de linguagem natural. 26
2.7 Representação de uma SSN. Fonte Henderson e Lane (1998, pg. 533). 32

3.1 Fluxograma da solução adoptada. 35
3.2 Arquitectura do sistema desenvolvido. 38
3.3 Análise sintáctica do LCFlex para a frase "where is the coordinator of Carl". 43
3.4 Peso das instâncias consoante a sua distância à instância a classificar 50

4.1 Exemplo de medida do grau de corrupção de uma frase. 56
4.2 Gráfico do número de frases em função do WER obtidos pelos 3 sistemas de reconhecimento de fala. 56
4.3 Gráfico da precisão do novo sistema para cada tipo de frase. 63
4.4 Gráfico da cobertura (*recall*) do novo sistema para cada tipo de frase. 64
4.5 Distribuição em função do WER das frases para as quais não foram tomadas decisões. 69
4.6 Histograma das relações semânticas extraídas em função do WER das frases. 70

A.1 Arquitectura geral de um sistema de aprendizagem baseada em memória . . . 84
A.2 Esquema da TBL. 85

Lista de Tabelas

1.1 Ferramentas utilizadas nas diferentes versões da interface de fala do Carl. . . 8

2.1 Algumas entradas do dicionário de pronúncia utilizado neste trabalho. 24
2.2 Quantidade de palavras ambíguas no Brown Corpus segundo DeRose (1988). Fonte Jurafsky e Martin (2000, pg. 299). 28

3.1 Tipos de mensagem HRCL presentemente implementados no robô e breve descrição de cada um deles. Fonte Seabra Lopes *et al.* (2003a). 41
3.2 Conjunto de etiquetas morfossintácticas do Penn Treebank. 45
3.3 Primeiras 10 transformações não lexicais obtidas com o *corpus* do WSJ. . . . 46
3.4 Primeiras 10 transformações obtidas para palavras desconhecidas, obtidas com o *corpus* do WSJ. 47

4.1 Quantidade de frases de cada tipo presentes no conjunto de teste aos sistemas de reconhecimento de fala. 54
4.2 Número de frases bem reconhecidas por cada sistema de reconhecimento de fala. 54
4.3 Resultados médios por frase dos erros cometidos pelos sistemas de reconhecimento de fala. 57
4.4 Quantidade de frases de cada tipo presentes no conjunto de teste aos sistemas de compreensão de linguagem natural. 59
4.5 Decisões tomadas para o tipo de frase, independentemente da validade do resultado. 60
4.6 Decisões tomadas acerca dos tipos das frases vindas de todos os reconhecedores de fala. 62
4.7 Decisões tomadas sobre os tipos de frases com WER inferior a 25%. 66
4.8 Resultados da precisão, *recall* e *F-measure* para as frases com WER inferior a 25%. 66
4.9 Decisões tomadas sobre os tipos de frases com WER inferior a 50%. 67
4.10 Resultados da precisão, *recall* e *F-measure* para as frases com WER inferior a 50%. 67
4.11 Decisões tomadas sobre os tipos de frases com WER inferior a 75%. 68
4.12 Resultados da precisão, *recall* e *F-measure* para as frases com WER inferior a 75%. 69
4.13 Quantidade de relações semânticas extraídas das frases bem reconhecidas. . . 69

4.14 Resumo dos resultados obtidos com três configurações representativas. 71

Glossário

AAAI American Association for Artificial Intelligence.

ACL Agent Communication Language.

ALE Attribute Logic Engine.

APSG Augmented Phrase Structure Grammar.

BNF Backus Naur Form.

CARL Communication Action Reasoning and Learning in robotics.

CMU Carnegie Mellon University.

CPK-NLP Center for PersonKommunikation - Natural Language Processing.

DARPA Defense Advance Research Projects Agency.

FCT Fundação para a Ciência e Tecnologia.

FOL First Order Logic.

FSG Finite State Grammar.

GLR Generalized Left-Right.

HAPI HTK Application Programming Interface.

HMM Hidden Markov Model.

HRCL Human-Robot Communication Language.

HTK HMM Toolkit.

IBL Instruction-Based Learning.

IEETA Instituto de Engenharia Electrónica e Telemática de Aveiro.

***k*-NN** *k*-Nearest Neighbor.

LL Left to right, Leftmost derivation.

LPC Linear Predictive Coding.

LR Left to right, Rightmost derivation.

LTPOS Language Technology Part-Of-Speech.

MBL Memory Based Learning.

MDP Minimum Distance Parsing.

NLG Natural Language Generation.

NLP Natural Language Processing.

NLU Natural Language Understanding.

POS Part-Of-Speech.

Sphinx CT Sphinx 2 com treino.

Sphinx ST Sphinx 2 sem treino.

SLP Spoken Language Processing.

SMAPI Speech Manager Application Programming Interface.

SSN Simple Synchrony Networks.

TBL Transformation Based Learning.

TiMBL Tilburg Memory Based Learner.

WER Word Error Rate.

WSJ Wall Street Journal.

Capítulo 1

Introdução

"At last, a computer that understands you like your mother."
anúncio da McDonnell-Douglas, 1985

A linguagem é um sistema ou conjunto de sinais, fonéticos ou visuais, que servem para a expressão do pensar e do sentir (Priberam, 2004). É consensual na sociedade e na comunidade científica que a linguagem foi e é o grande motor da evolução da humanidade. Sem uma linguagem comum seria impossível guardar e transmitir de forma eficiente o conhecimento humano adquirido ao longo dos séculos. A actual forma de linguagem entre os seres humanos é fruto de milhões de anos de desenvolvimento do seu cérebro. É uma linguagem simultaneamente precisa e flexível, pois é possível definir de forma rigorosa todo e qualquer conceito, raciocínio, sentimento ou emoção.

Nos dias de hoje, os computadores e equipamentos electrónicos obrigam o utilizador a aprender formas não intuitivas de comunicação, através de comandos precisos, linguagens de programação, menus, hiper-ligações e botões. As interfaces entre as máquinas e os seres humanos têm vindo a ficar mais sofisticadas, caminhando aos poucos em direcção a formas mais humanas de comunicação, mas ainda são necessários comandos do tipo "cp ./ficheiro.txt /mnt/floppy" para copiar um ficheiro para uma disquete, ou pode ser feito de forma gráfica, não sendo ainda comum poder-se dizer "computador, copia para a disquete o texto que acabei de escrever".

Linguagens de programação como Java, C, PERL, LISP, Prolog, etc., são precisas porque contêm regras rígidas e estruturas lógicas bem definidas, que permitem à máquina saber exactamente o que fazer mediante um determinado comando. No entanto, a forma de comunicação entre os Homens não é tão estruturada, podendo conter ambiguidades cujas interpretações dependem do contexto, do conhecimento do mundo, de regras gramaticais, culturais e de conceitos abstractos (Insite, 2001). A esta forma de comunicação designamos de linguagem natural. Com vista a tornar mais fácil a utilização dos computadores é necessário desenvolver métodos e ferramentas de *software* que permitam a este analisar e interpretar frases representadas em linguagem humana. Esse conjunto de métodos é designado por compreensão de linguagem natural, em Inglês Natural Language Understanding (NLU).

Com a difusão de interfaces em linguagem natural, mais pessoas terão acesso a tecnologias que de outra forma seriam complicadas de utilizar. Por outro lado, uma máquina dotada de NLU tem a vantagem de poder aceder, processar e organizar maiores quantidades de informação, uma vez que a esmagadora maioria do conhecimento da humanidade está registado em linguagem natural.

Outra característica desejável nas máquinas é a capacidade de comunicação à distância porque liberta as pessoas de terem de permanecer perto delas. Um dos exemplos mais difundidos da utilidade/aceitação desta capacidade são os comandos à distância de electrodomésticos como os aparelhos de televisão, os vídeos, etc.. Mais prático ainda seria não ser necessário qualquer dispositivo para comandar à distância. A fala permite interagir à distância suprimindo a necessidade de um aparelho que sirva de interface. A junção das vantagens da linguagem natural e da comunicação mãos livres torna a linguagem natural transmitida por fala ideal para interacção com as máquinas.

1.1 Comunicação Humano-Robô

Na comunicação Humano-Robô, à semelhança do que acontece para outras máquinas, uma questão relevante é o meio por qual deve ser veiculada a informação. Como é sabido, a fala é o principal meio de comunicação entre os Homens. Por este facto e pelas vantagens enunciadas, seria desejável que se pudesse comunicar com os robôs utilizando principalmente a fala. Na verdade a linguagem natural falada parece ser a única interface suficientemente flexível e amigável para interagir com robôs inteligentes (Seabra Lopes e Teixeira, 2000).

A fala permite que o utilizador fique com as mãos livres enquanto comunica, não requer que os intervenientes no diálogo estejam em linha de vista nem tão pouco em contacto, o que no caso da interacção com robôs móveis ou com partes móveis pode ser uma vantagem importante. No entanto o reconhecimento de fala ainda não é muito eficaz mesmo em ambientes sossegados, piorando o seu desempenho com o aumento do ruído ambiente. Torna-se por isso importante desenvolver técnicas de NLU robustas ao ruído que pode existir nas frases reconhecidas. Por ruído consideram-se palavras mal reconhecidas, em falta na frase ou até mesmo frases mal formadas pelo orador.

De um modo geral, as interfaces de linguagem natural falada são compostas por um reconhecedor de fala seguido por um módulo de compreensão de linguagem natural. A abordagem clássica à compreensão de linguagem natural consiste em primeiro determinar a estrutura sintáctica da frase e posteriormente extrair a informação semântica com base na estrutura sintáctica obtida.

1.2 CARL

Um dos projectos pertencentes à área transversal de robótica inteligente do Instituto de Engenharia Electrónica e Telemática de Aveiro (IEETA) é o Communication Action Reasoning

and Learning in robotics (CARL).

O projecto CARL teve início em Julho de 1999, com financiamento da Fundação para a Ciência e Tecnologia (FCT). Pretendia-se desenvolver um robô capaz de entender instruções expressas sob a forma de conceitos familiares ao utilizador humano.

O robô do projecto CARL, designado Carl, está a utilizar como base a plataforma móvel *Pioneer 2 - DX* da ActivMedia Robotics. Esta plataforma contém entre outras coisas: três rodas (duas motrizes e uma direccional), 16 sonares repartidos em dois conjuntos de 8 (um conjunto à frente e outro atrás), pára-choques com sensores de colisão, uma câmara de vídeo e um computador baseado no processador Pentium 266 MMX, com 64 MB de memória RAM e o sistema operativo Linux instalado. Por cima desta plataforma foi montada uma estrutura em fibra de vidro que aumenta a altura do Carl para aproximadamente 110 centímetros. A estrutura em fibra de vidro alberga 10 sensores infravermelhos, um vector de microfones VoiceTracker da Acoustic Magic, uma coluna de som e um computador portátil baseado no processador Pentium III com 256 MB de memória RAM e o sistema operativo Linux instalado. Os dois computadores comunicam entre si através de uma ligação Ethernet. O robô utiliza a informação dos diversos sensores e odometria para navegar em qualquer ambiente.

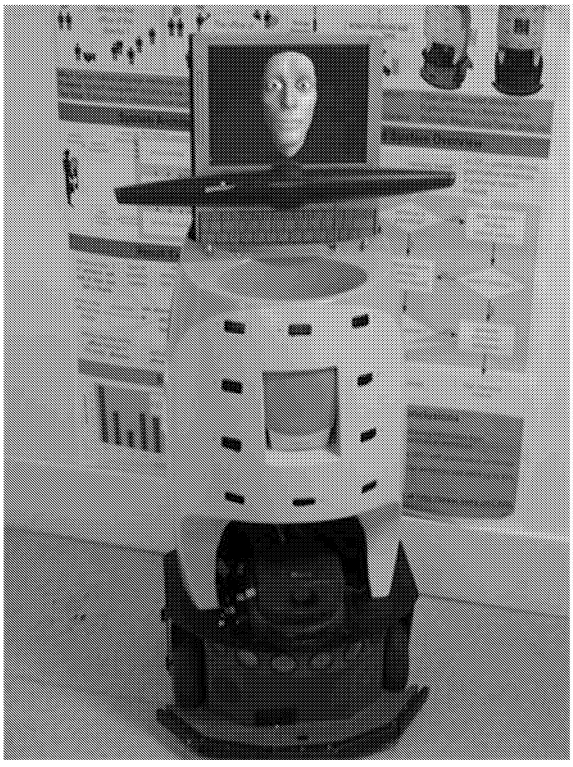


Figura 1.1: Fotografia do robô Carl.

1.3 Evolução da Interface de Fala do Carl

Nesta secção vai ser feita uma breve descrição da evolução da interface em linguagem natural falada do robô do projecto CARL. O objectivo é enquadrar o leitor com as dificuldades que sucessivamente foram surgindo no desenvolvimento da interface de fala, de modo a que se compreendam melhor os problemas que se pretendem resolver com a abordagem descrita nesta dissertação.

As sucessivas versões da interface de fala do Carl foram desenvolvidas por alunos em estágio final de Licenciatura e pelo autor, também como aluno em estágio final de Licenciatura, como bolseiro de investigação e a versão em discussão nesta dissertação como aluno de Mestrado.

Primeira Versão

A primeira interface de fala era composta por:

reconhecedor de fala - *grapHvite* da Entropic;

ferramentas para processamento de linguagem natural - Center for PersonKommunikation - Natural Language Processing (CPK-NLP) da universidade de Aalborg;

sintetizador de fala - *ViaVoice OutLoud* da IBM.

O reconhecedor de fala foi construído utilizando a biblioteca de funções HTK Application Programming Interface (HAPI) (Odell *et al.*, 1998) distribuída com o *grapHvite* (Power *et al.*, 1998). Este necessitava de uma gramática de reconhecimento para poder funcionar. Inicialmente esta gramática foi elaborada utilizando o *NetBuilder*, também distribuído com o *grapHvite*, consistindo em 5 comandos para orientação interactiva da navegação do robô (Ferreira e da Silva Ferreira, 2000). A aquisição de som era efectuada por um microfone sem fio que o utilizador tinha de transportar, ficando o receptor ligado ao Carl.

Para efectuar o *parsing* sintáctico foi escolhido o CPK-NLP (Brøndsted, 1999). O CPK-NLP precisava de uma gramática de *parsing* definida no formalismo de unificação Augmented Phrase Structure Grammar (APSG). Para se garantir a coerência entre o reconhecedor de fala e o módulo de *parsing* sintáctico, optou-se por definir apenas uma gramática no formalismo APSG, utilizada directamente pelo CPK-NLP e convertida para o formato aceite pelo *grapHvite*.

A síntese de fala não foi muito trabalhada nesta fase devido à falta de tempo. A voz utilizada pelo sintetizador era a que vinha pré-configurada uma vez que era uma voz aceitável. As frases a sintetizar pelo robô encontravam-se explicitamente escritas algures no programa. Para cada pergunta existia um conjunto possível de respostas, sendo a resposta uma frase escolhida aleatoriamente desse conjunto.

Este sistema tinha a vantagem de garantir que qualquer frase reconhecida tinha uma análise válida e detalhada, em virtude da gramática para o *parsing* sintáctico ser a mesma que

a utilizada no reconhecimento de fala. Uma das falhas da altura era o facto de a gramática ainda ser pouco abrangente na medida em que permitia apenas a interpretação de um conjunto pequeno de frases (aproximadamente 3500), mas era suficiente para o papel que o Carl deveria desempenhar: anfitrião no IEETA (Rodrigues, 2000).

Ainda foi tentado o treino dos modelos acústicos do reconhecedor de fala. Infelizmente não foi possível porque uma função da HAPI necessária para essa tarefa não funcionava correctamente. Tentou-se contactar o suporte da Entropic em vão. Esta companhia tinha sido adquirida pela Microsoft, não existindo mais suporte ao *grapHvite* e seus componentes.

Segunda Versão

Numa segunda versão da interface de fala, alteraram-se dois aspectos em relação à anterior: o número de frases aceites pela gramática foi aumentado, mudou-se o reconhecedor de fala devido a não se conseguir treinar os modelos acústicos do anterior e à não existência de suporte ao *grapHvite*. Procedeu-se também a uma actualização do sintetizador de fala. Assim a segunda versão da interface era constituída por:

reconhecedor de fala - *ViaVoice* da IBM;

ferramentas para processamento de linguagem natural - CPK-NLP da universidade de Aalborg;

sintetizador de fala - *ViaVoiceTTS* da IBM.

O novo reconhecedor de fala foi desenvolvido utilizando a biblioteca de funções Speech Manager Application Programming Interface (SMAPI), distribuída com o IBM *ViaVoice* (ViaVoice, 2000; SMAPI, 2000a,b). Novamente com o intuito de garantir a coerência entre o reconhecedor de fala e o módulo de *parsing* sintáctico, desenvolveu-se uma ferramenta que convertia a gramática escrita no formato APSG para o formato aceite pelo *ViaVoice*, o formato Finite State Grammar (FSG). Com este novo sistema de reconhecimento de fala foi possível treinar os modelos acústicos existentes. Foi criado um novo modelo acústico baseado nos originais de Inglês dos Estados Unidos da América e ajustado para a pronúncia inglesa dos oradores portugueses.

Com o intuito de participar no concurso "AAAI Robot Competition: Hors d'Oeuvres Anyone?", organizado pela American Association for Artificial Intelligence (AAAI) (Michaud e Gustafson, 2002), o conjunto de frases admitido pela gramática do Carl foi aumentado (Rodrigues *et al.*, 2002). O objectivo dos robôs participantes no concurso era servir canapés a participantes na recepção, devendo interagir e dialogar com eles. Assim foram incluídas na gramática do Carl frases que lhe permitiam, por exemplo, oferecer comida às pessoas. Para facilitar a utilização do Carl no concurso mudou-se de microfone, uma vez que o microfone anterior para ser utilizado teria de ser passado de pessoa para pessoa. O novo microfone ficou fixo no robô. Assim qualquer pessoa que quisesse falar com o Carl apenas precisava de se colocar diante dele e falar. Para que o Carl captasse o som emitido pelo utilizador captando

o mínimo som envolvente, foi escolhido um vector de microfones que utilizava a informação vinda de cada elemento do vector para filtrar e obter principalmente o som proveniente de um interlocutor à sua frente.

A título de curiosidade, o Carl classificou-se em 3^o lugar no concurso. No entanto, apesar deste bom resultado, existiram algumas dificuldades:

- mesmo com o vector de microfones a captar maioritariamente o som proveniente da sua frente, o ruído ambiente era muito elevado e nem o Carl nem nenhum outro robô conseguiram reconhecer satisfatoriamente o que as pessoas diziam;
- as pessoas dificilmente acertavam nas frases aceites pela gramática. Este é aliás um dos pontos negativos da abordagem estritamente por gramáticas, uma vez que se o utilizador não acertar exactamente numa frase aceite, não há hipótese de o robô poder interpretar a frase proferida.

Terceira Versão

Em resposta aos problemas evidenciados no concurso da AAAI, tentou-se utilizar uma nova abordagem no Carl, surgindo desta forma a terceira versão, actualmente em utilização, da interface de fala do robô. A filosofia desta versão consiste em tentar reconhecer tudo, ou quase tudo, que seja pronunciado pelo utilizador. Como consequência teve de ser abandonada a abordagem anterior baseada numa gramática rígida, que obrigava o utilizador a apenas poder enunciar as frases contidas na gramática.

Foi efectuado um teste ao desempenho do *ViaVoice* quando utilizado em modo de ditado, sem qualquer gramática podendo por isso ser reconhecida qualquer sequência de palavras existentes no léxico do sistema. Os resultados obtidos foram maus. O reconhecedor de fala só muito dificilmente conseguia acertar na frase pronunciada. A solução encontrada foi um compromisso entre as duas abordagens: a utilização de modelos de linguagem probabilísticos (Jurafsky e Martin, 2000). A gramática probabilística é formada por n-gramas, sendo cada n-grama uma sequência de n palavras com uma determinada probabilidade. Com esta solução é possível o reconhecimento de qualquer sequência de palavras do léxico, existindo também informação que faz com que algumas sequências de palavras sejam mais frequentes que outras.

Esta nova abordagem implicou novas ferramentas para a análise sintáctica, pois o CPK-NLP não se comporta bem com frases que não consigam ser representadas totalmente pelas estruturas sintácticas definidas. Também se deu uma nova mudança do reconhecedor de fala devido ao *ViaVoice* não suportar n-gramas e pelo fim do suporte *ViaVoice* por parte da IBM.

A terceira versão da interface de fala ficou composta por:

reconhecedor de fala - *Nuance 8.0* da Nuance (Nuance, 2001a,b,c);

ferramentas para processamento de linguagem natural - os anotadores Brill Tagger (Brill, 1995) e Language Technology Part-Of-Speech (LTPOS) (Mikheev, 1994), o clas-

sificador Tilburg Memory Based Learner (TiMBL) (Daelemans *et al.*, 2002) e o sistema Attribute Logic Engine (ALE) (Carpenter e Penn, 2001);

sintetizador de fala - *ViaVoiceTTS* da IBM (ViaVoiceTTs, 2000).

Construiu-se o novo reconhecedor de fala tendo por base a biblioteca de funções do *Nuance 8.0*. Depois definiu-se o modelo de linguagem através de bigramas, utilizando como conjunto de treino uma amostra de frases passíveis de serem ditas ao Carl.

Nesta fase ainda não era utilizada nenhuma ferramenta capaz de determinar a estrutura sintáctica de qualquer frase proveniente do reconhecedor de fala, porque este podia retornar qualquer sequência de palavras do léxico, por muito absurda que fosse. Para determinar se uma frase era correcta ou incorrecta foi utilizado o TiMBL pois tem a vantagem de funcionar com base em exemplos[1]. Dá-se um conjunto de treino composto por frases correctas e incorrectas, em que se diz explicitamente a que categoria pertence cada frase e o TiMBL é capaz de classificar qualquer frase que faça parte do seu conjunto de treino ou não. Para aumentar a informação disponível sobre cada frase, utilizou-se o Brill Tagger. Este determina a que categoria sintáctica pertence cada palavra de uma frase.

As frases consideradas correctas são enviadas para o módulo de determinação da estrutura sintáctica da frase, construído com base no ALE, sendo as frases consideradas incorrectas ignoradas. O ALE também extrai a informação semântica contida nas frases. No entanto o ALE é um sistema baseado numa gramática rígida, à semelhança do CPK-NLP, com a vantagem de ignorar as frases que não consegue processar. Assim, apesar do novo sistema ser capaz de reconhecer qualquer sequência de frases contidas no léxico, apenas consegue processar aquelas que são completamente definidas pela gramática presente no ALE, ou seja o problema da utilização de uma gramática rígida no sistema de NLU mantém-se.

Para se ter uma noção da quantidade de frases ignoradas, segundo Ferreira *et al.* (2002) e Seabra Lopes *et al.* (2003b) a taxa de frases correctas num teste com o *Nuance 8.0* utilizando bigramas foi de 47.8% enquanto a taxa de palavras correctas foi de 78.0%. Isto significa que muitas vezes apesar da frase reconhecida não ser a proferida pelo orador, a maioria das palavras estão correctas, devendo ser possível extrair alguma informação da frase.

Embora marginal para o presente trabalho, nesta terceira fase foi efectuado trabalho em geração de linguagem natural, em inglês Natural Language Generation (NLG) (Soares, 2002).

A Tabela 1.1 resume as ferramentas utilizadas nas diferentes versões da interface de fala do robô Carl.

1.4 Outros Robôs

Nesta secção vão ser brevemente apresentados alguns robôs interactivos com capacidade de entenderem linguagem natural falada.

[1]Este método é explicado com detalhe no Anexo A.1. Para já importa realçar que o TiMBL não necessita de "saber" nada sobre estruturas de frases.

Tabela 1.1: Ferramentas utilizadas nas diferentes versões da interface de fala do Carl.

Ferramentas	**Versão 1**	**Versão 2**	**Versão 3**
Reconhecimento de fala	grapHvite	ViaVoice	Nuance 8.0
Processamento de linguagem natural	CPK-NLP	CPK-NLP	Brill Tagger, LTPOS, TiMBL, ALE
Síntese de fala	ViaVoice OutLoud	ViaVoiceTTS	ViaVoiceTTS

1.4.1 Godot

Projecto desenvolvido na School of Informatics, University of Edinburgh no Reino Unido (Bos *et al.*, 2003), o Godot é um robô baseado na plataforma móvel *RWI Magellan Pro*, equipada com um computador de bordo a utilizar o sistema operativo Linux. Tem a forma cilíndrica com cerca de 50 centímetros de altura e 41 centímetros de diâmetro e vários sensores infravermelhos, de ultrasons, de colisão, de vídeo e de odometria que lhe permitem navegar no mundo real (ver Figura 1.2).

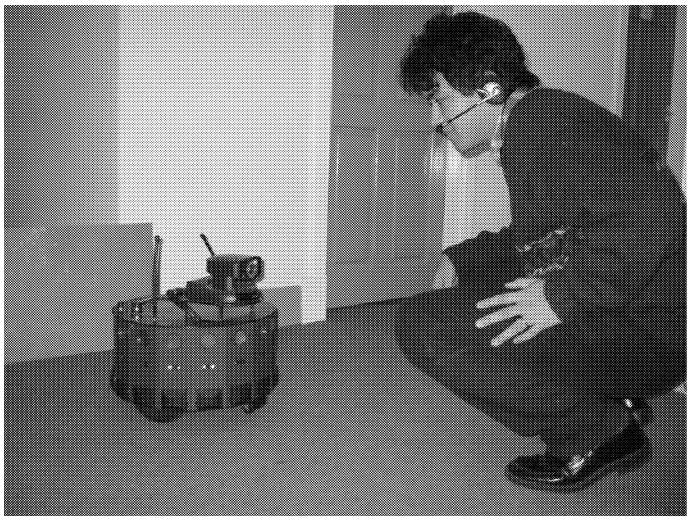


Figura 1.2: Fotografia do robô Godot e um utilizador. Fonte Godot (2003).

O sistema de diálogo está implementado sobre o Open Agent Architecture (Martin *et al.*, 1999), um sistema de agentes de *software* que pode ser executado de um modo partilhado e em diferentes plataformas. O sistema é composto por agentes para o reconhecimento de fala, Nuance 8.0 (Nuance, 2001c), para síntese de fala, Festival (Taylor *et al.*, 1998), para resolução de ambiguidades da linguagem e para gestão de diálogo. Para a resolução das ambiguidades utiliza o formalismo semântico *Discourse Representation Theory*. A inferência lógica é auxiliada por dois agentes: um é o construtor de modelos de conhecimento MACE (McCune, 2001), o outro é o demonstrador de teoremas SPASS (SPASS, 2004). Todo o sistema é coordenado por um gestor de diálogo.

1.4.2 IBL: Instruction-Based Learning for Mobile Robots

Com financiamento do UK Engineering and Physical Sciences Research Council, este projecto é desenvolvido conjuntamente pelo Centre for Interactive Intelligent Systems, University of Plymouth e pelo Institute for Communicating and Collaborative Systems, University of Edinburgh. O seu objectivo é explorar um método pouco estudado de aquisição de conhecimento e aprendizagem: aquisição de conhecimento através de instruções em linguagem natural. Os autores do projecto argumentam que este modo de aprendizagem é eficaz na aprendizagem humana e será essencial para a adaptação de sistemas inteligentes futuros às necessidades de qualquer utilizador (Bugmann *et al.*, 2001).

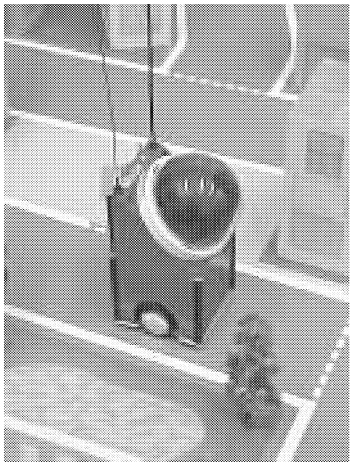


Figura 1.3: Fotografia do robô utilizado no projecto IBL. Fonte: IBL (2004).

O objectivo do projecto é investigar a aprendizagem baseada em instruções, em inglês Instruction-Based Learning (IBL), numa aplicação do mundo real como a aprendizagem de um percurso. Os utilizadores iniciam um diálogo com um robô móvel equipado com visão artificial, para lhe ensinar como navegar num ambiente labiríntico simplificado (ver Figura 1.3). Os autores defendem que este cenário experimental vai limitar problemas de percepção e controle e também a complexidade do processamento de linguagem natural.

Este projecto foca o problema de como instruções em linguagem natural podem ser utilizadas por um agente inteligente para construir uma hierarquia de funções complexas sobre um conjunto limitado de funções de baixo nível para percepção, acção e cognição. É estudado como as representações internas necessárias para o robô navegar e sentir o mundo podem suportar uma interface de fala utilizável. Devido à utilização de visão e fala artificiais, um sistema deste género pode auxiliar pessoas diminuídas visualmente e em cadeiras de rodas.

1.4.3 JIJO-2

O Jijo-2 é um projecto conjunto do Linguistics Department and CSLI da Stanford University dos Estados Unidos da América e do Real-World Intelligence Center do Electrotechnical Laboratory do Japão (Matsui *et al.*, 1999). O robô foi concebido para fornecer serviços de

escritório tais como responder a perguntas sobre a localização de pessoas, indicar e levar pessoas a determinados locais e entrega de objectos, com todas as informações a serem trocadas sob a forma de diálogos em linguagem natural.

Tem por base a plataforma móvel *Nomad 200* com um computador de bordo a executar o sistema operativo Linux. Possui vários sensores infravermelhos, de ultrasons, de colisão e odométricos. Está também equipado com um vector de microfones, um Digital Signal Processor para processar o sinal de som e duas câmaras de vídeo a cores (ver Figura 1.4). Para a síntese de fala utiliza o sintetizador para Japonês *Shaberimbo*. Parte do processamento da informação é efectuado fora do robô, com comunicação via rádio entre o robô e o computador exterior.

Figura 1.4: Fotografia do robô Jijo-2 e um utilizador. Fonte Matsui *et al.* (1999).

O sistema de reconhecimento de fala para Japonês, independente do orador, foi desenvolvido no laboratório do Jijo-2 (Matsui *et al.*, 1999). O robô tem um gestor de diálogo e um módulo de inferência que avalia o contexto do diálogo de modo a que se possa referir a actores do diálogo indirectamente e de uma forma natural.

1.4.4 Philips: Futebol Robótico

O futebol robótico da Philips é desenvolvido por investigadores do Centre for Industrial Technology, como um passatempo nas suas horas vagas (Peijnenburg, 2003). O *hardware* dos robôs foi construído à medida das necessidades, não tendo sido comprada nenhuma plataforma pronta a utilizar (ver Figura 1.5). Presentemente estão a desenvolver um sistema de reconhecimento de fala para os robôs. O objectivo é fazer com que os robôs tenham comportamentos mais parecido com o dos humanos, tornado-os mais agradáveis de ver a funcionar

e de trabalhar. Isto também vai permitir futuras interacções com treinadores humanos e no futuro com jogadores.

Está previsto que um operador humano utilize um microfone para falar com os robôs, sendo as respostas destes dadas através do sintetizador de fala Festival. O reconhecedor de fala foi desenvolvido pela secção de investigação da Philips em Aachen. O diálogo, a gramática e os modelos acústicos foram concebidos especificamente para os robôs. O sistema é independente do orador e por isso qualquer membro da equipa pode utilizá-lo. O vocabulário é limitado e o diálogo é restrito no sentido que tem de seguir certos passos interactivos para descrever um comando. No entanto o orador pode se exprimir em linguagem natural ao falar com o robô.

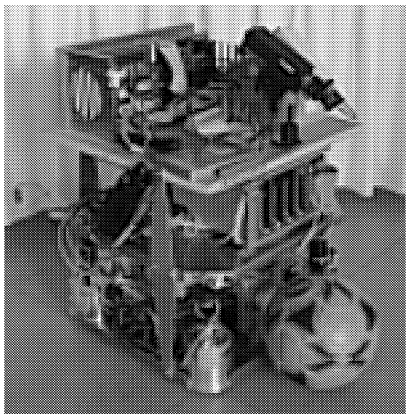


Figura 1.5: Fotografia de um robô da Philips com uma bola. Fonte Philips (2002).

1.4.5 Robita

O projecto Robita (Real-world Oriented BI-modal Talking Agent) foi desenvolvido pelo Perceptual Computing Group, Waseda University do Japão (Fujie *et al.*, 2004). Tem por objectivo o desenvolvimento de um sistema de conversação apto a participar numa conversa de grupo (ver Figura 1.6). Numa conversa de grupo existem alguns desafios que não surgem na conversação de um para um: detecção de quem está a falar e para quem e definição de estratégia para entrar na conversa.

No Robita esses problemas foram resolvidos com recurso a interfaces multi-modais: reconhecimento de cara, detecção da direcção da cara, detecção da origem do som, reconhecimento de fala e utilização dos braços do robô. Os autores do projecto defendem que a utilização sistemática e combinada destas capacidades tornam o diálogo amigável.

1.5 Objectivos

Os sistemas de compreensão de linguagem podem ser caracterizados pela precisão, quantidade de informação correcta que o sistema consegue obter, e pela cobertura (*recall*), quantidade de informação relevante extraída. Apesar de não serem características antagónicas, verifica-se

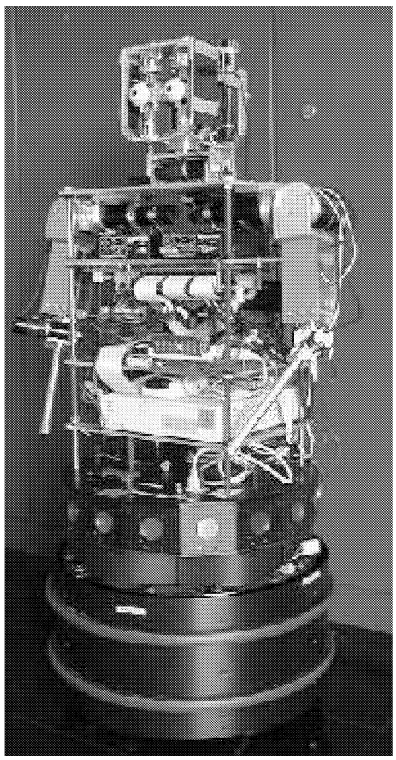


Figura 1.6: Fotografia do robô Robita. Fonte Robita (2002).

que sistemas com boa precisão têm uma fraca cobertura e vice-versa. Também é geralmente assumido que as frases a analisar pelos sistemas são correctas do ponto de vista sintáctico e semântico. Contudo os reconhecedores de fala não são 100% fiáveis, nem a linguagem falada é sempre correcta. Por isso, quando um sistema tenta analisar frases enunciadas oralmente, deve estar preparado para extrair informação de frases mal estruturadas.

O objectivo deste trabalho é conceber uma solução que, aliando uma boa precisão a uma boa cobertura, consiga extrair informação correcta, de frases correctas ou não. O sistema deverá ser capaz de extrair informação detalhada, conseguindo-o fazer para um vasto conjunto de frases obtidas por um sistema de reconhecimento de fala. O trabalho desenvolvido deverá ser implementado no robô Carl. O Carl é uma boa plataforma de teste pois a tarefa de perceber frases enunciadas oralmente é dificultada pelos ambientes ruidosos por onde navega, e pelo facto de interagir com pessoas de várias nacionalidades a falar Inglês, sendo por isso frequente os modelos acústicos do reconhecedor de fala estarem desajustados do utilizador.

Do ponto de vista do projecto CARL, este trabalho é vantajoso pois deverá aumentar o leque de frases que o robô pode entender. Quanto mais frases o Carl compreender, mais reactivo e comunicativo se torna, ficando mais agradável de se utilizar. Se o robô não extrair qualquer informação de uma frase, não sabe o conteúdo da mensagem e por isso não sabe o que deverá fazer, permanecendo sem reacção.

A solução proposta neste trabalho para compreender as frases mal reconhecidas e mal estruturadas passa pela concepção de um módulo de compreensão de linguagem natural apto

a extrair informação quer de frases correctas quer de frases com erros. O módulo a desenvolver além de aumentar o número de frases compreendidas deve fazê-lo mantendo um nível elevado de qualidade de análise, isto é, deverá ser capaz de extrair o máximo de informações correctas de frases correctas e incorrectas. Para tentar atingir esse objectivo trabalhou-se essencialmente na NLU.

1.6 Estrutura da Dissertação

Após introduzir o assunto e objectivos a que nos propomos no presente capítulo, no segundo capítulo apresenta-se a abordagem habitual ao processamento de linguagem natural falada. Primeiro são descritos os tipos de reconhecedores de fala existentes e a teoria em que se apoiam os reconhecedores de fala modernos. Seguidamente são apresentadas as diferentes abordagens para resolução do problema de processamento de linguagem natural. Por fim são descritas duas técnicas empregues no processamento de linguagem natural utilizadas no decurso do trabalho aqui apresentado: a anotação morfossintáctica e o *parsing* robusto. Para cada uma destas técnicas são discutidas as diferentes abordagens existentes.

No capítulo três é apresentado o sistema desenvolvido. Primeiro é justificada a abordagem escolhida, depois é efectuada uma descrição detalhada ao funcionamento do sistema, terminando com um exemplo real que visa ajudar à compreensão do que foi descrito. Na parte final do capítulo são apresentadas as ferramentas utilizadas no decorrer do trabalho.

No quarto capítulo são apresentados os testes efectuados. Seguem-se os resultados obtidos e uma breve discussão dos mesmos.

O documento é finalizado, no capítulo cinco, com um resumo do trabalho apresentado, os principais resultados obtidos e sugestões para trabalho futuro.

1.7 Resultados Publicados

As soluções encontradas e os resultados obtidos com as diversas versões da interface de fala do Carl foram alvo de algumas publicações.

Na primeira versão da interface, em que o autor colaborou na qualidade de aluno em estágio final de Licenciatura em Matemática Aplicada à Tecnologia com uma equipa formada por dois alunos em projecto final de Licenciatura em Engenharia Electrónica e Telecomunicações, a publicação focava a arquitectura geral da interface de fala do robô, referindo as soluções encontradas para o reconhecimento de fala, análise sintáctica e síntese de fala (Rodrigues, 2000; Ferreira e da Silva Ferreira, 2000). O documento foi publicado na revista do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro (Ferreira *et al.*, 2001).

No Robótica 2002, Festival Nacional de Robótica, foi publicada uma comunicação sobre a segunda versão da interface de fala do Carl (Rodrigues *et al.*, 2002). O trabalho reportado focou a mudança do reconhecedor de fala e o treino dos seus modelos acústicos, a melhoria

e expansão da gramática utilizada pelo reconhecedor de fala e a experiência que tivemos no concurso "AAAI Robot Competition: Hors d'Oeuvres Anyone?". O desenvolvimento desta segunda versão, em que a análise sintáctica era uma evolução da anterior e o reconhecedor de fala era novo, foi efectuado maioritariamente pelo autor como bolseiro.

Da terceira versão da interface de fala do Carl resultaram várias publicações. Esta versão foi desenvolvida por dois alunos em estágio final de licenciatura, pelo autor como bolseiro e por outro grupo de trabalho do projecto CARL. No documento publicado no Robótica 2003, 3^o Festival Nacional de Robótica (Teixeira *et al.*, 2003), foram descritos os testes e resultados a um modelo linguístico do reconhecedor de fala definido por bigramas, à anotação morfossintáctica, à aprendizagem baseada em memórias e à geração de frases para a síntese de fala. No Eurospeech 2003, principal conferência na área da fala, foram comunicadas algumas soluções encontradas com base nos testes descritos no Robótica 2003 e foi descrito brevemente o modo de funcionamento do ALE (Seabra Lopes *et al.*, 2003b). No EFTA 2003 foi descrito com maior detalhe o sistema baseado no ALE e também foram mencionados testes efectuados para NLG (Seabra Lopes *et al.*, 2003a).

Já foram publicados alguns resultados preliminares sobre o trabalho aqui apresentado no "1st International Workshop on Natural Language Understanding and Cognitive Science", um *workshop* associado à ICEIS2004, 6^{th} International Conference on Enterprise Information Systems (Rodrigues *et al.*, 2004). Na altura da comunicação apenas tinha sido efectuada uma parte dos testes, mas os resultados obtidos já mostravam a capacidade de o novo sistema processar frases incorrectas, alargando o leque de frases ao alcance da compreensão do Carl. O poster foi apresentado no dia 14 de Abril do corrente ano, tendo várias pessoas demonstrado interesse no trabalho exposto.

Capítulo 2

Processamento de Linguagem Natural Falada

O processamento de linguagem natural falada, em inglês Spoken Language Processing (SLP), tem como objectivo interpretar frases obtidas por um reconhecedor de fala em linguagem natural. A filosofia da SLP difere daquela presente no processamento de linguagem natural, em inglês Natural Language Processing (NLP), na medida em entra em linha de conta com o desempenho imperfeito dos sistemas de reconhecimento de fala.

Como se pode ver na Figura 2.1, um sistema de SLP tem por entrada um sinal de fala do orador. Esse sinal é captado por um microfone e processado por um reconhecedor de fala. Este último gera um conjunto de hipóteses que representa as palavras que mais provavelmente foram pronunciadas. A representação desse conjunto pode assumir diversas formas como uma ou mais sequências ordenadas de palavras, malhas de palavras, etc... O conjunto de hipóteses é de seguida encaminhado para um módulo de compreensão de linguagem natural que tem por objectivo determinar a informação contida na frase.

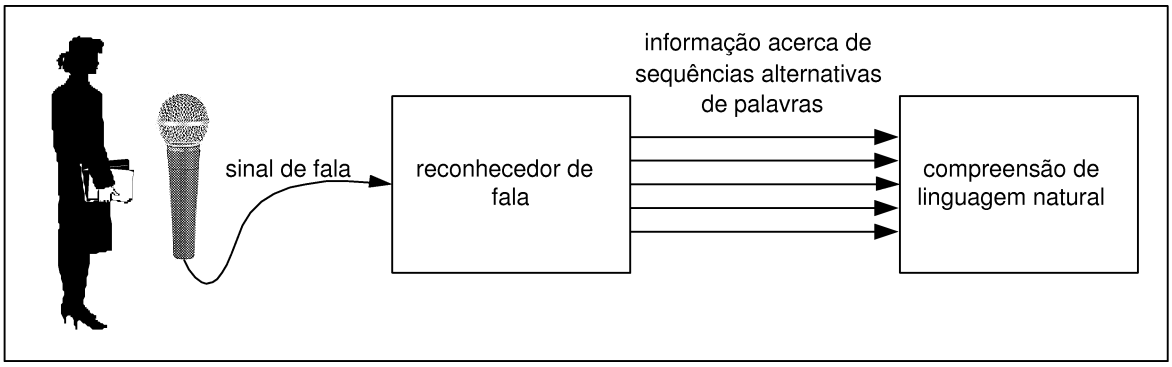


Figura 2.1: Arquitectura geral de um sistema de processamento de linguagem natural falada.

Uma característica destes sistemas, tal como se pode verificar na Figura 2.1, é que usualmente não têm por entrada uma única hipótese de frase tal como os sistemas de NLP. Esta opção deriva do facto de um reconhecedor de fala não poder determinar com toda a certeza

qual a frase que foi pronunciada. Para minimizar este problema, os reconhecedores de fala retornam informação relativa sobre as várias hipóteses de sequências alternativas de palavras que julgam ter sido ditas. A vantagem desta configuração é que, assim, o módulo de NLU pode utilizar uma maior quantidade de informação por forma a obter uma análise mais correcta para a frase.

Geralmente, um sistema de NLP, ao deparar-se com uma frase que não esteja totalmente de acordo com a sua especificação gramatical, ignora-a, não retornando qualquer análise. Este comportamento não é o mais apropriado para processar frases originadas por um reconhecedor de fala, uma vez que as frases enunciadas oralmente contêm frequentemente segmentos não gramaticais nomeadamente repetições, falsos inícios, pausas com ruído e expressões próprias de cada indivíduo ou até próprias de uma região. As frases são também frequentemente mal reconhecidas sempre que a língua não é a nativa do orador e/ou este está inserido num ambiente ruidoso. Para processar frases desta natureza são necessários componentes imunes a vários tipos de erros, que consigam obter a análise mais completa possível para cada frase (Rosé e Lavie, 2001). Em virtude disto, a SLP tem por objectivo obter de qualquer frase uma análise para a máxima porção possível desta. Se não for possível obter uma análise para a frase completa, o sistema deve tentar encontrar o maior subconjunto de palavras para o qual consegue obter um resultado válido.

2.1 Reconhecimento de Fala

Um reconhecedor de fala é um sistema que faz a correspondência entre sinais sonoros e sequências de caracteres que representam palavras e frases. O grande objectivo é transcrever para texto as frases comunicadas oralmente pelos utilizadores destes sistemas.

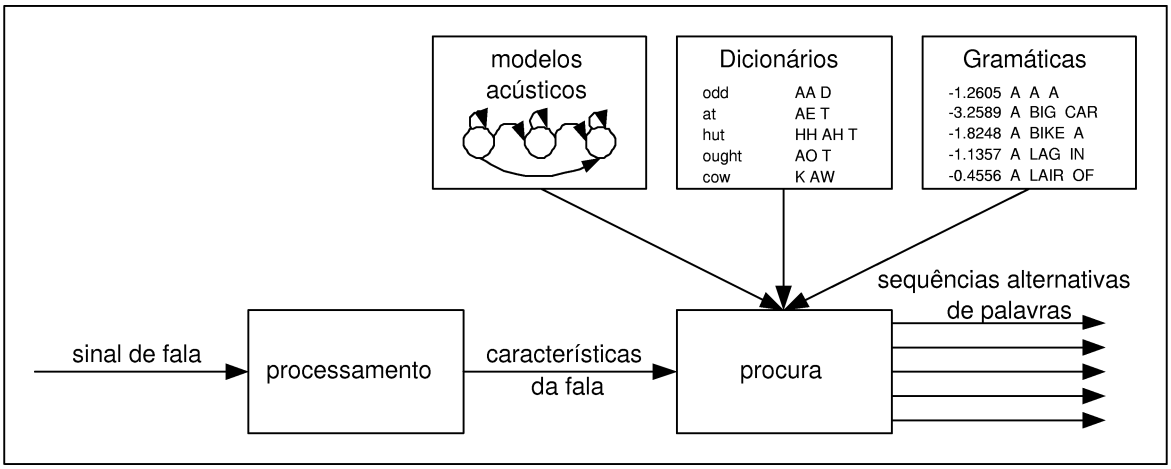


Figura 2.2: Arquitectura geral de um sistema de reconhecimento de fala.

Pode considerar-se que o reconhecimento da fala se processa em duas fases. Primeiro as características da fala são extraídas do sinal sonoro. Depois é efectuada a procura das palavras presentes no dicionário cujos modelos acústicos melhor reflectem as características extraídas

do sinal. A informação presente na gramática é utilizada para se determinar as sequências de palavras mais prováveis.

De seguida vai ser referida a classificação destes sistemas em função de algumas das suas características. Depois vai ser discutido com maior detalhe o modo como se processa o reconhecimento de fala.

2.1.1 Classificação dos Reconhecedores de Fala

Os sistemas de reconhecimento de fala podem ser caracterizados quanto ao tipo de discurso como:

- reconhecedor de comandos - reconhece apenas palavras isoladas ou frases muito sintéticas como por exemplo "turn left". O seu vocabulário é muito limitado, tendo normalmente menos de 20 palavras;
- reconhecedor de texto livre - reconhece sequências de palavras isoladas em que utilizador necessita de fazer breves pausas entre as palavras. O vocabulário utilizado pode ser bastante extenso, contendo mais de 20 000 palavras;
- reconhecedor de discurso contínuo - reconhece frases pronunciadas de forma natural, isto é, o utilizador não precisa de fazer pausas deliberadas entre palavras. Existem sistemas deste tipo que toleram falsos inícios, hesitações e pausas. Podem inclusivamente tolerar que duas palavras seguidas sejam ditas como se fossem uma única palavra formada por concatenação.

Podem também ser caracterizados quanto ao orador como:

- dependente do orador - o utilizador antes de utilizar o sistema pela primeira vez deve ler alguns textos pré-definidos, de modo a que os modelos acústicos do reconhecedor sejam adaptados a ele. Estes sistemas têm normalmente um bom desempenho se o orador o utilizar com o modelo acústico e em condições idênticas às de treino: microfone igual, mesmo ruído ambiente, mesma placa de som, etc.;
- independente do orador - os modelos acústicos destes reconhecedores são treinados para uma população alvo. Cada novo utilizador não necessita de adaptar a si os modelos acústicos do sistema. Normalmente o desempenho destes sistemas não é tão bom como os dependentes do orador, mas têm a vantagem de qualquer novo utilizador poder iniciar a utilização imediatamente.

Os reconhecedores de fala também podem ser distinguidos quanto ao modelo de linguagem:

- texto livre - não possui qualquer modelo de linguagem. As palavras reconhecidas são independentes umas das outras, podendo ser reconhecida com igual probabilidade qualquer sequência de palavras;

- gramática probabilística - a gramática é definida segundo um modelo de n-gramas. Este modelo considera que a probabilidade de ocorrência de uma sequência de palavras tem uma determinada probabilidade. Com este modelo existem sequências mais prováveis que outras, mas geralmente qualquer sequência de palavras é possível. As probabilidades das sequências são treinadas com recurso a um *corpus*. Nos sistemas correntes n toma o valor 2 ou 3, os designados bigramas e trigramas;

- gramática determinística - a gramática é especificada por regras sintácticas. Existem vários formalismos para especificação das gramáticas. O reconhecedor de fala só reconhece frases que sejam completamente descritas pela gramática.

2.1.2 Fundamentos do Reconhecimento de Fala Contínua

O objectivo de um sistema de reconhecimento de fala é, dado um sinal de fala, determinar qual a sequência $\tilde{W}$ de palavras, $\tilde{W} = \{\tilde{w}_1, \tilde{w}_2, ..., \tilde{w}_{N_{\tilde{W}}}\}$, proferidas pelo orador. O bloco de extracção de características do som tem por entrada a realização acústica de $\tilde{W}$, e por saída uma sequência de características acústicas $X = x_1, x_2, ..., x_{N_X}$. Para a codificação das características acústicas, têm sido desenvolvidas representações que dão ênfase a características do sinal independentes do orador, desvalorizando características que são dependentes do orador. Depois de determinadas as características acústicas do sinal, o sistema procura a melhor estimativa $\hat{W}$ da sequência de palavras original $\tilde{W}$.

O problema da determinação da sequência de palavras que melhor descreve as características acústicas do sinal é traduzido por, do ponto de vista estatístico, encontrar a estimativa $\hat{W}$ que maximiza $P(W|X)$, em que $P(W|X)$ é a probabilidade *a posteriori* da sequência de palavras W, dada a sequência de características acústicas X:

$$\hat{W} = \arg\max_{W} P(W|X) \tag{2.1}$$

Para os sistemas de reconhecimento de fala contínua seria necessário calcular $P(W|X)$ para todos os modelos de todas as frases possíveis, tarefa impraticável para sistemas a funcionar em tempo real, ou seja com tempos de resposta na ordem das décimas de segundo. A determinação de $\hat{W}$ pode ser decomposta em probabilidades mais simples de calcular. Aplicando o teorema de Bayes (Childers, 1997, pgs. 28-30) à equação 2.1, obtemos:

$$\hat{W} = \arg\max_{W} \frac{p(X|W)}{P(X)} P(W) \tag{2.2}$$

onde $P(X)$ é a probabilidade *a priori* da sequência de vectores acústicos, sendo independente da sequência de palavras escolhida, $P(X|W)$ é a verosimilhança da sequência acústica X condicionada pela ocorrência da sequência de palavras W e $P(W)$ é a probabilidade *a priori* da sequência de palavras W ser pronunciada. Como o valor de $P(X)$ é apenas dependente da realização acústica, pode ser considerado um factor de escala e retirado da equação 2.2. Uma vez que é constante para todas as sequências de palavras, não influencia qual aquela que

maximiza o valor de $\hat{W}$. Assim $\hat{W}$ pode ser determinado pela expressão:

$$\hat{W} = \arg\max_{W} p(X|W)P(W) \tag{2.3}$$

Verifica-se a existência de dois factores distintos no reconhecimento de fala: $P(X|W)$ referente aos modelos acústicos e $P(W)$ referente aos modelos de linguagem.

De seguida vão ser descritos o processo de extracção das características acústicas do sinal de fala e os modelos acústicos e de linguagem mais usuais.

2.1.3 Extracção das Características Acústicas do Sinal de Fala

O sinal de fala é amostrado a uma frequência entre os 8KHz e os 20KHz, dependendo dos sistemas. Como o sinal de fala não tem informação significativa acima dos 10KHz, a amostragem do sinal acima dos 20KHz não traz benefícios significativos[1].

Como as altas frequências da fala têm uma magnitude relativamente baixa quando comparadas com as baixas frequências, devido ao efeito conjunto da excitação glotal e radiação que parecem decrescer 6dB por década, aplica-se ao sinal um filtro passa-alto. Esta operação designada de pré-ênfase tem por objectivo aumentar a robustez do reconhecimento de fala, uma vez que com uma menor diferença de magnitude entre frequências consegue-se estimar melhor os coeficientes do modelo Linear Predictive Coding (LPC).

Para se poder caracterizar um sinal qualquer, este deve ser estacionário. Caso contrário os parâmetros extraídos desse sinal podem não o representar correctamente, uma vez que poderão não se manter constantes ao longo de todo o sinal, tal como é assumido. Por esta razão o sinal de fala é normalmente dividido em segmentos de cerca de 20 milissegundos, pois neste intervalo de tempo considera-se que o sinal se mantém estacionário. Segmentos adjacentes têm normalmente uma sobreposição da ordem da centena de amostras. Toda a análise subsequente é efectuada para cada segmento de sinal.

O tracto vocal pode ser visto como um tubo ressoador com área de secção variável. Em virtude disso o espectro do som emitido é bem descrito por um modelo espectral só com pólos onde a cada frequência de ressonância, ou formante, deverá corresponder um par de pólos complexos conjugados. Por isso o modelo LPC é adequado para descrever sinais de fala pois é um modelo espectral só com pólos. Para sons com zeros como as vogais nasais, o modelo é menos adequado sendo necessário um maior número de pólos para modelar os zeros. A ideia subjacente ao modelo LPC é que uma amostra pode ser descrita, a menos de um erro, como uma combinação linear das p amostras anteriores (LPC de ordem p):

$$s(n) = a_1 \times s(n-1) + ... + a_p \times s(n-p) + e_n \tag{2.4}$$

em que $s(n), ..., s(n-p)$ são as amostras do sinal, $a_1, ..., a_p$ são os coeficientes do modelo

[1] Segundo o Teorema da Amostragem, a frequência mais alta de um sinal amostrado é igual a metade da frequência de amostragem (Proakis e Manulakis, 1996).

considerados constantes em cada segmento de sinal analisado e e_n é o erro. Normalmente a ordem do modelo LPC varia entre 10 e 16. Para uma frequência de amostragem de 10KHz podemos ter 5 formantes implicando um modelo de pelo menos ordem 10, 5 pares de pólos, um por formante. Se for utilizada uma ordem superior, dever-se-á obter uma melhor aproximação do espectro do sinal, mas com ordens superiores a 16 os ganhos na descrição do sinal não são muitos. É de referir que se deve utilizar um modelo de ordem par, para não obrigar o sistema a ter pólos reais.

Os coeficientes cepstrais são um conjunto muito importante de parâmetros pois são mais adequados para descrever o sinal de fala que os coeficientes do modelo LPC. O cepstro de um sinal é definido como a transformada de Fourier do logaritmo do seu espectro. A aplicação do logaritmo ao espectro do sinal de fala vai permitir separar facilmente o efeito do tracto, característico dos sons, e da fonte, característica relacionada com o falante: passam de factores multiplicativos a factores aditivos. Outra vantagem dos coeficientes cepstrais é poderem ser obtidos recursivamente a partir dos coeficientes do modelo LPC. Usualmente o número de coeficientes cepstrais é de $1.5 \times p$, em que p é a ordem do modelo LPC.

2.1.4 Modelos de Markov

A representação das palavras é construída com recurso a um inventário de modelos probabilísticos elementares. À sequência de parâmetros acústicos extraídos de cada segmento do sinal de fala é atribuído o modelo que tem maior probabilidade de gerar essa sequência. Os modelos probabilísticos associados às unidades linguísticas são, regra geral, representados sob a forma de modelos de Markov não observáveis, em Inglês Hidden Markov Model (HMM). Cada modelo representa uma unidade linguística básica que nos sistemas mais sofisticados são fones mas na realidade podem ser outras quaisquer, inclusivamente palavras inteiras. Cada fone utilizado no reconhecimento de fala é modelado no tempo por um HMM. Os modelos de palavras resultam da concatenação destes modelos de fones de acordo com um dicionário de pronunciações e as frases resultam da concatenação dos modelos de palavras conforme o modelo de linguagem utilizado (ver Figura 2.3).

Os HMMs são uma composição de dois processos estocásticos (aqui designados por Q e X): o processo Q representado por uma cadeia de Markov com N_q estados ($Q = \{q_1, q_2, ..., q_{N_q}\}$) e o processo X que representa uma sequência de variáveis de saída que tomam valores no espaço dos vectores de observações acústicas $x_1, x_2, ..., x_{N_x}$ e modelam os acontecimentos presentes no sinal de fala. Num HMM aplicado ao reconhecimento de fala, a cadeia de Markov não observável (Q) modela a estrutura temporal da fala enquanto o processo estocástico observável (X) modela a variabilidade espectral. Esta combinação já provou ser suficientemente poderosa para cobrir as mais importantes fontes de ambiguidade vocal e suficientemente flexível para permitir a realização de sistemas de reconhecimento com dicionários com dezenas de milhar de palavras. Basta verificar que os melhores sistemas de reconhecimento de fala actuais têm por base HMMs (Lamere *et al.*, 2003; Odell *et al.*, 1998; ViaVoice, 2000; SMAPI, 2000a; Nuance, 2001c).

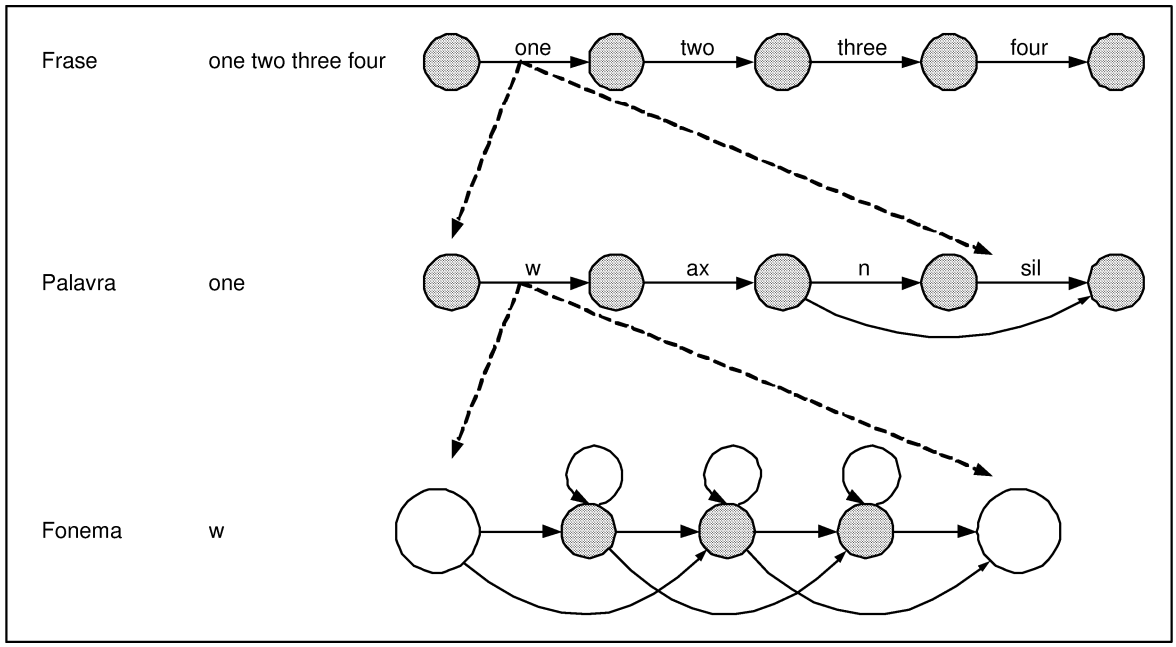


Figura 2.3: Esquema de como as unidades linguísticas básicas, representadas neste caso por HMMs com 5 estados, são encadeadas de modo a formarem palavras e frases. Fonte Teixeira e Vaz (2004).

A cadeia de Markov é um processo aleatório em que a probabilidade de se encontrar num determinado estado só depende de um número finito de estados anteriores (Childers, 1997). No caso de depender apenas do estado imediatamente anterior temos as cadeias de Markov de primeira ordem, muitas vezes designadas por cadeias de Markov. Cada estado define uma ocorrência, sendo os estados mutuamente exclusivos. A cadeia tem um estado inicial, um número finito ou finitamente numerável de estados e probabilidades de transição associadas a cada estado. Para o processamento de fala só interessam cadeias de Markov com um número finito de estados e discretas no tempo, nas quais só é possível mudança de estado em intervalos de tempo igualmente espaçados. Como cada estado define uma ocorrência, uma realização deste tipo de processos pode ser descrita pela sequência dos estados observados ao longo do tempo, sabendo-se a cada instante o estado em que o sistema se encontra. Por isso este modelo é designado por modelo de Markov observável.

No entanto o modelo de Markov observável é muito restritivo para ser aplicável em muitos problemas reais (Rabiner, 1989). Pode-se derivar um modelo mais sofisticado se considerarmos que uma determinada observação tem uma probabilidade de ocorrência em função do estado em que se encontra o sistema. Neste caso cada estado não define uma ocorrência, mas um conjunto de ocorrências com diferentes probabilidades. Devido a ser possível que uma observação ocorra em diferentes estados, dada uma observação não é possível determinar com certeza o estado em que o sistema se encontra. Em virtude disto, este tipo de processos estocásticos são designados de modelos de Markov não observáveis (HMMs).

Em cada instante t o HMM muda de estado, gerando no estado q_j um vector de caracte-

rísticas x_t com densidade de probabilidade $p(x_t|q_j)$. A transição do estado q_j para o estado q_k tem uma probabilidade associada tomando o valor $p(q_k|q_j)$. Assim pode ser representado por um conjunto de estados ligados entre si por transições probabilísticas, com uma distribuição de probabilidades associada a cada estado, tal como está representado na Figura 2.4.

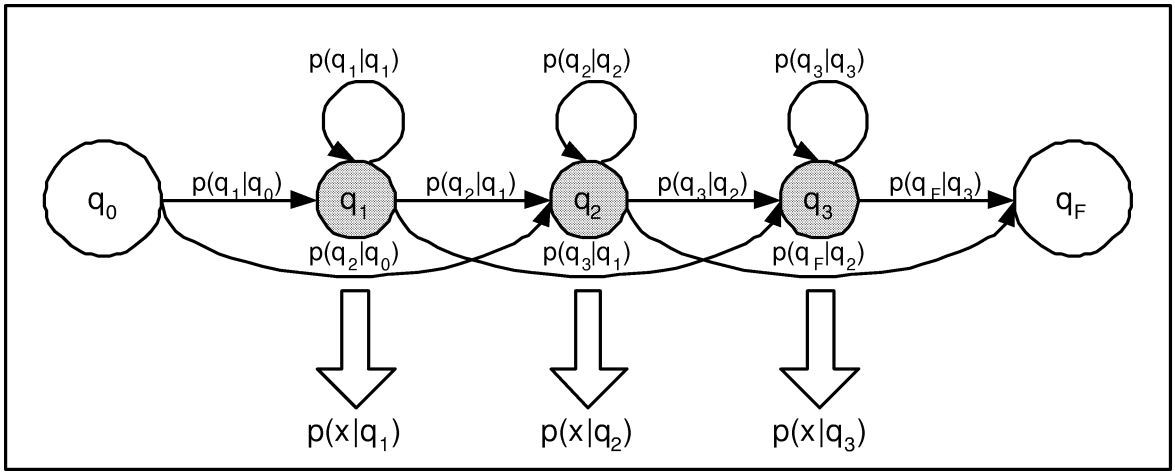


Figura 2.4: Esquema de um modelo de Markov com cinco estados numa topologia esquerda-direita. Fonte Meinedo (2000).

A comparação dos modelos com as características acústicas é efectuada com o cálculo de $p(X|W)$ para as diversas sequências de modelos de palavras W. A utilização de HMMs no cálculo de $p(X|W)$ implica que exista um mapeamento entre um conjunto de estados $Q = \{q_1, q_2, ..., q_{Nq}\}$ e a correspondente sequência de vectores acústicos $X = x_1, x_2, ..., x_{N_x}$. Considere-se $p(X|W)$ a soma da densidade conjunta da sequência de vectores X e da sequência de estados Q dada a sequência de palavras W, ao longo de todas as possíveis sequências de estados $\hat{Q}$. O seu valor é dado por:

$$p(X|W) = \sum_{\hat{Q}} p(\hat{Q}, X|W) \tag{2.5}$$

Substituindo $p(X|W)$, dado pela equação 2.5, na equação 2.3 obtemos:

$$\hat{W} = \arg\max_{W} P(W) \sum_{\hat{Q}} p(\hat{Q}, X|W) \tag{2.6}$$

Com vista a clarificar o processo de cálculo da adequação do modelo de palavra às características acústicas, $p(X|W)$, na Figura 2.5 está representado graficamente o método de cálculo. Por questões de simplicidade, considere-se apenas um HMM isolado e não a concatenação de vários. Cada coluna da malha contém a probabilidade de uma sequência parcial terminar num determinado instante de tempo T. Cada intervalo entre duas colunas corresponde à entrada das características acústicas de um segmento do sinal de fala. As setas representam as transições do modelo compondo todos os caminhos possíveis, desde o instante de tempo inicial até ao final. O cálculo processa-se de uma forma orientada por colunas, através de fórmulas de

recursão que envolvem os valores das colunas adjacentes, as probabilidades de transição dos modelos e os valores das distribuições de saída para a estado do HMM correspondente. Para a probabilidade α de ter sido observada uma determinada sequência parcial, o cálculo começa na coluna mais à esquerda, cujos valores são inicializados com os valores π de probabilidade inicial do HMM, acabando no lado oposto.

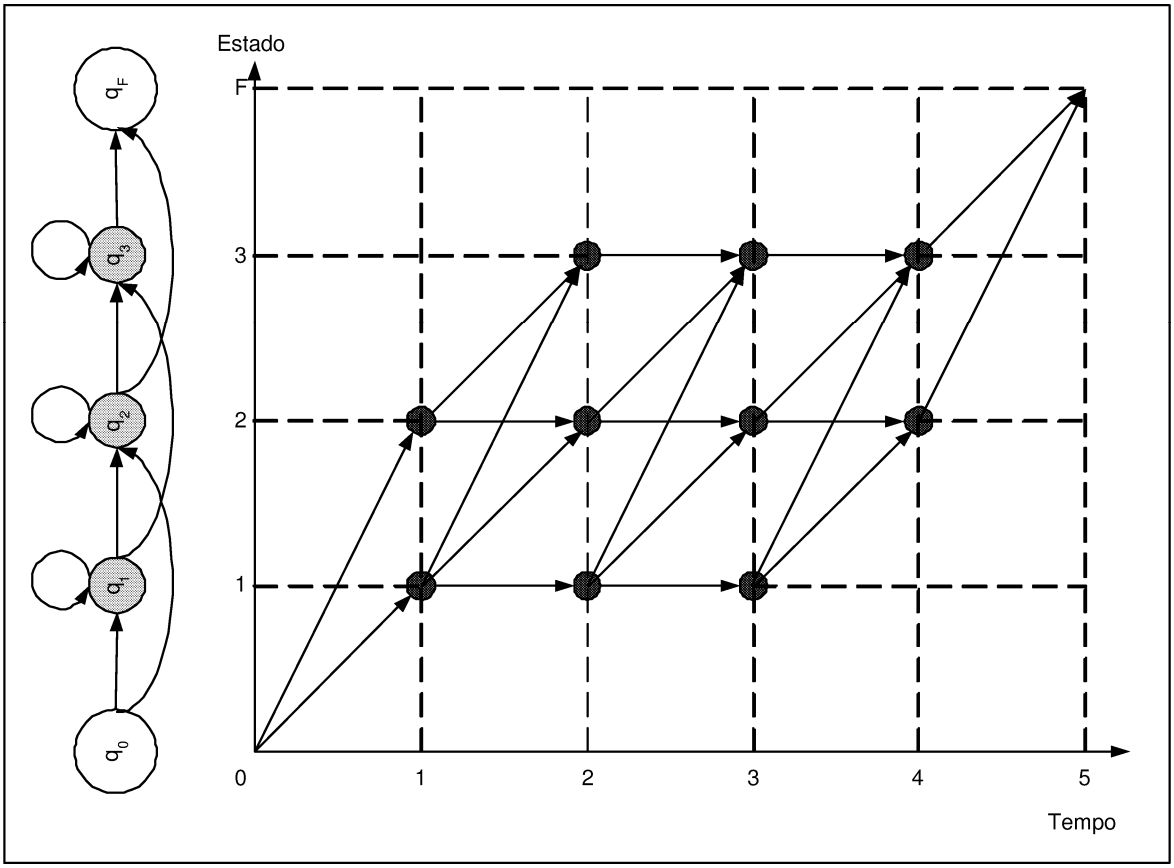


Figura 2.5: Malha de pesquisa de hipóteses de frase.

2.1.5 Descodificação

O sinal de fala quando produzido por um orador a falar naturalmente, sem qualquer preocupação com o ritmo do discurso, frequentemente não exibe uma clara indicação dos limites das palavras. A detecção desses limites faz parte do processo de formulação de hipóteses, sendo importante na medida em que vai condicionar a sequência de HMMs que melhor descreve o sinal. No processo de formulação de hipóteses, todos os modelos de palavras são comparados com a sequência de características acústicas medidas do sinal. Estas hipóteses podem ser representadas por uma sequência única de palavras, por uma colecção das x melhores sequências ou como um conjunto de hipóteses de palavras parcialmente sobrepostas. O modo como são representadas variam de sistema para sistema, mas de um modo geral os sistemas apresentam, se desejado, mais que uma alternativa para o sinal que captaram.

Devido ao domínio de pesquisa da sequência de palavras para o cálculo de $p(X|W)$ poder ser muito vasto, dependendo da quantidade de frases aceites pelo sistema, o cálculo de $\hat{W}$ é efectuado com recurso a algoritmos de optimização. Um dos algoritmos de procura que tem sido mais utilizado para o reconhecimento de fala contínua é o algoritmo de Viterbi (Rabiner e Juang, 1993). Este é um algoritmo de programação dinâmica recursivo concebido para encontrar o caminho mais curto num grafo, realizando uma busca eficiente. Quando aplicado ao reconhecimento de fala utilizando HMMs, a sua tarefa é encontrar o caminho mais provável de um grafo cujos pesos dos caminhos são dados probabilisticamente em função do tempo/estado, os caminhos ilustrados na Figura 2.5.

A descodificação de Viterbi é eficiente em problemas com um domínio de pesquisa pequeno, sendo exigente do ponto de vista computacional quando o domínio de pesquisa, que cresce em função do tamanho do vocabulário utilizado, é grande. Com vista a tornar este algoritmo aplicável a grandes vocabulários, várias técnicas de organização de dados e de poda de alternativas têm sido utilizadas de modo a reduzir o domínio de pesquisa. Um desses algoritmos é o *beam search*. Neste algoritmo é definido um intervalo de poda Δ relativo ao logaritmo da verosimilhança da hipótese mais provável para um instante t ($\log P_{\max}(t) - \Delta$). As hipóteses fora desse intervalo são eliminadas da busca.

Com o critério de Viterbi, a equação 2.6 é aproximada por:

$$\hat{W} = \arg\max_{W} P(W) \times \max_{Q \in Q_W} p(Q, X|W) \tag{2.7}$$

onde Q_W representa o conjunto de estados do HMM que correspondem à sequência de palavras W.

2.1.6 Dicionário de Pronúncia

É utilizado para definir a transcrição fonética das palavras, isto é, exprime o modo como as palavras são pronunciadas através da concatenação de certos fones. Cada fone corresponde a um HMM. Caso uma palavra possa ser pronunciada de modos diferentes, por exemplo devido a sotaques regionais, esta pode ser definida por dois ou mais conjuntos diferentes de fones, como o caso da palavra "hello" da Tabela 2.1.

Tabela 2.1: Algumas entradas do dicionário de pronúncia utilizado neste trabalho.

Palavra	**Transcrição Fonética**
Carl	K AA R L
hello	HH AH L OW
hello	HH EH L OW

2.1.7 Modelos de Linguagem

O modelo de linguagem define as relações entre todas as palavras pertencentes ao léxico do sistema. Em linguagem natural existem regras sintácticas e semânticas que condicionam as sequências de palavras. Ao modelar o conjunto de frases aceites numa determinada linguagem, o modelo linguístico ajuda a reduzir o domínio de pesquisa dos algoritmos de descodificação, uma vez que elimina sequências de palavras inválidas do domínio de pesquisa destes algoritmos. Também contribui para aumentar a precisão do sistema uma vez que restringe o número de palavras candidatas num determinado instante t, diminuindo assim o número possível de erros cometidos a cada instante.

Os modelos linguísticos podem ser construídos com recurso a uma gramática composta por regras rígidas ou por modelos estatísticos, sendo bastante usual a descrição do modelo com recurso a n-gramas. Quando são definidos por um conjunto de regras rígidas, uma sequência de palavras ou é aceite ou é rejeitada, não existindo um favorecimento das sequências mais plausíveis em detrimento das mais raras. Neste tipo de modelos existe o risco da gramática estar mal definida e serem rejeitadas frases que até fazem sentido. Porventura os modelos mais interessantes são os probabilísticos, pois permitem modelar as frases aceites sem serem muito restritivos.

O modelo n-grama assume que a probabilidade de ocorrência de uma palavra depende das n-1 palavras anteriores. Assim para uma dada sequência $W = \{w_1, ..., w_n\}$ de n palavras, a probabilidade dessa sequência é:

$$p(W) = p(w_1, ..., w_n) = p(w_1) \times \prod_{i=2}^{n} p(w_i | w_1, ..., w_{i-1}) \tag{2.8}$$

onde w_1 é escolhido apropriadamente para lidar com a condição inicial. Alguns dos modelos com maior sucesso das passadas duas décadas são os modelos bigrama e trigrama, ou seja, modelos nos quais n toma os valores 2 e 3 respectivamente. É de referir que para um vocabulário de tamanho V existem V^2 bigramas e V^3 trigramas possíveis, o que para 20 000 palavras dá 400 milhões de bigramas e 8 biliões de trigramas!

2.2 Compreensão de Linguagem Natural

A compreensão de linguagem natural (NLU) é o conjunto de métodos que permitem aos computadores "entenderem" textos em linguagem natural. Por "entender" um texto depreende-se reconhecer o contexto, fazer a análise léxica, morfológica, sintáctica e semântica, ter capacidade de identificar a informação pertinente e interpretar os sentidos das frases.

A arquitectura clássica de um sistema de NLU está esquematizada na figura 2.6. Na fase de pré-processamento, se existir, é comum efectuar a análise morfológica. Esta consiste na identificação da categoria morfológica (substantivo, verbo, pronome, etc.) e de como a palavra é formada, detectando entre outras outras coisas o número (singular ou plural) e o género

(feminino ou masculino). Em alguns sistemas, este tipo de análise pode não surgir como uma fase isolada da cadeia de processamento, sendo integrada na análise sintáctica. Nestes casos a morfologia das palavras está tradicionalmente registada no ficheiro de definição lexical, sendo as propriedades das palavras definidas na altura da determinação das entradas do léxico. Na análise sintáctica é construída uma árvore que representa a estrutura da frase. Por vezes o pré-processamento não existe, mas a extracção da estrutura sintáctica seguida da análise da informação semântica é um encadeamento muito comum nos sistemas actuais.

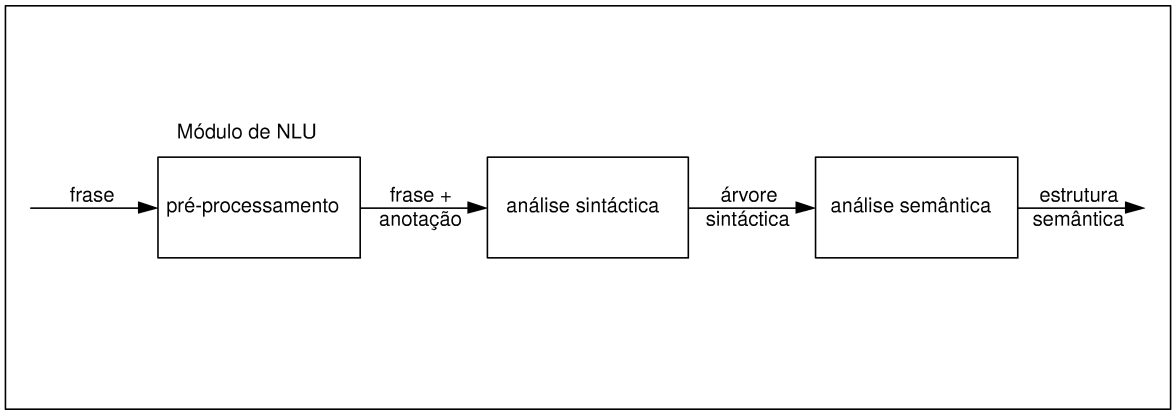


Figura 2.6: Esquema de um módulo de compreensão de linguagem natural.

A investigação em NLU tem sido efectuada mediante duas abordagens diferentes: baseada em conhecimento e baseada em dados. Com vista a que se compreenda melhor a escolha efectuada, primeiro vão ser discutidas de forma muito breve as características, vantagens e desvantagens de cada abordagem.

2.2.1 Abordagens Existentes

Abordagem baseada em conhecimento

Esta abordagem baseia-se na construção de gramáticas constituídas por um léxico, por regras sintácticas e por regras semânticas. Tanto as palavras do léxico como as regras são definidas por um conjunto de características e por vezes também por expressões em linguagem de programação. Um *parser* encarrega-se então de mapear os atributos das palavras que compõem a frase nas regras sintácticas da gramática, obtendo-se a estrutura da frase. Com o decorrer do tempo, foram desenvolvidos vários formalismos para definição de gramáticas, uns mais conformes com as teorias linguísticas, outros mais direccionados para alguns modelos de processamento ou aplicações específicas.

Sistemas desta natureza têm conseguido obter bons resultados para domínios de linguagem pequenos, ou seja para aplicações focadas num determinado assunto. Uma vantagem destes sistemas é que existe a garantia que a análise de qualquer frase, salvo algum erro no desenho do sistema, é uma análise correcta. Com esta abordagem podem-se efectuar análises bastante detalhadas, identificando todas as relações e entidades existentes numa frase (Cole

et al., 1995; Rosé e Lavie, 2001). No entanto estas gramáticas atingem frequentemente várias centenas ou milhares de regras que descrevem diferentes tipos de frases. Quando as gramáticas atingem determinados tamanhos, tornam-se difíceis de manter, de expandir e de reutilizar, originando problemas de escalabilidade da aplicação. Existe também o problema das frases extra-gramaticais. Sempre que uma frase não seja conforme a gramática um sistema deste género não consegue obter qualquer análise. Isto implica que toda a informação presente na frase é ignorada.

Abordagem baseada em dados

Os modelos de linguagem orientados aos dados assumem que a percepção e produção de linguagem humana funciona por representações de experiências linguísticas passadas, em vez de se basearem em regras gramaticais abstractas. Partindo de uma colecção de frases cujas análises são conhecidas (*corpus*), são construídos modelos probabilísticos de linguagem. Uma nova frase é analisada por analogia, utilizando fragmentos de análises previamente conhecidas. A análise mais provável é decidida pela frequência de ocorrência dos fragmentos (Bod e Scha, 1996). A grande vantagem desta abordagem é que o sistema consegue obter análises de frases para as quais não fora preparado previamente. Esta vantagem é muito grande uma vez que liberta o utilizador de conhecer as frases e o modelo de interacção da máquina (Bellegarda e Silverman, 2003).

Para obter estes modelos de linguagem são necessários extensos *corpora* de representações linguísticas ocorridas. Um dos problemas encontrados é a esparsidade dos dados, pois, apesar de existirem milhões de frases *on-line*, frases perfeitamente razoáveis não estão presentes (Lee, 2003). Outro factor menos positivo é que é necessário anotar previamente os *corpora* para treino. Também é importante verificar se o *corpus* de treino tem o mesmo contexto do sistema a construir, caso contrário o modelo de linguagem obtido pode não ser o mais adequado para a aplicação em vista. Por último, outra desvantagem destes sistemas é que não existe a garantia que a análise esteja 100% correcta, apesar da taxa de análises correctas ser elevada.

2.2.2 Anotação Morfossintáctica

A anotação morfossintáctica é o processo de atribuir uma etiqueta de categoria lexical a cada palavra de um *corpus*. Tem aplicação em vários problemas de NLP tais como: aquisição de informação para a análise sintáctica, extracção de informação para sistemas de classificação de documentos, sistemas de síntese de fala, etc. (Màrquez *et al.*, 2000).

O grande problema da anotação morfossintáctica é a resolução de ambiguidades. Este problema deriva do facto de certas palavras poderem desempenhar diferentes funções, conforme a frase em que estão inseridas. Por exemplo a palavra *book* é um verbo na frase "*book that flight*" e um substantivo na frase "*hand me that book*". Para resolver as ambiguidades, os anotadores necessitam de identificar correctamente o contexto em que cada palavra está a ser utilizada (Jurafsky e Martin, 2000). Neste caso por contexto tomam-se palavras e as

etiquetas possíveis adjacentes à palavra em questão.

Para se ter uma noção da dimensão deste problema, a tabela 2.2 mostra a quantidade de palavras ambíguas no Brown Corpus segundo DeRose (1988).

Tabela 2.2: Quantidade de palavras ambíguas no Brown Corpus segundo DeRose (1988). Fonte Jurafsky e Martin (2000, pg. 299).

Não ambíguas (1 etiqueta)	**35.340**	
Ambíguas (2 a 7 etiquetas)	**4.100**	
2 etiquetas	3.760	
3 etiquetas	264	
4 etiquetas	61	
5 etiquetas	12	
6 etiquetas	2	
7 etiquetas	1	"still"

Existem vários algoritmos de anotação morfossintáctica que dão origem a anotadores com diferentes filosofias. Em Cole *et al.* (1995) os anotadores são considerados com sendo baseados em regras ou baseados em probabilidades. Segundo Màrquez *et al.* (2000) estes devem ser divididos em: linguísticos, estatísticos e simbólicos[2]. Aqui vai ser considerada a divisão efectuada por Màrquez *et al.* (2000).

Em Allen (1994) é argumentado que, atribuindo sempre a etiqueta mais provável a cada palavra ignorando o contexto, frequentemente se obtém uma taxa de etiquetas correctas de cerca de 90%. Este valor deve ser considerado como o ponto de partida para o desempenho de algoritmos mais elaborados. Se um método não obtiver uma taxa de etiquetações correcta acima daquele valor conclui-se que não é um método eficaz.

Anotadores Morfossintácticos Linguísticos

Os algoritmos deste tipo de anotadores assentam numa arquitectura de duas fases. Na primeira fase é utilizado um dicionário para atribuir a cada palavra uma lista de potenciais etiquetas. Na fase seguinte regras de desambiguação escritas à mão são utilizadas para seleccionar, da lista de potenciais etiquetas, a etiqueta a atribuir a cada palavra. A título de exemplo, uma regra de desambiguação poderia ser: uma palavra é um substantivo em vez de um verbo se for precedida de um artigo (Jurafsky e Martin, 2000).

Estes sistemas têm por vantagem uma elevada precisão (Cole *et al.*, 1995). Por exemplo, implementações como o EngCG (English Constraint Grammar) podem atingir taxas de sucesso de cerca de 99% (Voutilainen, 1997). Pelo lado negativo é frequente que no resultado final continue a existir ambiguidade, ou seja, palavras às quais o sistema não conseguiu atribuir apenas uma etiqueta. Para resolver este problema, surgiram abordagens que aplicam métodos

[2]Na realidade, os anotadores aqui designados por simbólicos são considerados em Màrquez *et al.* (2000) como da família de *machine-learning*. Apesar da designação ser diferente, em português é mais representativo chamar anotador de aprendizagem simbólica ou anotador simbólico.

de desambiguação estatística às palavras que continuam ambíguas, de modo a ser obtido um resultado completamente desambiguado (Cole *et al.*, 1995). Outro problema é o número de regras necessárias nos modelos linguísticos. Este pode variar entre algumas centenas a vários milhares, implicando normalmente alguns anos de desenvolvimento (Màrquez *et al.*, 2000). O anotador EngCG tinha numa primeira versão 1100 regras, sendo depois expandidas para 3600 regras (Voutilainen, 1997).

Anotadores Morfossintácticos Estatísticos

Estes anotadores também são designados por anotadores estocásticos. O modelo estatístico utilizado para desambiguar as etiquetas das palavras é construído a partir de *corpora*. Este consiste no conjunto das frequências de co-ocorrência dos diferentes tipos de fenómenos linguísticos. As propriedades do modelo são normalmente adquiridas com n-gramas, ou seja, a probabilidade de uma dada sequência de comprimento n é estimada a partir do *corpus* de treino. Nos anotadores morfossintácticos são normalmente utilizados bigramas ou trigramas, sequências de duas ou três etiquetas respectivamente (Màrquez *et al.*, 2000). A etiqueta atribuída a cada palavra é aquela que maximiza a expressão:

$$Prob(etiqueta|palavra) \times Prob(etiqueta|n\ etiquetas\ anteriores)$$

Este método tem por vantagem poder captar informação importante que poderá passar despercebida à pessoa que desenvolve o sistema. Outra característica relevante é a garantia de que a solução obtida não é ambígua. Contudo o seu desempenho é inferior ao das mais recentes abordagens de anotação linguística. O desempenho das várias abordagens estatísticas é tipicamente de 96% de etiquetas correctas (Cole *et al.*, 1995). Um aspecto menos positivo neste método é que a informação linguística é capturada indirectamente, normalmente em dezenas de milhares de registos de probabilidades.

Anotadores Morfossintácticos Simbólicos

Apesar da abordagem estatística envolver algum tipo de aprendizagem, supervisionada ou não, dos parâmetros do modelo a partir de um *corpus* de treino, consideram-se anotadores morfossintácticos simbólicos apenas os sistemas que implementam modelos de informação mais sofisticados que os n-gramas (Màrquez *et al.*, 2000). Este tipo de sistemas apresentam diversas técnicas. Uma das mais bem sucedidas e também implementada numa ferramenta integrada no novo sistema do Carl é a anotação baseada em transformações (Seabra Lopes *et al.*, 2003b; Brill, 1995). Esta abordagem é uma instância da Transformation Based Learning (TBL) explicada no Anexo A.2.

A primeira implementação deste tipo de anotador foi o *Brill tagger* (Brill, 1995). Este sistema possui uma taxa de decisões correctas superior a 95% no caso de palavras conhecidas e acima de 80% no caso de palavras desconhecidas. Por palavras conhecidas entende-se palavras

contidas no conjunto de treino. Esta avaliação foi efectuada com os *corpora* Penn Treebank e British National.

2.2.3 *Parsing*

O *parsing* é o processo de obtenção da estrutura gramatical, definida por uma gramática, de uma sequência de símbolos finais (palavras). Esta tarefa pode ser vista como a determinação do modo como uma certa sequência de símbolos finais pode ser obtida através do símbolo inicial da gramática e utilizando as regras definidas nesta. Esta determinação pode ser efectuada essencialmente de dois modos: ascendente (em Inglês *bottom-up*), que consiste em começar da sequência de símbolos finais e tentar chegar ao símbolo inicial da gramática, e descendente (em Inglês *top-down*), que consiste em partir do símbolo inicial da gramática e obter a sequência de símbolos finais. Os *parsers* podem ser concebidos quer para linguagens de programação quer para linguagens naturais.

Definindo uma gramática onde estão expressas as possíveis estruturas de frases, torna-se possível obter a estrutura de qualquer frase aceite pela gramática. A obtenção da estrutura de uma frase, designada de análise sintáctica, é importante na medida que nessa estrutura ficam reflectidas as relações entre as palavras, permitindo por exemplo saber numa frase qual a acção associada a um determinado sujeito. Na generalidade dos sistemas e também no sistema discutido neste trabalho, é com base na estrutura sintáctica que é determinada a informação semântica. A semântica é a codificação do conteúdo informativo transmitido em cada frase. Como é sabido, a mesma informação pode ser transmitida sob a forma de diferentes frases. Quando frases diferentes têm o mesmo significado, a estrutura sintáctica pode ser diferente mas a informação semântica é igual.

2.2.4 *Parsing* Robusto

Em virtude da natureza ambígua das linguagens naturais, característica não presente nas linguagens de programação, não é invulgar os *parsers* para linguagem natural aceitarem algumas frases gramaticalmente incorrectas por forma a aumentarem o seu conjunto de frases interpretáveis. Os *parsers* com esta característica são designados de *parsers* robustos. Um *parser* concebido para processar frases em linguagem natural construídas livremente, deverá retornar uma análise correcta ou aproximadamente correcta[3] de cerca de 90% das frases que processa, de modo a ser utilizável em aplicações reais. Para isso deverá conseguir resolver os três problemas seguintes, os quais originam grandes dificuldades aos *parsers* que utilizam algoritmos convencionais (Cole *et al.*, 1995):

- segmentação - segmentar o texto em unidades com estrutura sintáctica definida na sua gramática. Este problema é mais dirigido a análise de texto escrito. O texto retornado

[3]Neste contexto aproximadamente correcta significa que a análise deverá abranger quase todas as palavras da frase.

pelos reconhecedores de fala, devido a reflectir uma única frase dita pelo orador, deverá constituir uma unidade com estrutura sintáctica definida;

- desambiguação - seleccionar a única análise semântica correcta das várias alternativas que podem surgir;
- subgeração - lidar com os casos que não são abrangidos totalmente pela gramática.

Os *parsers* convencionais não conseguem obter qualquer informação quando confrontados com problemas de subgeração e segmentação. Para a desambiguação têm a desvantagem de utilizarem apenas informação semântica detalhada dependente de contexto (Cole *et al.*, 1995).

O *parsing* robusto tem sido abordado de vários modos: simbolicamente, estatisticamente e por modelos neuronais. Enquanto as abordagens estatística e por modelos neuronais são inerentemente robustas, podendo os *parsers* ser treinados a partir de um *corpus* anotado, os *parsers* simbólicos são os mais aptos a efectuar uma análise mais profunda e detalhada baseada em princípios linguísticos. Para certas aplicações uma análise superficial pode ser suficiente, contudo se for requerida uma análise mais completa é necessário mais processamento, onde a robustez deve ser de novo levada em consideração (Rosé e Lavie, 2001).

Parsing Robusto por Modelos Neuronais

As redes conexas (redes neuronais) são populares por serem robustas e possuírem algoritmos de aprendizagem eficazes. Estas qualidades e a sua importância na modelação de processos cognitivos provocaram algum interesse na sua aplicação à NLU (Henderson e Lane, 1998).

Em vários domínios, as redes neuronais são uma alternativa viável aos métodos estatísticos. Não é o caso do *parsing* sintáctico, mas trabalhos recentes têm mostrado arquitecturas de redes neuronais aptas a resolver este problema: as Simple Synchrony Networks (SSN) (Henderson, 2000). Este tipo de rede possui dois tipos de memória (ver Figura 2.7). Uma que apenas representa informação acerca dos constituintes sintácticos (contexto dos constituintes) e outra que não tem qualquer informação sobre esses constituintes, apenas sobre as palavras (contexto "Gestalt"). Neste tipos de modelos, as palavras são fornecidas sucessivamente à rede neuronal, pela ordem que se encontram na frase. De cada vez que uma palavra é inserida, a rede neuronal efectua um ciclo pelos seus constituintes e combina o contexto obtido das palavras anteriores com a palavra que foi inserida. Uma descrição detalhada deste método pode ser consultada em Henderson e Lane (1998).

Têm a vantagem de aprenderem a sua própria representação interna do conhecimento, ficando assim menos dependentes da forma como o sistema foi concebido pelo programador. São tolerantes ao ruído e a erros nos dados, sendo particularmente aptas a obter modelos a partir de dados esparsos, um problema frequente dos *corpora*. No entanto, as redes neuronais não têm tido bons desempenhos no *parsing* sintáctico. Contudo, há quem defenda que num futuro próximo o seu desempenho poderá ser idêntico ao dos *parsers* concebidos com outras abordagens (Henderson, 2000).

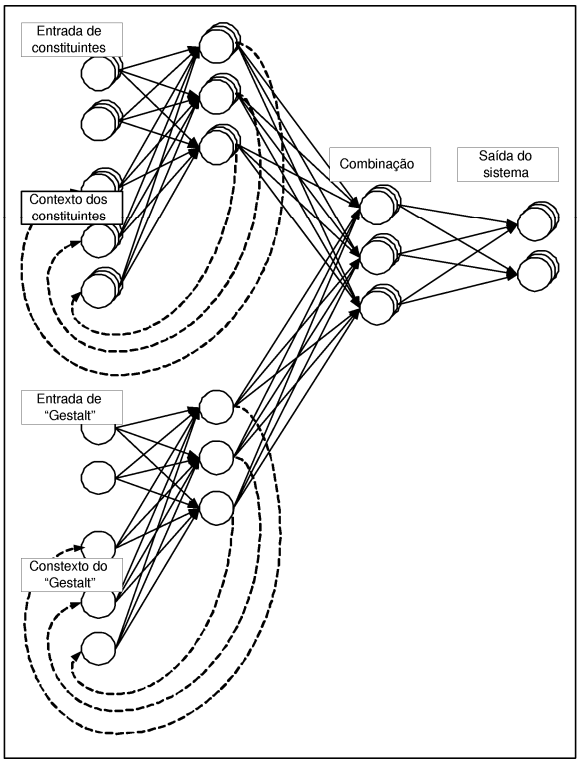


Figura 2.7: Representação de uma SSN. Fonte Henderson e Lane (1998, pg. 533).

Parsing Robusto Estatístico

Estes *parsers* funcionam com base em árvores adjuntas. Cada árvore representa uma parte elementar da estrutura sintáctica de uma frase (Bod, 2000). Na fase de treino, são estimadas as probabilidades de co-ocorrência das árvores, treinadas a partir de um *corpus*. A determinação da possível estrutura de uma frase é efectuada pela conjugação de árvores, sendo escolhida a estrutura cujas árvores têm maior probabilidade (Bod, 1998).

Estes sistemas têm por vantagem serem inerentemente robustos e por se conseguirem obter com eles bons resultados. Por exemplo em Chiang (2000) e em Bod (2000) são reportadas taxas de análises correctas superiores a 85%. No entanto estes *parsers* têm a tendência para favorecer as árvores sintácticas mais pequenas. Este comportamento deve-se ao facto da probabilidade da árvore final ser calculada como o produto das probabilidades das árvores elementares que a compõem (Bod, 2000). Outra desvantagem deste método é a velocidade do processamento não ser suficiente para a sua utilização em interfaces de linguagem natural falada.

Parsing Robusto Simbólico

Este tipo de *parsing* é baseado em gramáticas. Em algumas implementações os *parsers* podem ignorar regras gramaticais de modo a conseguirem obter uma maior cobertura. Assim estes *parsers* aceitam frases que não são conformes a sua gramática.

Abordagens mais antigas envolvem heurísticas embutidas na gramática de modo a suportarem possíveis erros na construção das frases. Neste caso, o trabalho de tornar o *parser* robusto e eficiente tem de ser refeito a cada nova gramática desenvolvida. Algumas abordagens recentes ao *parsing* robusto simbólico baseiam-se em técnicas de *parsing* superficial (em Inglês *shallow*) e parcial. Em vez de tentar obter a análise de toda uma frase não gramatical, esta abordagem tenta construir análises para segmentos máximos contíguos da frase. O resultado é uma forma prática de robustez, não envolvendo nenhum tipo de heurística associada à definição da gramática, mas provavelmente mais superficial que análises geradas por outros tipos de flexibilidade.

Este tipo de *parsers* conseguem obter análises bastante detalhadas, mesmo para frases não completamente conformes à gramática. Têm também a vantagem de algumas implementações, como o LCFlex, suportarem várias formas de flexibilidade como a inserção de estruturas gramaticais em falta (Rosé e Lavie, 1998). A eficácia e eficiência de um *parser* robusto simbólico estão directamente relacionadas com três factores:

- flexibilidade - que tipos de desvios à gramática devem ser suportados;
- pesquisa - como é que são procuradas todas as interpretações possíveis de uma frase e como é gerido o tamanho do domínio de procura;
- resolução de ambiguidades e escolha do resultado - quais os métodos utilizados para classificar potenciais soluções e em que informações estes se baseiam; como é gerida a ambiguidade provocada pela maior flexibilidade das regras.

Capítulo 3

Sistema Híbrido de Processamento de Linguagem Natural Falada

3.1 Abordagem Escolhida

O principal objectivo deste sistema é obter o máximo de informação possível de cada frase. Com vista a tirar proveito das vantagens próprias de cada abordagem o sistema é híbrido, implementando metodologias das abordagens baseadas em dados e em conhecimento. O sistema é também híbrido do ponto de vista da metodologia do processamento, uma vez que pode fazer a análise sintáctica com um *parser* robusto simbólico ou de um modo estatístico. A prioridade é a obtenção de uma análise tão profunda quanto a abordagem baseada em conhecimento permite, tirando proveito da abrangência da abordagem baseada em dados nos casos em que a primeira falha. Na Figura 3.1 está representado o fluxograma do sistema que vai ser explicado de seguida.

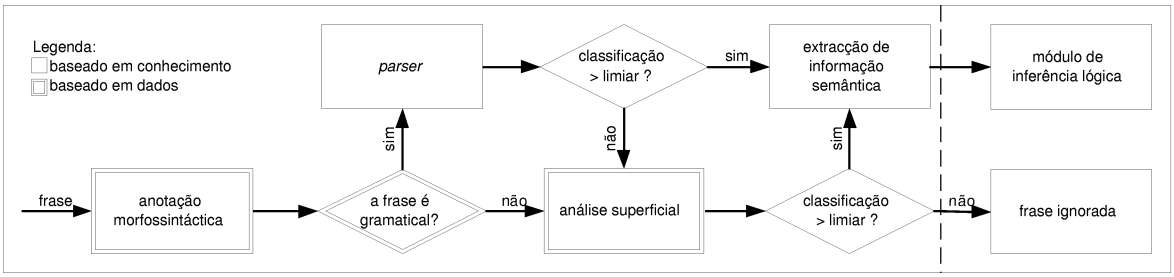


Figura 3.1: Fluxograma da solução adoptada.

No bloco de anotação morfossintáctica, o sistema começa por determinar a categoria lexical de cada palavra da frase. O objectivo é saber a função de cada palavra, quais os substantivos, os verbos, os advérbios, etc.. Depois, com base na frase e na informação morfossintáctica é decidido se a frase é gramatical ou não.

Caso a frase seja gramatical, a análise sintáctica é efectuada pelo *parser* de modo a se

determinar a sua estrutura sintáctica. Esta estrutura permite efectuar uma análise semântica mais detalhada, pois nela estão reflectidas as relações entre os diversos elementos que constituem a frase, permitindo consequentemente extrair mais informação.

Caso a frase não seja gramatical, ou se o resultado do *parser* for avaliado como não válido, a análise sintáctica é feita de um modo superficial. Uma vez que não é possível determinar a estrutura sintáctica da frase, o sistema tenta estabelecer analogias entre a frase em análise e frases que já conhece a respectiva análise. Esta análise superficial é efectuada por algoritmos de classificação automática de dados, baseados no algoritmo k-Nearest Neighbor (k-NN). Apesar de não se conseguir um grande detalhe na análise da frase, este método é bastante robusto porque compara a frase com um conjunto de frases conhecidas, atribuindo à frase a mesma análise da frase que determinou ser a mais semelhante. Assim existe a garantia que a análise obtida não difere muito da análise correcta.

Finalmente a análise sintáctica é encaminhada para o bloco de extracção da informação semântica, a não ser que seja proveniente do bloco de análise superficial e seja considerada não válida. Neste último caso, a frase é ignorada. O bloco de extracção de informação semântica tem em conta a profundidade da análise sintáctica: se for profunda, são extraídas todas as informações presentes na frase, se for superficial, apenas são extraídas as informações mais importantes. A representação semântica utilizada é a mesma do actual sistema do Carl, sendo informação representada por um conjunto de relações que descrevem os actores presentes na frase e seus atributos e as diversas relações entre os actores (Seabra Lopes *et al.*, 2003a). A análise final é encaminhada para o módulo de inferência lógica, um módulo não incluído no âmbito deste trabalho.

No decurso das várias etapas de processamento são utilizadas três técnicas:

- anotação morfossintáctica;
- *parsing* robusto;
- aprendizagem baseada em memórias.

3.2 Descrição Detalhada

A entrada do sistema é um sinal sonoro de fala. Um reconhecedor de fala encontra a sequência de palavras que melhor descreve esse sinal. Devido aos problemas inerentes ao reconhecimento de fala este não é 100% fiável, dependendo o seu desempenho da adequação dos modelos acústicos ao orador e do ruído ambiente onde o sistema está inserido. Como o Carl é um robô móvel concebido para interagir com qualquer pessoa em qualquer ambiente, não se pode fazer nenhuma consideração sobre a qualidade do reconhecimento num determinado instante.

Na implementação aqui descrita, tal como na maioria dos sistemas actuais, a análise final das frases é obtida após uma sucessão de etapas de processamento. É frequente cada etapa produzir resultados alternativos para a mesma informação de entrada, devido quer a

potenciais erros na percepção da frase quer a ambiguidades presentes na linguagem natural falada. Com vista a obter o melhor resultado global, cada etapa atribui uma classificação aos resultados que obteve, baseado em parâmetros de informação que serão explicados à medida que for explicada cada etapa. A ideia subjacente a esta opção é tentar não eliminar nenhuma alternativa minimamente viável, reservando a decisão para o momento em que existe mais informação disponível sobre a qualidade das soluções: o final de todo o processo de análise. Por isso o reconhecedor de fala envia para o módulo de NLU duas alternativas para a frase pronunciada. Não são enviadas mais alternativas pois testes efectuados demonstram que a frase pronunciada está na primeira alternativa em 47,6% dos casos, sendo 7,9% dos casos a segunda anternativa (Ferreira *et al.*, 2002). A percentagem de vezes que a frase pronunciada aparece na terceira alternativa e seguintes é muito pequena.

O reconhecedor de fala utilizado é baseado no Nuance 8.0 com trigramas extraídos de uma gramática da Carnegie Mellon University (CMU). De início eram utilizados bigramas treinados com recurso ao conjunto de frases aceites pela gramática utilizada no Carl. No entanto esse conjunto era muito pequeno, cerca de 12 000 frases, para ser considerado representativo da língua inglesa e esse facto reflectia-se na qualidade do reconhecimento. Assim é preferível utilizar um subconjunto da gramática treinada pela CMU, porque assim a proporção entre os diversos trigramas é mais representativa da língua inglesa. Foi construída uma ferramenta que extrai da gramática da CMU todos os trigramas, com as suas probabilidades associadas, constituídos pelas palavras desejadas. A gramática da CMU possui 64 000 palavras. Os textos utilizados para construir o modelo de linguagem foram retirados de várias fontes como notícias (Gouvêa, 2000), tendo sido utilizada como ferramenta de treino dos trigramas o CMU-Cambridge Toolkit (Clarkson e Rosenfeld, 1997).

O módulo de NLU recebe as duas sequências alternativas de palavras provenientes do reconhecedor de fala e analisa ambas. Este módulo é composto por duas partes: uma apta a efectuar uma análise detalhada para as frases aceites, ou quase aceites, pela especificação gramatical e outra capaz de extrair informação de frases com grandes desvios à gramática do sistema. A NLU pode ser decomposta por uma sequência de 3 processos que vão ser detalhados de seguida: pré-processamento, análise sintáctica e análise semântica. A arquitectura do sistema está esquematizada na Figura 3.2.

3.2.1 Pré-Processamento

A fase de pré-processamento tem dois objectivos: determinar a categoria sintáctica de cada palavra da frase e decidir se a frase é conforme ou possui apenas um pequeno desvio à gramática encaminhando-a para uma análise detalhada ou, caso contrário, para uma análise superficial.

A determinação automática da categoria sintáctica das palavras já tinha sido testada utilizando o Brill Tagger. Os resultados obtidos foram bons (Ferreira *et al.*, 2002; Soares, 2002) e é argumentado que este etiquetador morfosintáctico tem bom desempenho (ver Secção 3.3.4). Aproveitando o bom desempenho desta ferramenta e com o intuito de reduzir a complexidade

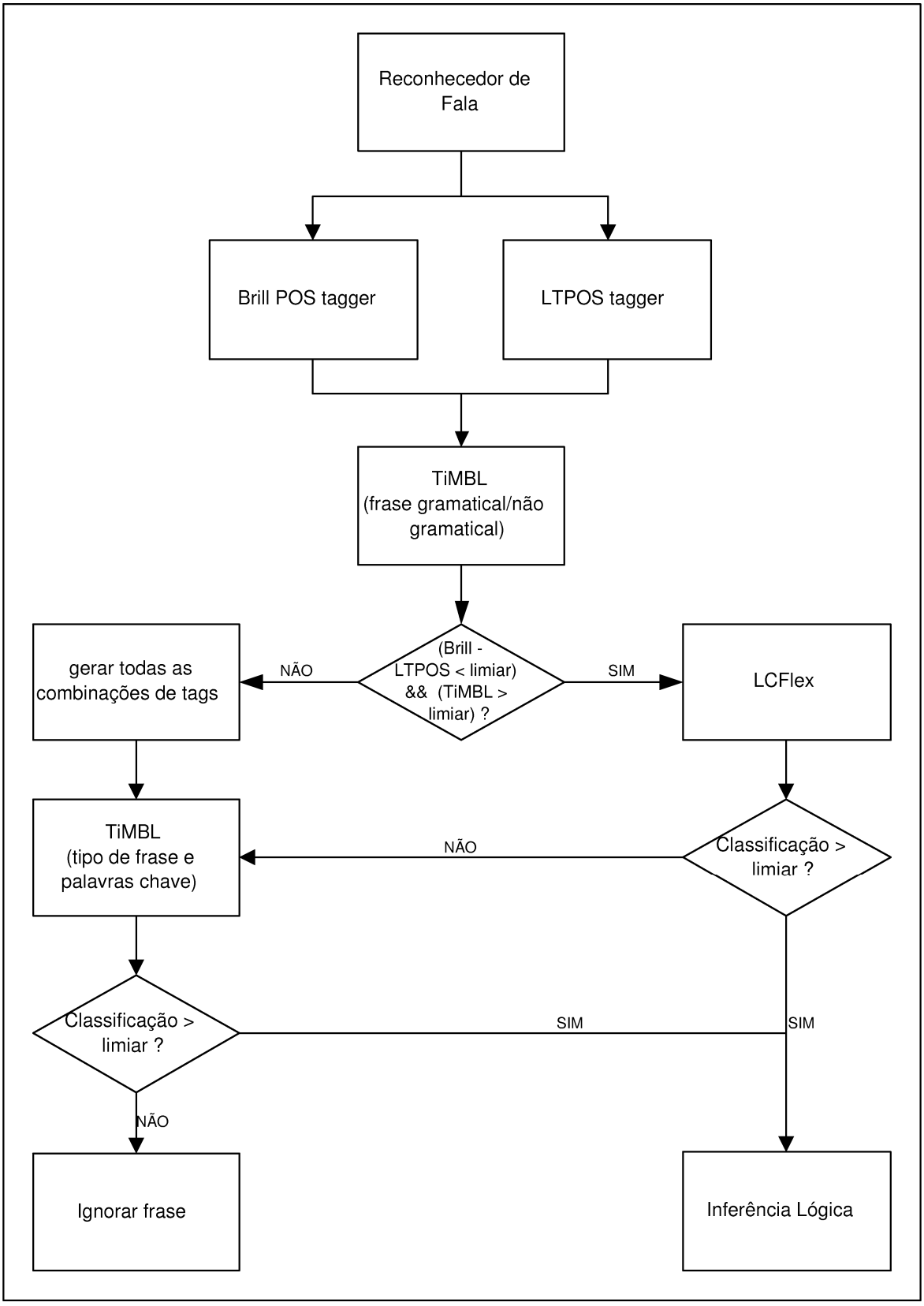


Figura 3.2: Arquitectura do sistema desenvolvido.

de desenvolvimento e posterior actualização do sistema, ao invés das categorias lexicais serem indicadas aquando da definição do léxico, estas são determinadas por este anotador durante a execução do programa. Esta opção tem a vantagem de ser apenas necessário definir o léxico para o reconhecedor de fala, pois o léxico do módulo de NLU passa a ser sempre o conjunto de etiquetas do Penn Treebank (ver tabela 3.2) utilizado pelo Brill Tagger. No entanto é sabido que o Brill Tagger comete erros de análise. Para minorar este problema é utilizado outro anotador morfossintáctico com o mesmo conjunto de etiquetas: o LTPOS. Como critério de confiança é considerado que as etiquetas estão correctas se ambos os anotadores estiverem de acordo em todos os casos. Caso contrário o sistema associa à frase os diferentes conjuntos de etiquetas morfossintácticas obtidas por ambos os anotadores.

O TiMBL baseia a sua análise em exemplos adquiridos. Os exemplos são fornecidos em vectores de pares característica/valor e são utilizados para construir um classificador de dados. Para decidir se a frase contém grandes desvios à gramática do sistema foi utilizado o algoritmo IGTREE do TiMBL. Este algoritmo foi escolhido porque a sua execução é bastante rápida (consultar secção 3.3.7) e é adequado quando o vector característica/valor utilizado não tem valores numéricos (quantitativos) como é o caso. O vector utilizado foi:

```
palavra1, etiqueta1, palavra2, etiqueta2, ..., palavra10, etiqueta10, classe
```

em que a classe toma dois valores: 1 se a frase for conforme a gramática e 0 caso contrário. Esta configuração já tinha sido testada e utilizada na versão anterior da interface de fala do Carl (Seabra Lopes *et al.*, 2003b).

Se o IGTREE considerar que a frase é gramatical e os anotadores estiverem de acordo com mais de 75% das etiquetas, a frase é considerada conforme ou com pequenos desvios à gramática e é passada para o *parser* robusto: o LCFlex. Caso contrário o sistema considera que a frase tem muito ruído e apenas tenta efectuar uma análise superficial recorrendo ao algoritmo IB1-IG do TiMBL. A frase é considerada gramatical se a sua classificação for superior a 45%. A classificação é dada por:

$$\frac{número\ de\ vizinhos\ com\ classe\ 1}{número\ total\ de\ vizinhos} \times 100\%$$

3.2.2 Análise Sintáctica

A análise sintáctica pode ser efectuada quer pelo LCFlex quer pelo TiMBL. O primeiro efectua uma análise detalhada, obtendo a estrutura sintáctica da frase enquanto o segundo apenas tenta identificar quais os sujeito e acção presentes na frase, obtendo assim uma análise superficial mas também mais imune a erros sintácticos da frase.

O LCFlex necessita de uma gramática para proceder à análise da frase. Para a análise sintáctica o tipo de informação que interessa é o papel ou classe que cada palavra desempenha na frase (verbo, substantivo, pronome, etc..) e não a palavra em si. Em virtude disto as entradas do léxico são as 36 etiquetas morfossintácticas do Penn Treebank. Como as entradas lexicais são poucas, foi possível construir uma gramática bastante abrangente com

poucas regras. O reduzido tamanho, além da vantagem de permitir uma fácil manutenção/ verificação da gramática também possibilita configurar o LCFLEX para poder inserir estruturas gramaticais em falta na frase, sem prejudicar de modo sensível a velocidade de obtenção da análise. Para o novo módulo de NLU o LCFLEX foi configurado para poder iniciar a análise em qualquer palavra da frase, ignorar até 3 palavras consecutivas, adicionar qualquer número de determinantes, até 1 sintagma nominal e até 1 sintagma verbal. Depois de analisar a frase, o LCFLEX retorna a estrutura sintáctica desta. As etiquetas morfossintácticas presentes na estrutura são substituídas pelas palavras que lhes correspondem. Esta informação é fornecida à componente de análise semântica se na análise tiverem sido incluídas pelo menos 75% das palavras da frase, caso contrário o resultado é considerado inválido e o sistema tenta efectuar uma análise sintáctica superficial.

A análise com o TiMBL é efectuada com a construção de várias hipóteses de frase. Em cada hipótese são sugeridos substantivos para o sintagma nominal, verbos para o sintagma verbal e, em alguns casos, mais atributos como advérbios e adjectivos. O conjunto de hipóteses geradas é composto por todas as combinações possíveis dos substantivos, verbos e atributos. Todas as hipóteses são avaliadas pelo TiMBL e é escolhida aquela que obtiver uma melhor classificação. Esta análise é definida como superficial, pois não se considera nenhuma estrutura para a frase, apenas a sua realização de superfície, as palavras. O algoritmo escolhido para efectuar a análise foi o IB1-IG do TiMBL. Este algoritmo é mais preciso que o IGTREE e considera os algarismos como quantidades e não como apenas mais um símbolo. Algumas características/valor que reflectem as propriedades das frases são numéricas, sendo o modelo de vector escolhido:

```
número de palavras, proporção de verbos, número máximo de verbos consecutivos,
   proporção de substantivos/pronomes,
   número máximo de substantivos/pronomes consecutivos,
   proporção de palavras Wh, número máximo de palavras Wh consecutivas,
   proporção de advérbios, proporção de adjectivos, proporção de artigos,
   um verbo (verbo1), etiqueta do verbo1, outro verbo (verbo2),
   etiqueta do verbo2, um substantivo/pronome (np1), etiqueta do np1,
   outro substantivo/pronome (np2), etiqueta do np2, uma palavra Wh,
   etiqueta da palavra Wh, um adjectivo, etiqueta do adjectivo,
   uma preposição, etiqueta da preposição, classe
```

em que a classe pode tomar os valores: i, y, w ou d se a frase for do tipo[1] "achieve", "ask", "ask_if" ou "tell", respectivamente, ou pode tomar o valor f se a frase for considerada não válida. Caso não exista apenas um único vector que descreva a frase, como por exemplo para os casos em que os anotadores não estão de acordo, são gerados todos os vectores possíveis e é considerado como mais apropriado aquele que tiver uma melhor classificação, dada por:

$$\frac{número\ de\ vizinhos\ da\ classe\ válida\ mais\ frequente}{número\ de\ vizinhos\ da\ classe\ mais\ frequente} \times 100\%$$

[1]Os tipos de frase estão descritos com mais detalhe na secção 3.2.3

Se esta classificação for superior a 45%, a informação é fornecida à componente de extracção de informação semântica.

3.2.3 Análise Semântica

O objectivo da componente de análise semântica é extrair a informação presente na frase. A representação do conteúdo semântico das frases é efectuada sob a forma de mensagens Human-Robot Communication Language (HRCL). O formato HRCL foi inspirado na linguagem Agent Communication Language (ACL) (Genesereth e Ketchpel, 1994). A sua concepção tem em linha de conta o modo como o robô "vê" o mundo e o género de capacidades que ele deverá possuir (Seabra Lopes e Teixeira, 2000). Na Tabela 3.1 estão os tipos de mensagem HRCL que estão implementados no robô e uma breve descrição para cada um deles.

Tabela 3.1: Tipos de mensagem HRCL presentemente implementados no robô e breve descrição de cada um deles. Fonte Seabra Lopes *et al.* (2003a).

Tipo de Mensagem	**Descrição**
achieve(S,R,C)	S pede a R para fazer a acção C no seu meio físico
ask(S,R,C)	S pergunta a R uma instância da preposição C
ask_if(S,R,C)	S pergunta a R se a preposição C é verdadeira
bye(S,R)	S despede-se de R
dye(S,R)	S (humano) pede a R (robô) para se desligar
register(S,R)	S anuncia a sua presença a R
tell(S,R,C)	S diz a R que a preposição C é verdadeira
thanks(S,R)	S agradece a R

Todas as mensagens são formadas por uma ou mais unidades com significado: as relações. As relações reflectem os actores presentes nas frases, as suas propriedades e relações entre si. Na secção 3.2.4 será apresentado um exemplo para melhor se perceber a natureza das relações extraídas.

Uma vez que a informação sintáctica que entra na componente de análise semântica tem dois formatos e níveis de detalhe distintos, dependendo se é fornecida pelo LCFlex ou pelo TiMBL, esta componente tem dois modos distintos de extracção de relações. Se a informação provém do LCFlex são extraídas todas as informações encontradas com o máximo detalhe possível. Se a informação provém do TiMBL apenas são extraídas as informações consideradas relevantes para o tipo de mensagem em questão, designadamente são identificados o emissor, o receptor e a principal relação contida na frase, ou seja um máximo de 3 relações.

Esta etapa é a última fase do processamento. No final deste passo são calculados os graus de confiança nas análises obtidas, recorrendo às classificações parciais calculadas nas diferentes etapas de todo o processo de análise. Para cada frase pronunciada pelo orador podem existir diversas análises. A análise considerada a correcta é aquela que obtiver melhor classificação final. A classificação final é a média das classificações obtidas nas diversas etapas de processamento, incluindo o grau de confiança do reconhecedor de fala na frase obtida. Se

a classificação final da análise escolhida for superior a um determinado limiar, esta é enviada para o módulo de inferência lógica do robô, caso contrário todas as análises obtidas são simplesmente ignoradas.

3.2.4 Exemplo de Análise

Nesta secção vão ser mostradas e explicadas as diferentes etapas porque passou uma frase incluída no teste, desde a saída do reconhecedor até à extracção das relações semânticas. Foi seleccionada uma frase reconhecida correctamente para se entender melhor a sua análise, pois quando as frases apresentam erros, nem sempre os resultados das diversas hipóteses são facilmente interpretáveis. A frase fornecida pelo reconhecedor de fala foi:

```
where is the coordinator of Carl
```

Os anotadores morfossintácticos concordaram em todas as etiquetas atribuídas às palavras, sendo por isso a classificação da anotação morfossintáctica 100. As etiquetas[2] atribuídas a cada palavra, pela mesma ordem que as palavras têm na frase:

```
WRB VBZ DT NN IN NNP
```

O algoritmo IGTREE do TiMBL decidiu que a frase tinha 100% de hipóteses de ser correcta, com base no vector:

```
where,WRB,is,VBZ,the,DT,coordinator,NN,of,IN,Carl,NNP,-,-,-,-,-,-,-,-
```

Em virtude desta decisão, a sequência das etiquetas morfossintácticas da frase foi enviada para o LCFlex. O resultado do LCFlex está ilustrado na Figura 3.3(a). Para se perceber melhor a estrutura sintáctica obtida, esta encontra-se representada graficamente na Figura 3.3(b).

Como a árvore construída pelo LCFlex inclui todas as etiquetas da frase, e por conseguinte todas as palavras, esta etapa teve uma classificação de 100%. A árvore obtida é enviada para a componente de extracção semântica. No final, a mensagem HRCL que representa a frase é:

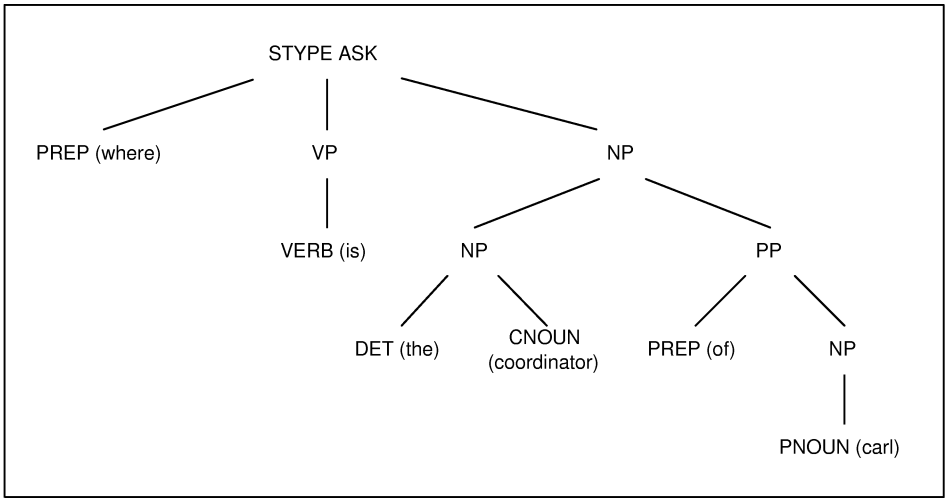
```
ask, where, _where, [be(none_, at, _coordinator, _where),
   type_(_coordinator, coordinator),
   of(none_, none_, _coordinator, _carl), name_(_carl, carl)]
```

O `ask` inicial indica o tipo HRCL da frase. O `where` é a palavra que indica que tipo de pergunta é a frase e o `_where` seguinte é a variável que representa essa palavra. A relação `be(none_,at,_coordinator,_where)` revela que se está a perguntar a localização de uma entidade a que o sistema se refere como `_coordinator`, informando a relação `type_(_coordinator,coordinator)` que a entidade `_coordinator` é um `coordinator`. A declaração `of(none_,none_,_coordinator,_carl)` afirma que a entidade `_coordinator` é

[2] Ver o seu significado na Tabela 3.2.

```
#S(EDGE :CAT <START> :P1 0 :P2 6 :NEEDED NIL :CHILDREN (263)
:FSTRUCT
((STYPE ASK) (PREP where) (VP ((VERB is)))
 (NP
  ((NP ((DET the) (CNOUN coordinator)))
   (PP ((PREP of) (NP ((PNOUN Carl))))))))
:RULE DF-1 :USED NIL :SKIP NIL :INSERT NIL :UNIF NIL
:TSTAMP NIL)
```

(a) Saída do sistema.



(b) Representação da árvore sintáctica.

Figura 3.3: Análise sintáctica do LCFlex para a frase "where is the coordinator of Carl".

da entidade `_carl`. Finalmente sabemos através de `name(_carl,carl)` que a entidade `_carl` tem o nome `carl`.

Este conjunto de relações é enviado para o módulo de inferência lógica, porque a classificação final da análise é 100%, uma vez que todas as classificações parciais foram 100%.

3.3 Ferramentas utilizadas

Nesta secção vão ser apresentadas e caracterizadas brevemente as ferramentas utilizadas para efectuar a anotação morfossintáctica, o *parsing* robusto e a aprendizagem baseada em memórias.

Para o reconhecimento de fala foram utilizados o Nuance 8.0 e os Sphinx 2 e 3, para a anotação morfossintáctica o Brill Tagger e o LTPOS, para o *parsing* robusto o LCFlex e para a análise superficial o TiMBL.

3.3.1 Nuance 8.0

Desenvolvido pela Nuance, este sistema de reconhecimento de fala pode ser obtido gratuitamente. Baseia-se numa arquitectura cliente/servidor, onde o cliente é responsável pela aquisição e pelo pré-processamento da entrada áudio e o servidor responsável pela correspondência entre a onda sonora e um texto, isto é, pelo reconhecimento de fala propriamente dito. O Nuance 8.0 funciona em tempo real, é independente do orador e suporta um vasto vocabulário e modelos linguísticos determinísticos ou baseados em n-gramas (Nuance, 2001a,b).

3.3.2 Sphinx 2

Foi desenvolvido pela CMU com financiamento da Defense Advance Research Projects Agency (DARPA) dos Estados Unidos da América. Este sistema utiliza HMMs semi-contínuos, funciona em tempo real, independente do orador e suporta um vasto vocabulário. Utiliza modelos linguísticos n-gram, inclusivamente podem ser obtidos gratuitamente modelos linguísticos trigramas com 64000 palavras para o Inglês. Este reconhecedor de fala foi disponibilizado, sob a forma de código fonte, no início do ano 2000.

3.3.3 Sphinx 3

Este sistema é a evolução do Sphinx 2. Utiliza HMMs contínuos, mantendo todas as características do seu antecessor. É um pouco mais lento que o Sphinx 2, mas no entanto tem um melhor desempenho. Está disponível ao público, sob a forma de código fonte, desde Dezembro de 2002.

3.3.4 Brill Tagger

Este anotador morfossintáctico começou a ser desenvolvido em 1992 por Eric Brill do Departamento de Ciência dos Computadores da Universidade de John Hopkins. O Brill Tagger introduziu uma nova abordagem na anotação morfossintáctica: a aprendizagem baseada em transformações, em Inglês Transformation Based Learning (TBL) (ver Anexo A.2).

Este anotador é composto por um classificador inicial e um módulo que aprende e aplica as transformações necessárias. Existem dois conjunto de transformações: um para palavras conhecidas, presentes no *corpus* de treino, outro para palavras desconhecidas. As transformações permitidas são definidas através de um conjunto de *templates* como por exemplo: muda da etiqueta x para a y se a etiqueta anterior é z. As regras são aprendidas aplicando as *templates* ao *corpus* de treino (Brill, 1995).

O Brill Tagger pode ser treinado com um *corpus* à escolha, sendo também possível mudar o classificador inicial e o conjunto de *templates*. No entanto optou-se por utilizar tal como é disponibilizado uma vez que foram obtidos bons resultados em vários testes. A configuração fornecida tem um classificador inicial estocástico e regras treinadas a partir dos *corpora* Brown e Wall Street Journal (WSJ) que classificam as palavras com as etiquetas presentes na Tabela 3.2.

Tabela 3.2: Conjunto de etiquetas morfossintácticas do Penn Treebank.

Etiq.	**Descrição**	**Exemplo**	**Etiq.**	**Descrição**	**Exemplo**
CC	conj. coordenativa	*and*, *but*	PP	pronome possessivo	*your*, *one's*
CD	número cardinal	*one*, *two*	RB	advérbio	*quickly*
DT	determinante	*a*, *the*	RBR	advérbio comparativo	*faster*
EX	"there" existencial	*there*	RBS	advérbio superlativo	*fastest*
FW	palavra estrangeira	*mea culpa*	RP	partícula	*up*, *off*
IN	prep./conj. subor.	*of*, *in*, *by*	SYM	símbolo (mat./cient.)	+, % , &
JJ	adjectivo	*yellow*	TO	"to"	*to*
JJR	adjectivo comparativo	*bigger*	UH	interjeição	*ah*, *oops*
JJS	adjectivo superlativo	*wildest*	VB	verbo infinitivo	*eat*
LS	marcador de lista	*1*, *2*, *One*	VBD	verbo pret. perfeito	*ate*
MD	verbo auxiliar	*can*	VBG	verbo gerúndio	*eating*
NN	subs. singular/massa	*llama*	VBN	verbo pret. imperf.	*eaten*
NNS	subs. plural	*llamas*	VBP	v. pres. não 3$^{\underline{a}}$ pess.	*eat*
NNP	subs. próprio singular	*Carl*	VBZ	v. pres. 3$^{\underline{a}}$ pess.	*eats*
NNPS	subs. próprio plural	*Carolinas*	WDT	determinante Wh	*which*, *that*
PDT	pronome indefinido	*all*, *both*	WP	pronome Wh	*what*, *who*
POS	terminação possessiva	*'s*	WPZ	pron. possessivo Wh	*whose*
PRP	pronome pessoal	*I*, *you*, *he*	WRB	advérbio Wh	*how*, *where*

O conjunto de *templates* utilizado foi mudar da etiqueta de palavras conhecidas de a para a b quando:

1. A palavra anterior (seguinte) tem a etiqueta z.

2. A palavra número dois antes (depois) tem a etiqueta z.

3. Uma das duas palavras anteriores (seguintes) tem a etiqueta z.

4. Uma das três palavras anteriores (seguintes) tem a etiqueta z.

5. A palavra anterior tem a etiqueta z e a palavra seguinte tem a etiqueta w.

6. A palavra anterior (seguinte) tem a etiqueta z e a anterior (seguinte) a essa tem a etiqueta w.

Onde a, b, z e w são variáveis de entre o conjunto de etiquetas possíveis. As 10 primeiras regras obtidas com este conjunto para o *corpus* do WSJ estão na tabela 3.3.

Tabela 3.3: Primeiras 10 transformações não lexicais obtidas com o *corpus* do WSJ.

	Muda etiqueta		
#	**De**	**Para**	**Condição**
1	NN	VB	a etiqueta anterior é TO
2	VBP	VB	uma das 3 etiquetas anteriores é MD
3	NN	VB	uma das 2 etiquetas anteriores é MD
4	VB	NN	uma das 2 etiquetas anteriores é DT
5	VBD	VBN	uma das 3 etiquetas anteriores é VBZ
6	VBN	VBD	a etiqueta anterior é PRP
7	VBN	VBD	a etiqueta anterior é NNP
8	VBD	VBN	a etiqueta anterior é VBD
9	VBP	VB	a etiqueta anterior é TO
10	POS	VBZ	a etiqueta anterior é PRP

Este anotador morfossintáctico também consegue etiquetar palavras para as quais não foi treinado, palavras desconhecidas. Inicialmente as palavras desconhecidas são etiquetadas de NN. Para a etiquetagem destas palavras, o conjunto de *templates* utilizado é mudar da etiqueta NN y se:

1. Apagando o prefixo (sufixo) x, $|x| \leq 4$, resulta numa palavra conhecida (x pode ser qualquer sequência de caracteres de tamanho compreendido entre 1 e 4).

2. Os primeiros (últimos) (1, 2, 3, 4) caracteres de uma palavra são x.

3. Acrescentando a sequência de caracteres x como prefixo (sufixo) resulta numa palavra ($|x| \leq 4$).

4. A palavra w aparece sempre imediatamente à esquerda (direita) da palavra.

5. O caracter z aparece na palavra.

Sendo y uma etiqueta, w uma palavra conhecida, x uma sequência de caracteres e z um caracter. As 10 primeiras regras obtidas com este conjunto para o *corpus* do WSJ estão na tabela 3.4.

Tabela 3.4: Primeiras 10 transformações obtidas para palavras desconhecidas, obtidas com o *corpus* do WSJ.

	Muda etiqueta		
#	**De**	**Para**	**Condição**
1	NN	NNS	o sufixo é "s"
2	NN	CD	tem o caracter "."
3	NN	JJ	tem o caracter "-"
4	NN	VBN	o sufixo é "ed"
5	NN	VBG	o sufixo é "ing"
6	??	RB	o sufixo é "ly"
7	??	JJ	adicionando o sufixo "ly" obtém-se uma palavra
8	NN	CD	a palavra "$" pode aparecer à esquerda
9	NN	JJ	o sufixo é "al"
10	NN	VB	a palavra "would" pode aparecer à esquerda

Este anotador possui uma taxa de decisões correctas superior a 95% no caso de palavras conhecidas e acima de 80% no caso de palavras desconhecidas (Brill, 1995).

3.3.5 LTPOS

Este anotador morfossintáctico, cujo nome é o acrónimo de Language Technology Part-Of-Speech, começou a ser desenvolvido por Andrei Mikheev do Language Technology Group da Universidade de Edinburgo. É um anotador morfossintáctico estocástico baseado em HMMs para etiquetação e desambiguação.

É enunciado que possui várias estratégias para determinar as etiquetas de palavras desconhecidas, que não pertençam ao *corpus*, mas não são divulgadas quais (Mikheev, 1994). É também afirmado que possui uma taxa de sucesso de 96% a 98% para palavras conhecidas e de 88% a 92% para palavras desconhecidas.

À semelhança do Brill Tagger, as palavras são classificadas com as etiquetas do Penn Treebank, apresentadas na Tabela 3.2.

3.3.6 LCFlex

É um *parser* simbólico flexível de canto-esquerdo (em Inglês *left-corner*) que utiliza uma tabela de *parsing* (Rosé e Lavie, 1998). Uma das suas principais vantagens é a de se poder controlar as suas características de flexibilidade através de parâmetros globais. Assim é possível moldar o seu comportamento às necessidades específicas de cada aplicação. Quanto maior for a flexibilidade permitida maior é a robustez alcançada mas também maior é a ambiguidade, sendo por isso necessário maior poder de cálculo, em termos quer de velocidade quer de memória disponível. Idealmente, o LCFlex deve ser ajustado de modo a atingir a maior robustez possível, mantendo uma velocidade de análise adequada à aplicação em causa. Uma

vez que este ajuste é dependente da aplicação, da gramática e da máquina que executa a aplicação, para atingir parâmetros óptimos são necessários alguns testes práticos e experiência do utilizador. Em futuras versões, os autores do LCFlex prevêem semi-automatizar este processo.

A arquitectura de base do LCFlex é o algoritmo de *parsing* de canto-esquerdo. Utiliza o fecho transitivo e reflexivo (em Inglês *reflexive and transitive closure*) da relação do canto-esquerdo (em Inglês *left-corner relation*) para filtragem descendente e previsão ascendente de modo a limitar a ambiguidade e com isso aumentar a eficiência (Noord, 1997). Procura analisar a maior subsequência possível de palavras de cada frase, podendo para isso ignorar palavras para efeitos de análise. Esta abordagem tem a vantagem de ser independente da gramática e orientada a frases completas, ao invés dos *parsings* superficial e parcial. Permite também efectuar inserção de palavras ou de estruturas gramaticais em falta.

A inserção generalizada de regras gramaticais foi demonstrada ser impraticável em aplicações com gramática e léxico vastos (Rosé, 1997; Rosé e Lavie, 2001). No entanto, o utilizador pode definir que combinações de palavras ou de estruturas gramaticais podem ser inseridas, permitindo só algumas inserções. A inserção de palavras pode não só ajudar a chegar a uma análise para uma frase não gramatical, como ajudar a identificar que tipo de erro causou a não gramaticalidade da frase.

Outro tipo de característica presente neste *parser* é a unificação flexível. Esta permite "ignorar" regras gramaticais caso tal seja necessário para a obtenção de uma análise da frase. A aplicação da unificação flexível a todas as regras gramaticais é ineficiente e por isso é possibilitado ao utilizador definir o conjunto de regras ao qual se aplica esta funcionalidade.

Finalmente, o LCFlex efectua o *parsing* em dois níveis. Na primeira passagem, feita com um nível de flexibilidade reduzido, são gerados vários fragmentos da análise da frase. Na segunda passagem, efectuada com maior flexibilidade, os fragmentos gerados pela primeira passagem são combinados com vista à obtenção da análise final da frase. Para resolução de ambiguidades e selecção da análise final utiliza um algoritmo padrão ao qual foi adicionado um método de poda de soluções semelhante ao introduzido em Lavie (1996). Este método de poda elimina análises que cubram apenas subconjuntos da frase que já estão integralmente contemplados por análises mais completas e gerais. A resolução de ambiguidades é cumprida por uma versão adaptada do método de desambiguação estatística discutido em Lavie (1996) e Carroll e Briscoe (1993).

3.3.7 TiMBL

O seu desenvolvimento foi efectuado maioritariamente por Ko van der Sloot tendo sido um esforço conjunto do programa Induction of Linguistic Knowledge da Universidade de Tilburg e do CNTS - Language Technology Group da Universidade de Antuérpia.

É um programa que implementa vários algoritmos de Memory Based Learning (MBL), que historicamente são descendentes do algoritmo de classificação de dados numéricos k-NN

(ver Anexo A.1). Todos os algoritmos implementados têm em comum o facto de possuírem explicitamente uma representação do conjunto de treino em memória. As diferenças mais significativas dos vários algoritmos implementados no TiMBL residem em:

- a definição de similaridade;
- o modo como as instâncias são armazenadas em memória;
- o modo como a pesquisa pela memória é efectuada.

Dos vários algoritmos implementados no TiMBL, dois foram utilizados no novo módulo. Em ambos os algoritmos foi utilizada a mesma métrica para a similaridade:

$$\Delta(X,Y) = \sum_{i} w_i.\delta(x_i,y_i) \tag{3.1}$$

Em que $\delta(x_i,y_i)$ é a distância da característica i das instâncias X e Y (equação 3.2) e w_i o peso atribuído a essa distância, calculado utilizando o Gain Ratio (equação 3.3).

$$\delta(x_i,y_i) = \begin{cases} abs\left(\frac{x_i-y_i}{max_i-min_i}\right) & \text{se numérico, senão} \\ 0 & \text{se } x_i = y_i \\ 1 & \text{se } x_i \neq y_i \end{cases} \tag{3.2}$$

$$w_i = \frac{H(C) - \sum_{v\in V_i} P(v) \times H(C|v)}{si(i)} \tag{3.3}$$

onde C é o conjunto de classes possíveis, V_i é o conjunto de valores para a característica i, $H(C) = -\sum_{c\in C} P(c)\log_2 P(c)$ a entropia das classes e $si(i)$ a informação de divisão da característica i dada por:

$$si(i) = -\sum_{v\in V_i} P(v)\log_2 P(v) \tag{3.4}$$

Com esta métrica, a relevância (pesos w_i) de cada característica é calculada automaticamente a partir do conjunto de treino. Esta operação é muito útil pois permite que a escolha das características a serem levadas em consideração seja menos rigorosa, um facto muito conveniente no caso do domínio de aplicação do classificador não permitir determinar claramente quais as características mais relevantes.

Os algoritmos utilizados no módulo em desenvolvimento foram o IGTREE e o IB1-IG. De seguida vai ser efectuada uma breve descrição de cada um.

IGTREE

Este algoritmo constrói uma árvore de decisão ordenada por ordem de relevância das características. Na raíz da árvore está a característica mais relevante, sendo a relevância atribuída pelo Gain Ratio (equação 3.3). Na altura da classificação, a árvore é percorrida da raíz até uma folha na qual é indicada a classe da instância a classificar.

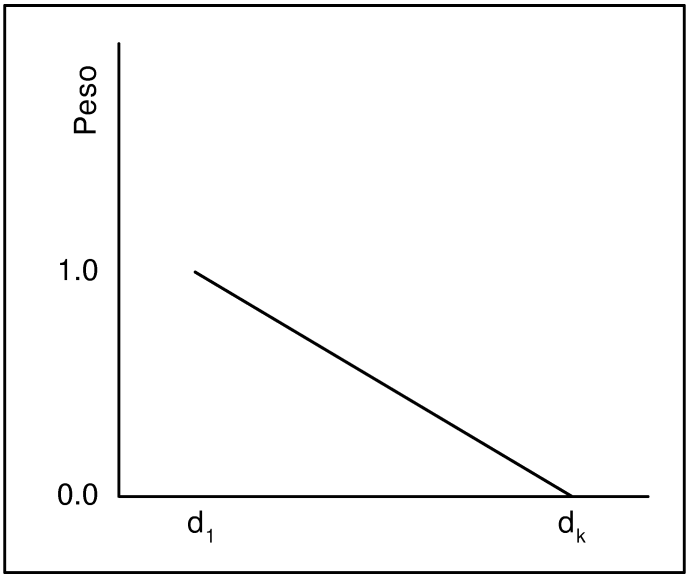


Figura 3.4: Peso das instâncias consoante a sua distância à instância a classificar

Não é indicado para características numéricas pelo facto de os números serem simplesmente tratados como caracteres e não como quantidades. A sua execução é bastante rápida uma vez que o IGTREE faz uma aproximação heurística à procura do vizinho mais próximo, através de uma árvore de decisão ordenada por relevância das características. Em virtude do modo como a procura da classe é efectuada e de não ser indicado para dados numéricos acaba por divergir bastante do k-NN clássico.

IB1-IG

Este algoritmo é igual ao k-NN mas com a métrica apresentada na equação 3.1. Na configuração utilizada o algoritmo considerava os 3 vizinhos mais próximos. Como existe a probabilidade de ocorrerem empates na medida de distância, o número de instâncias consideradas pode ser na realidade muito superior a 3.

Com o intuito de dar mais importância às instâncias mais próximas, de modo a que influenciem mais a decisão que as mais afastadas, cada instância é pesada de acordo com a equação (ver Figura 3.4):

$$g_j = \begin{cases} \frac{d_k - d_j}{d_k - d_1} & \text{se } d_k \neq d_1 \\ 1 & \text{se } d_k = d_1 \end{cases} \tag{3.5}$$

Onde d_j é a distância entre a instância a classificar e a j-ésima instância mais próxima, d_1 a distância à instância mais próxima e d_k à mais afastada.

Capítulo 4

Resultados

Foram realizados dois testes: um ao desempenho do reconhecimento de fala, outro ao desempenho do sistema de NLU.

O teste ao reconhecimento de fala justifica-se por este ser o principal calcanhar de Aquiles nos sistemas de SLP, principalmente quando estes são utilizados em ambientes ruidosos e/ou por pessoas cuja língua mãe não é a que está implementada no sistema. Se estes tivessem um bom desempenho, alguns problemas inerentes ao SLP não se colocariam. O teste consiste em comparar o desempenho de dois sistemas de reconhecimento de fala: um treinado para um determinado orador e ambiente e outro sem qualquer tipo de ajuste.

O teste ao desempenho do sistema de NLU tem o objectivo de determinar o grau de robustez deste quando confrontado com frases incorrectas do ponto de vista gramatical, uma vez que é comum as frases vindas do reconhecedor de fala apresentarem erros. Neste teste foram observadas as análises obtidas para várias frases retornadas pelo reconhecedor de fala. Apesar do sistema possuir diversos componentes, não foram efectuados testes a nenhum componente do sistema separadamente devido a apenas existir interesse no desempenho global do sistema e não no de cada componente. No entanto foram medidos os desempenhos parciais das componentes do sistema com o intuito de diagnosticar eventuais desequilíbrios na arquitectura, possibilitando assim identificar pontos fracos e efectuar correcções.

Os dados de entrada para o teste ao desempenho do sistema de NLU são todos os resultados provenientes do teste ao reconhecimento de fala, sem qualquer tipo de alteração ou manipulação. Como tal a conjugação destes dois testes fornece os resultados finais a todo o sistema desenvolvido nesta dissertação.

Nas secções seguintes vão ser descritos os procedimentos adoptados para cada teste, tal como os resultados obtidos.

4.1 Teste ao Reconhecimento de Fala

O objectivo deste teste é medir a diferença de desempenho entre um reconhecedor de fala com modelos acústicos independentes do orador e o mesmo reconhecedor de fala com modelos acústicos ajustados a um grupo de oradores. Este teste é importante em virtude do sistema ser principalmente utilizado por oradores cuja língua mãe não é a que é utilizada no Carl. Os modelos acústicos presentes no Carl são para o Inglês dos Estados Unidos da América e o Carl geralmente interage com portugueses. A possibilidade de criar um modelo acústico de raíz foi desde o início descartada devido a esta tarefa exigir uma grande quantidade de recursos, em especial dados para treino dos modelos, a que o grupo não tem acesso.

O problema dos utilizadores não nativos na língua do sistema é de difícil resolução uma vez que estes não são um grupo homogéneo (Compernolle, 1999). Para minorar este problema, uma das soluções possíveis é ajustar os modelos acústicos do sistema a um determinado grupo de utilizadores. O grupo de utilizadores escolhido foram os portugueses a falar Inglês.

Como o reconhecedor de fala em uso no Carl é baseado no Nuance 8.0 a ideia inicial era utilizar este sistema com o mesmo modelo de linguagem e dois modelos acústicos distintos. Um dos modelos acústicos seria o que estava a ser utilizado: um modelo independente do orador sem qualquer tipo de adaptação, o outro modelo seria baseado no original, mas em que seria efectuada uma ligeira adaptação a um ou mais oradores portugueses a falar inglês.

Pelo facto de se utilizar uma versão grátis, não houve acesso a informação suficiente para colocar o treino a funcionar. Uma vez que o principal objectivo deste teste é comparar o desempenho de um reconhecedor de fala com dois modelos acústicos diferentes, optou-se por efectuá-lo utilizando outro sistema de reconhecimento de fala: o Sphinx.

Primeiro tentou-se treinar o Sphinx 3 sem sucesso. Na altura a documentação para treinar os modelos acústicos ainda continha gralhas e a ajuda obtida não estava a ser muito eficaz. Para não despender mais tempo, optou-se por utilizar o Sphinx 2 cujas ferramentas e documentação para treinar os modelos acústicos estavam completas.

Assim o teste efectuado consistiu em comparar dois sistemas compostos pelo Sphinx 2 com o mesmo modelo de linguagem, mas com modelos acústicos diferentes: um com treino e outro sem treino. Com vista a se poder obter um paralelismo entre o Nuance e o Sphinx, procedeu-se à comparação do desempenho do Sphinx 2 sem treino com o Nuance 8.0, também sem treino. No Nuance foi utilizado o mesmo modelo de linguagem presente nos sistemas Sphinx 2.

Em conclusão, neste teste foram utilizados três sistemas de reconhecimento de fala, todos com o mesmo modelo de linguagem:

- Sphinx 2 com os modelos acústicos adaptados;
- Sphinx 2 com os modelos acústicos originais;
- Nuance com os modelos acústicos originais.

4.1.1 Metodologia do Teste

Primeiro foram gravados registos sonoros e respectivas transcrições textuais para treinar os modelos acústicos do Sphinx 2. Após o treino dos modelos, foram gravados mais registos sonoros de frases, diferentes das anteriores. Esta nova sequência de registos foi processada pelos três sistemas em análise. Os resultados foram gravados em ficheiros e analisados posteriormente. De seguida vai ser efectuada uma descrição mais detalhada de cada uma das etapas deste teste.

Fase de Treino dos Modelos Acústicos

Idealmente os registos sonoros utilizados para treinar os modelos acústicos deveriam conter amostras de vários oradores portugueses. No entanto, como já se tinha despendido bastante tempo nas sucessivas tentativas de treino e não existiam oradores imediatamente disponíveis, os registos sonoros foram todos produzidos pelo autor. A colecção dos registos de som é composta por 257 frases obtidas aleatoriamente da gramática do actual sistema do Carl. O som foi captado pelo microfone do Carl, dentro do laboratório do projecto. Uma vez que esta tarefa decorreu no horário normal de trabalho e não foi pedido às pessoas que tivessem qualquer tipo de cuidado, as amostras recolhidas contêm algum ruído de fundo proveniente das pessoas e das máquinas a trabalharem.

Cada frase pronunciada foi guardada num ficheiro de som separado. Para cada registo sonoro tem de existir uma transcrição textual da frase nele contida. Utilizando o registo sonoro e a transcrição textual, várias ferramentas do Sphinx 2 extraem diversas características das frases que irão servir para ajustar os modelos acústicos já existentes.

Fase de Teste

Para o teste foram seleccionadas 161 frases da gramática utilizada no actual sistema do Carl, com uma média de 5,7 palavras por frase. Este número é relativamente reduzido devido a restrições no modo de processamento dos resultados. O problema é que na análise só uma parte dos resultados é processada automaticamente, sendo uma parte considerável processada manualmente.

Cada frase foi gravada nas mesmas condições da gravação das frases para treino dos modelos acústicos. É de referir que o formato do som aceite pelo Nuance é diferente do aceite pelo Sphinx. O Sphinx aceita som adquirido a uma frequência de 16 KHz gravado no formato 16 bit linear. O Nuance aceita som amostrado a uma frequência de 8 KHz gravado em 8 bit Mu-Law. Para que os registos de som do Nuance e Sphinx fossem tão semelhantes quanto possível, o som foi gravado no formato aceite pelo Sphinx e posteriormente convertido para o formato aceite pelo Nuance. A conversão entre formatos foi efectuada utilizando um programa implementado em Matlab.

As frases foram seleccionadas aleatoriamente da gramática do Carl. No conjunto de frases

não estão representados alguns tipos de frases: o "dye(S,R)", o "bye(S,R)", o "register(S,R)" e o "thanks(S,R)". Acontece que para estes tipos existe apenas um conjunto muito pequeno de frases, menos de 10 frases para cada tipo num total de mais de 12000 frases presentes na gramática. Uma vez que estes são os tipos de frases com a análise mais simples, não foi considerado importante não aparecerem no teste global ao sistema. A Tabela 4.1 apresenta a distribuição por tipo de frase no conjunto de teste.

Tabela 4.1: Quantidade de frases de cada tipo presentes no conjunto de teste aos sistemas de reconhecimento de fala.

Tipo de Frase	**Número de Frases**
achieve	11
ask	50
ask_if	54
tell	46
total	161

Os três sistemas tentaram reconhecer todas as frases presentes nos 161 registos sonoros utilizados e guardaram as transcrições que obtiveram em ficheiros de texto.

4.1.2 Resultados

Para se avaliar o desempenho de cada sistema, compararam-se os resultados que cada um obteve com as transcrições das frases que realmente foram pronunciadas aquando a gravação dos registos sonoros.

A primeira medida de desempenho considerada foi a quantidade de frases bem reconhecidas (ver Tabela 4.2). Quer o Nuance quer o Sphinx 2 sem treino (Sphinx ST) reconheceram menos de 10 frases cada. O Sphinx 2 com treino (Sphinx CT), como se esperava, obteve um aumento de desempenho, na ordem de 550% quando comparado com o Sphinx ST, reconhecendo correctamente 26 frases.

Tabela 4.2: Número de frases bem reconhecidas por cada sistema de reconhecimento de fala.

Sistema	**Número de frases**
Nuance	7
Sphinx 2 sem treino	4
Sphinx 2 com treino	26

Contudo esta medida não será porventura a mais adequada para quantificar o desempenho de um reconhecedor de fala inserido num sistema de SLP. Por exemplo, pode acontecer que a mudança de um artigo definido para um indefinido não traga consequências ao nível da interpretação da ideia fundamental da frase, mas a frase já não é considerada bem reconhecida. Para um sistema de SLP interessa mais verificar qual o grau de corrupção da frase. O grau de

corrupção da frase reconhecida é calculado utilizando a frase que foi realmente pronunciada. O método consiste em chegar à frase reconhecida a partir da frase pronunciada. Para isso é permitido efectuar uma qualquer combinação de 3 operações, existindo uma penalização de cada vez que cada operação for utilizada. O grau de corrupção da frase é dado pela menor soma de penalizações que permitem transformar a frase pronunciada na frase reconhecida. As operações permitidas e respectivas penalizações são, segundo o *standard* da DARPA:

- inserir palavras. Cada palavra inserida tem uma penalização de 7 unidades;
- remover palavras. Cada palavra removida tem uma penalização de 7 unidades;
- substituir palavras. Cada palavra substituída tem uma penalização de 10 unidades.

A Figura 4.1 apresenta um exemplo para melhor se compreender este processo. Considere-se que foi pronunciada a frase "Peter is in the office" e é reconhecida a frase "Is Peter in the office". O objectivo agora é, partindo da frase reconhecida obter a frase pronunciada (Jurafsky e Martin, 2000, pg. 271). O primeiro caso presente na Figura 4.1 considera que se pode inserir a palavra "Is" antes da palavra "Peter" da frase reconhecida e remover o "is" a seguir à palavra "Peter". Esta solução envolve uma inserção e uma remoção, logo a penalização total é 7+7=14. O segundo caso considera que se pode substituir a palavra "Peter" pela palavra "Is" e de seguida substituir a palavra "is" por "Peter". Esta solução envolve duas substituições, logo a penalização total é 10+10=20. Como o primeiro caso conduz a uma menor soma de penalizações, seria aquela combinação de operações a escolhida, sendo neste caso o nível de corrupção da frase igual a 14.

Outra medida para a corrupção da frase, considerando o número de palavras inseridas, removidas e substituídas é o Word Error Rate (WER) (Jurafsky e Martin, 2000, pg. 271). O WER representa a taxa de erros presentes na frase: quanto mais baixo for o seu valor menos erros tem a frase. Sejam S, I e R os números de palavras substituídas, inseridas e removidas da frase, respectivamente. Seja T o número total de palavras da frase. O WER de cada frase é dado por:

$$WER = \frac{S + I + R}{T} \times 100\%$$

A Figura 4.2 contém um gráfico do número de frases em função do WER, para os 3 sistemas de reconhecimento de fala. Como se observa, o número de frases reconhecidas pelo Nuance com um WER inferior a 40% é baixo, cerca de 15 frases por intervalo de 20% de amplitude. O número de frases com erros vai aumentando até ao intervalo 80%-100%, altura em que atinge o máximo: aproximadamente 45 frases com um WER neste intervalo. O Sphinx ST apresenta um número crescente de frases com erros até um WER de 60%-80% e a partir desse ponto a quantidade de frases com erros diminui, sendo inferior ao registado com o Nuance a partir dos 80%. O Sphinx CT é dos 3 sistemas aquele que tem melhor desempenho. Por consulta do seu gráfico verifica-se que para aproximadamente 120 das 161 frases do teste o WER foi inferior a 60%. O número de frases com níveis de erros acima deste ponto diminui abruptamente, tendo uma subida no intervalo 80%-100%.

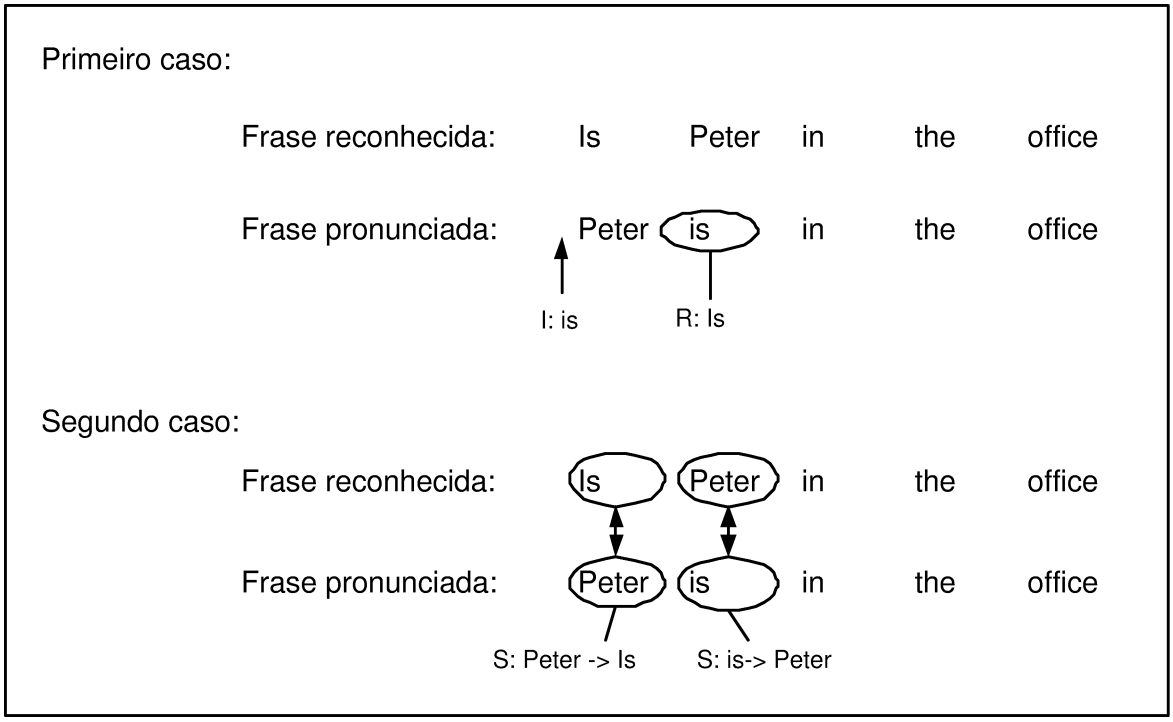


Figura 4.1: Exemplo de medida do grau de corrupção de uma frase.

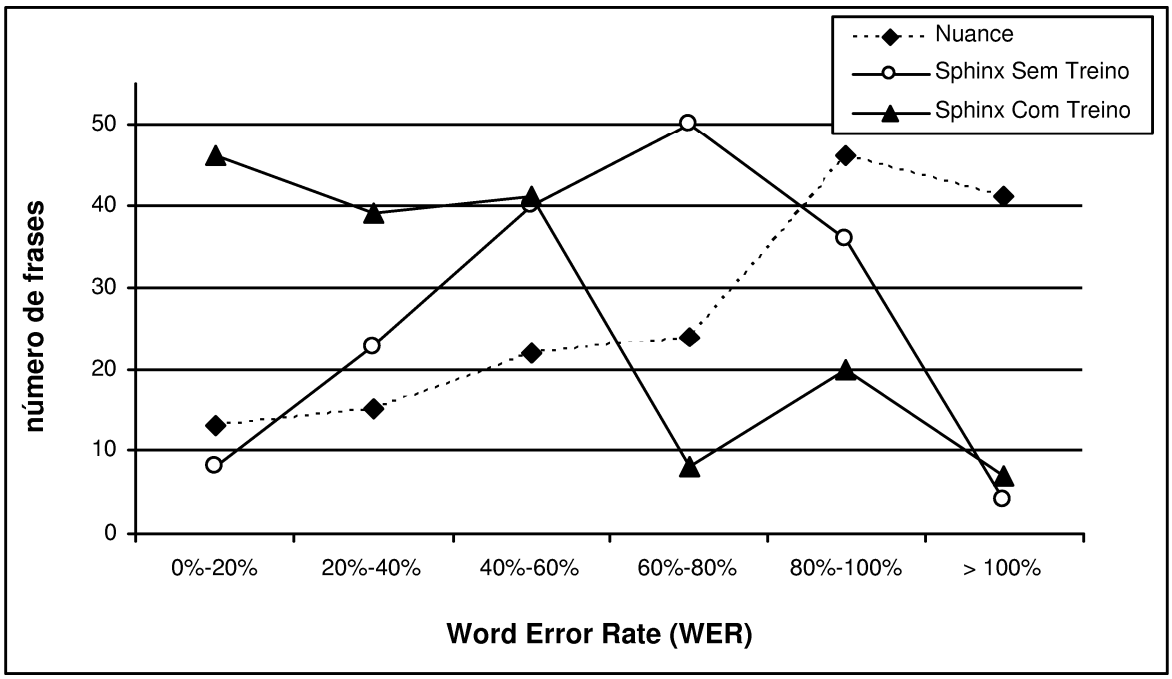


Figura 4.2: Gráfico do número de frases em função do WER obtidos pelos 3 sistemas de reconhecimento de fala.

Sejam $\overline{S}$, $\overline{I}$, $\overline{R}$ e $\overline{T}$ as médias de S, I, R e T, respectivamente. O WER médio de cada sistema é dado por:

$$\overline{WER} = \frac{\overline{S} + \overline{I} + \overline{R}}{\overline{T}} \times 100\%$$

O $\overline{WER}$ obtido para cada sistema em avaliação está apresentado na Tabela 4.3.

Tabela 4.3: Resultados médios por frase dos erros cometidos pelos sistemas de reconhecimento de fala.

(a) Nuance.

Medida	**Média (Desvio Padrão)**
palavras na frase	5,7 (1,5)
palavras inseridas	1,9 (1,9)
palavras removidas	0,2 (0,5)
palavras substituídas	2,4 (1,3)
$\overline{WER}$	79,1%

(b) Sphinx Sem Treino.

Medida	**Média (Desvio Padrão)**
palavras na frase	5,7 (1,5)
palavras inseridas	0,2 (0,6)
palavras removidas	1,6 (1,3)
palavras substituídas	1,7 (1,2)
$\overline{WER}$	61,6%

(c) Sphinx Com Treino.

Medida	**Média (Desvio Padrão)**
palavras na frase	5,7 (1,5)
palavras inseridas	0,5 (0,7)
palavras removidas	0,4 (0,7)
palavras substituídas	1,3 (1,0)
$\overline{WER}$	38,3%

Como se verifica na Tabela 4.3, o Nuance é o sistema que tem o WER maior. O resultado não deixa de ser curioso, uma vez que reconheceu correctamente mais frases que o Sphinx ST. O $\overline{WER}$ do Nuance e do Sphinx ST são claramente acima dos 50% (Tabelas 4.3(a) e 4.3(b)): isto significa que a maioria das palavras reconhecidas estão erradas. Por observação da Tabela 4.3(c) verifica-se que o mesmo não acontece com o Sphinx CT, uma vez que o seu WER médio é de 38,3%. O Sphinx CT além de ter reconhecido correctamente mais frases, comete nitidamente menos erros que qualquer um dos outros dois sistemas.

A inserção de palavras deve-se a palavras que não foram reconhecidas e que estão em falta na frase que resulta do processo de reconhecimento. A remoção de palavras diz respeito a palavras que não existiam na frase original, mas que o reconhecedor de fala incluiu no resultado final. Estes erros devem-se, de um modo geral, a dificuldades de alinhamento de palavras, ou seja, devido ao discurso ser em fala contínua e ser frequente não existir pausas claras entre duas palavras consecutivas, os reconhecedores de fala têm dificuldades em determinar quais os

segmentos de sinal que correspondem a uma palavra e quais os que correspondem às palavras adjacentes. Como se verifica também pela observação das Tabelas 4.3(a) e 4.3(b), a média de palavras inseridas pelo Nuance é próxima da média de palavras removidas pelo Sphinx ST, e vice-versa. Repare-se também que com a melhor adequação dos modelos acústicos ao orador, a grande melhoria do sistema Sphinx foi na média de palavras removidas. Passou de uma média de 1,6 palavras do Sphinx ST para uma média de 0,4 no Sphinx CT, uma redução de 75%.

4.2 Teste ao Sistema de Compreensão

Tal como referido na secção 3.2.3, o sistema desenvolvido tem por objectivo extrair uma série de relações semânticas de cada frase. Essas relações semânticas reflectem o conteúdo da frase e neste teste vamos verificar a qualidade dessas relações e portanto a quantidade de informação correcta obtida das frases.

A qualidade das relações semânticas é verificada por uma pessoa. Idealmente esta tarefa deveria ser realizada por várias pessoas, mas devido a poder ser muito prolongada no tempo não existia ninguém disponível. Com o objectivo de minimizar possíveis erros resultantes de existir apenas um revisor, optou-se por utilizar uma análise de referência que serve de verdade, a análise ideal. Uma vez que o novo sistema pretende seguir a especificação semântica do sistema implementado com base no ALE, considerou-se que a análise ideal é dada pelo ALE para as transcrições das frases que realmente foram pronunciadas, não das frases reconhecidas.

Foram comparadas as análises das frases reconhecidas obtidas pelo novo sistema, com as análises que seriam obtidas pelo sistema corrente do Carl (baseado no ALE). Assim pode-se comparar a diferença de desempenho entre o novo sistema e o sistema em uso, verificando desta forma se o novo sistema é mais vantajoso para a análise de linguagem natural falada. A comparação dos resultados obtidos com a análise ideal, permite ter uma ideia da quantidade de informação que é possível extrair das várias frases em análise, tendo elas sido bem reconhecidas ou não.

Para este teste a ideia era utilizar todas as frases provenientes dos 3 sistemas de reconhecimento de fala. No entanto não foram consideradas 33 frases reconhecidas que, devido a uma falha na especificação do léxico do ALE, este não as conseguia analisar em virtude de não conhecer duas palavras que era suposto conhecer, não conseguindo por isso analisar qualquer frase que incluísse essas palavras. Como a análise ideal é efectuada pelo ALE e assim sendo não existia termo de comparação, estas frases foram ignoradas, sendo a nova distribuição por tipo de frase apresentada na Tabela 4.4. É de realçar que o novo sistema conseguiu analisar todas as frases. Como foram submetidas 128 (161-33) frases provenientes de cada um dos 3 sistemas de reconhecimento de fala, no total temos 128×3=384 frases para analisar, o que é um número mais significativo. É óbvio que apenas partimos de 128 frases distintas, mas devido aos frequentes erros no reconhecimento de fala é frequente termos 3 resultados distintos para cada frase pronuniada.

Tabela 4.4: Quantidade de frases de cada tipo presentes no conjunto de teste aos sistemas de compreensão de linguagem natural.

Tipo de Frase	**Número de Frases**
achieve	7
ask	49
ask_if	34
tell	38
total	128

4.2.1 Metodologia do Teste

A metodologia do teste inicialmente pensada, e que seria desejável, foi modificada. As modificações foram motivadas por dois problemas encontrados com os sistemas de reconhecimento de fala.

O primeiro problema surgiu devido ao Sphinx 2 não determinar o grau de confiança na frase. No sistema desenvolvido, o nível de confiança do reconhecedor de fala na frase reconhecida é tido em conta na decisão final sobre a validade do resultado obtido. Uma vez que no Sphinx 2 não existe esta medida que indicia a fiabilidade do texto retornado pelo reconhecedor de fala, optou-se, por questões de equidade, por também não a utilizar no Nuance. A solução encontrada foi tomar a decisão final apenas com base nas classificações parciais obtidas nas diversas etapas de processamento do módulo de NLU. Obviamente que esta solução prejudica as decisões tomadas uma vez que o módulo de NLU passa a tratar todas as sequências de palavras de modo igual, ignorando a adequação dos sons que as geraram aos modelos linguísticos e acústicos do reconhecedor de fala.

A outra contrariedade foi o facto do Nuance não fornecer alternativas se ler o som de um ficheiro e do Sphinx 2 não ter qualquer informação sobre a confiança nas frases, ficando sem se saber a diferença de valor entre as alternativas. Em virtude disto passou a ser fornecida apenas uma alternativa de frase ao sistema de NLU, quando estava previsto que recebesse até duas alternativas para cada frase pronunciada. Com esta alteração o módulo de NLU passou a ter menos informação disponível para tomar decisões acerca da análise da frase.

Apesar destas modificações na metodologia do teste a essência deste permanece inalterada. Como o objectivo principal é comparar o desempenho do sistema em estudo com o desempenho do sistema actualmente implementado no Carl, apenas se tem de garantir que ambos os sistemas são testados nas mesmas circunstâncias, sendo estas adequadas para os sistemas em causa.

4.2.2 Resultados

A análise aos resultados obtidos visou quantificar três medidas de desempenho diferentes: número total de frases que o sistema conseguiu analisar, número de decisões correctas acerca

dos tipos de frases e quantidade de informação semântica extraída correctamente.

A primeira medida considerada para avaliar o desempenho de ambos os módulos de NLU foi a quantidade de frases para as quais conseguem obter uma análise, completamente correcta ou não. Esta medida não indica se as análises são correctas, mas indicia a versatilidade dos sistemas. Não se deve esquecer que um objectivo importante a atingir é fazer com que o Carl consiga reagir ao máximo número de frases possível.

As decisões sobre os tipos de frase são importantes pois os tipos de frase reflectem a intenção do utilizador: se está a querer saber alguma informação (frases do tipo "ask" e "ask_if"), se está a dar alguma ordem (frases do tipo "achieve") ou se está a dar alguma informação útil ao robô (tipo de frase "tell"). O tipo de frase tem influência no modo como a informação semântica obtida posteriormente vai ser agrupada na mensagem HRCL.

Por fim o teste à informação semântica obtida é importante pois é nesta etapa que se procura obter a informação relevante contida nas frases. É por isso necessário garantir que a informação obtida está correcta.

De seguida vão ser apresentados os resultados alcançados pelo sistema aqui proposto e pelo sistema que está actualmente em utilização no Carl nos três testes.

Teste 1: quantidade de frases analisadas

Na Tabela 4.5, apresentam-se os resultados obtidos para as frases provenientes de cada um dos sistemas de reconhecimento de fala. Nesta tabela e seguintes, o sistema em estudo é designado por "Novo sistema".

Tabela 4.5: Decisões tomadas para o tipo de frase, independentemente da validade do resultado.

(a) Frases vindas do Nuance.

Frases	**Novo sistema**	**ALE**
"achieve"	6	5
"ask"	0	0
"ask_if"	11	2
"tell"	21	0
todas	38	7

(b) Frases vindas do Sphinx Sem Treino.

Frases	**Novo sistema**	**ALE**
"achieve"	9	0
"ask"	0	0
"ask_if"	25	1
"tell"	28	3
todas	62	4

(c) Frases vindas do Sphinx Com Treino.

Frases	**Novo sistema**	**ALE**
"achieve"	0	0
"ask"	23	13
"ask_if"	38	6
"tell"	36	7
todas	97	26

Como se observa nas Tabelas 4.2 e 4.5 o ALE processa, para cada sistema, tantas frases como as que foram bem reconhecidas. Uma observação cuidada aos ficheiros de resultados per-

mite verificar que as frases processadas foram exactamente as frases bem reconhecidas. Num total de 128 frases o ALE conseguiu processar no melhor dos casos 26 frases (Tabela 4.5(c)), processando 4 em 128 se as frases forem obtidas através do Sphinx ST (Tabela 4.5(b)). O efeito destes resultados no comportamento do robô é o constante ignorar das frases que lhe são dirigidas, uma vez que não consegue obter qualquer análise para estas. Como é compreensível, nenhum utilizador fica a falar muito tempo para uma máquina que não tem qualquer reacção.

O novo sistema consegue sempre processar mais frases que o sistema implementado no Carl baseado no ALE. Repare-se que este sistema parece mais sensível ao WER que ao facto de a frase ser bem ou mal reconhecida. Na secção 4.1.2 verificou-se que o Nuance reconheceu correctamente mais frases que o Sphinx ST, mas este último tinha um WER menor. Nas Tabelas 4.5(a) e 4.5(b) verifica-se que o sistema em estudo conseguiu analisar 62 frases vindas do Sphinx ST e 38 vindas do Nuance. Tal como era esperado, qualquer dos sistemas teve o seu melhor desempenho com as frases reconhecidas pelo Sphinx CT, no caso do novo sistema 97 frases (Tabela 4.5(c)). Este resultado realça a importância do reconhecedor de fala ter modelos acústicos adequados para os utilizadores alvo.

Teste 2: determinação do tipo de frase

Uma das etapas da análise de uma frase é a determinação do seu tipo, isto é, tem de ser decidido se a frase é representada por uma mensagem HRCL do tipo "tell", "ask", etc.. Na Tabela 4.6 estão as decisões tomadas pelo sistema em estudo confrontadas com a resposta correcta dada pela análise ideal. A Tabela 4.6(a) apresenta a quantidade de frases provenientes dos três sistemas de reconhecimento de fala, para as quais o sistema conseguiu determinar o tipo de frase e a Tabela 4.6(b) mostra a quantidade de frases analisadas pelo ALE. Na Tabela 4.6(c) estão os resultados obtidos pelo novo sistema de NLU para as frases que o ALE também consegue analisar. Da observação desta tabela verifica-se que o novo sistema não cometeu nenhum erro na análise para este conjunto de frases, pois todas as decisões que tomou foram correctas.

Duas medidas utilizadas para avaliar o desempenho de sistemas de extracção de informação são a precisão e a cobertura (*recall*). A precisão reflecte a quantidade de informação correcta que o sistema consegue obter e a cobertura reflecte a quantidade de informação relevante extraída. Para caracterização destes sistemas também se utiliza uma terceira medida, designada por *F-measure*, obtida pela combinação da precisão com a cobertura. Seguidamente vão ser analisados os resultados apresentados nas Tabelas 4.6(a) e 4.6(b) do ponto de vista da precisão, cobertura e *F-measure* do sistema.

Tabela 4.6: Decisões tomadas acerca dos tipos das frases vindas de todos os reconhecedores de fala.

(a) Decisões do novo sistema para todas as frases.

	Decisão					
Verdade	"achieve"	"ask"	"ask_if"	"tell"	sem decisão	total
"achieve"	9	0	1	0	11	21
"ask"	2	23	6	14	102	147
"ask_if"	4	0	55	9	34	102
"tell"	0	0	12	62	40	114
total	15	23	74	85	187	

(b) Decisões do ALE para todas as frases.

	Decisão					
Verdade	"achieve"	"ask"	"ask_if"	"tell"	sem decisão	total
"achieve"	5	0	0	0	16	21
"ask"	0	13	0	0	134	147
"ask_if"	0	0	9	0	93	102
"tell"	0	0	0	10	104	114
total	5	13	9	10	347	

(c) Decisões do novo sistema só para as frases que o ALE também obtém resultados.

	Decisão					
Verdade	"achieve"	"ask"	"ask_if"	"tell"	sem decisão	total
"achieve"	5	0	0	0	0	5
"ask"	0	13	0	0	0	13
"ask_if"	0	0	9	0	0	9
"tell"	0	0	0	10	0	10
total	5	13	9	10	0	

Precisão obtida para todas as frases reconhecidas

A precisão do sistema diz respeito à quantidade de informação extraída pelo sistema que está efectivamente correcta, sendo definida por (Jurafsky e Martin, 2000, pg. 578):

$$precisão = \frac{número\ de\ respostas\ correctas\ dadas\ pelo\ sistema}{número\ de\ respostas\ dadas\ pelo\ sistema}$$

ou seja, nas Tabelas 4.6(a) e 4.6(b) diz respeito à análise por colunas.

Na Figura 4.3 são apresentados os resultados da análise à precisão do novo sistema. No eixo das abcissas, o segmento que se refere a "achieve" representa a coluna "achieve" da Tabela 4.6(a), o segmento designado por "ask" refere-se à coluna "ask" da mesma tabela, e assim sucessivamente. Por consulta à coluna "achieve", verificamos que o sistema para este tipo de frase classificou correctamente 9 frases como sendo deste tipo num total de 15, o que representa uma precisão de 60% (9/15) (Figura 4.3). Seguindo o mesmo raciocínio verificamos que a precisão para frases do tipo "ask" foi de 100% (23/23), para frases do tipo "ask_if" foi de 74% (55/74) e para frases do tipo "tell" 73% (62/85). Calculando a precisão global do sistema neste teste o resultado é:

$$\frac{9 + 23 + 55 + 62}{15 + 23 + 74 + 85} \times 100\% = 75,6\%$$

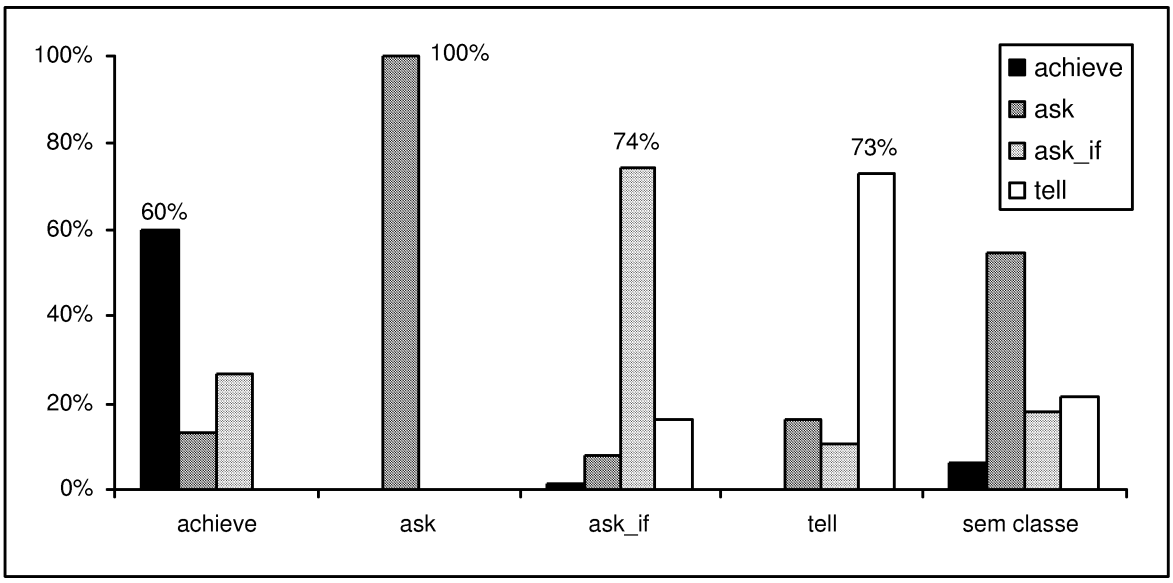


Figura 4.3: Gráfico da precisão do novo sistema para cada tipo de frase.

Como se verifica por observação da Tabela 4.6(b), a precisão alcançada pelo ALE é 100% para todos os tipos de frase, pois não comete quaisquer erros. As colunas "sem decisão" da Tabela 4.6(a) e 4.6(a) revelam o número de frases sobre as quais não foi tomada qualquer decisão.

Recall **obtido para todas as frases reconhecidas**

O *recall* mede a quantidade de informação relevante que o sistema extrai das frases, sendo portanto uma medida da cobertura do sistema. O seu valor é definido por (Jurafsky e Martin, 2000, pg. 578):

$$recall = \frac{número\ de\ respostas\ correctas\ dadas\ pelo\ sistema}{número\ total\ de\ respostas\ correctas\ possiveis}$$

Na Tabela 4.6(a) o *recall* diz respeito à análise por linhas. Para o tipo de frase "achieve", o novo sistema tomou 9 decisões correctas em 21 possíveis, o que perfaz um *recall* de 43%. Para o tipo de frase "ask" o *recall* é de 16% (23/147) e para o "ask_if" e para o "tell" 54%, 55/103 e 62/114, respectivamente. O *recall* global do novo sistema neste teste foi:

$$\frac{9 + 23 + 55 + 62}{21 + 147 + 102 + 114} \times 100\% = 38,8\%$$

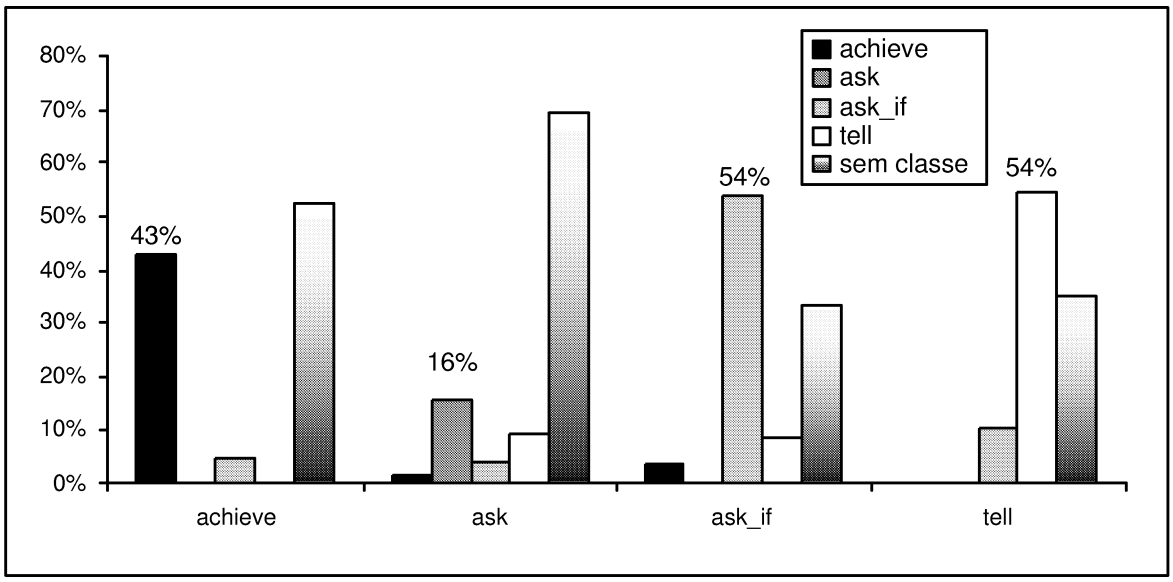


Figura 4.4: Gráfico da cobertura (*recall*) do novo sistema para cada tipo de frase.

O *recall* para o ALE, tal como se pode verificar por consulta da Tabela 4.6(b), para o tipo de frase "achieve" foi de 24% (5/21), para o "ask" 10% (13/134) e 9% para o "ask_if" e para o "tell" (9/102 e 10/114, respectivamente). O *recall* global obtido para o ALE neste teste foi:

$$\frac{5 + 13 + 9 + 10}{21 + 147 + 102 + 114} \times 100\% = 9,6\%$$

F-measure obtida para todas as frases reconhecidas

A *F-measure* conjuga as medidas de precisão e de cobertura segundo a expressão (Jurafsky e Martin, 2000, pg. 578):

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

onde P é o valor da precisão, R o valor do *recall* ou cobertura e β pode tomar valores superiores, inferiores ou iguais a 1. Se β for superior à unidade a precisão fica com maior influência no resultado que o *recall*, se β for inferior a 1 o *recall* tem mais influencia na *F-measure* que a precisão e se β tiver o valor 1 ambas as medidas têm o mesmo peso no valor da *F-measure*.

Como o objectivo deste trabalho é obter um sistema que tenha uma boa precisão sem sacrificar a cobertura, os valores da *F-measure* vão ser todos calculados com β igual a 1. Assim, a *F-measure* obtida pelo novo sistema para todas as frases reconhecidas é:

$$\frac{(1^2 + 1) \times 75,6\% \times 38,8\%}{1^2 \times 75,6\% + 38,8\%} = 51,3\%$$

e para o actual sistema do Carl baseado no ALE é:

$$\frac{(1^2 + 1) \times 100\% \times 9,6\%}{1^2 \times 100\% + 9,6\%} = 17,5\%$$

Como se verifica, o novo sistema tem melhor cobertura (*recall*) que o ALE, mas uma pior precisão. No entanto o valor alcançado para a *F-measure* pelo novo sistema é quase 3 vezes superior ao obtido pelo sistema em utilização no Carl. O desempenho menos conseguido na precisão é devido ao novo sistema tentar extrair informação também de frases que não estão completamente correctas do ponto de vista gramatical e portanto tem uma maior probabilidade de errar.

Os resultados foram apresentados para todas as frases, independentemente do WER de cada uma delas. Para percebermos melhor como evolui o comportamento do sistema em termos de precisão e de cobertura à medida que se tenta analisar frases com mais erros, de seguida vão ser mostrados resultados sobre a precisão, sobre o *recall* e *F-measure* para ambos os sistemas e para frases com WER até 25%, 50% e 75%.

Os resultados obtidos para frases com um WER inferior a 25% estão apresentados nas Tabelas 4.7(a) e 4.7(b), estando o resumo da respectiva análise nas Tabelas 4.8(a) e 4.8(b). Conforme se pode verificar, a precisão do novo sistema é 91,3% e a cobertura 84,0%. Estes valores conduzem a uma *F-measure* de 87,5%. O sistema actual do Carl, como se tinha verificado anteriormente, tem uma precisão de 100% mas a sua cobertura é inferior a 50% mesmo para frases com um WER inferior a 25%. Estes valores fazem com que a *F-measure* se situe nos 66,0%.

Os resultados obtidos com frases cujo o WER é inferior a 50% (ver Tabelas 4.10(a) e 4.10(b)) foram, para o novo sistema, uma precisão de 86,1%, um *recall* de 68,2% e uma *F-measure* de 76,1%. Para o mesmo conjunto de frases, o sistema actual do Carl mantém uma

Tabela 4.7: Decisões tomadas sobre os tipos de frases com WER inferior a 25%.

(a) Novo sistema.

	Decisão					
Verdade	"achieve"	"ask"	"ask_if"	"tell"	sem decisão	total
"achieve"	5	0	0	0	0	5
"ask"	0	19	0	1	4	24
"ask_if"	0	0	17	0	1	18
"tell"	0	0	5	22	1	28
total	5	19	22	23	6	75

(b) Sistema implementado no Carl (ALE).

	Decisão					
Verdade	"achieve"	"ask"	"ask_if"	"tell"	sem decisão	total
"achieve"	5	0	0	0	0	5
"ask"	0	13	0	0	11	24
"ask_if"	0	0	9	0	9	18
"tell"	0	0	0	10	18	28
total	5	13	9	10	38	75

Tabela 4.8: Resultados da precisão, *recall* e *F-measure* para as frases com WER inferior a 25%.

(a) Novo sistema.

Tipo de Frase	**Precisão**	*Recall*
"achieve"	100% (5/5)	100% (5/5)
"ask"	100% (19/19)	79% (19/24)
"ask_if"	77% (17/22)	94% (17/18)
"tell"	96% (22/23)	79% (22/28)
todos	91,3% (63/69)	84,0% (63/75)
F-measure	87,5%	

(b) Sistema implementado no Carl (ALE).

Tipo de Frase	**Precisão**	*Recall*
"achieve"	100% (5/5)	100% (5/5)
"ask"	100% (13/13)	54% (13/24)
"ask_if"	100% (9/9)	50% (9/18)
"tell"	100% (10/10)	36% (10/28)
todos	100% (37/37)	49,3% (37/75)
F-measure	66,0%	

Tabela 4.9: Decisões tomadas sobre os tipos de frases com WER inferior a 50%.

(a) Novo sistema.

	Decisão					
Verdade	"achieve"	"ask"	"ask_if"	"tell"	sem decisão	total
"achieve"	5	0	0	0	0	5
"ask"	0	23	5	6	19	53
"ask_if"	0	0	43	1	8	52
"tell"	0	0	7	47	9	63
total	5	23	55	54	36	173

(b) Sistema implementado no Carl (ALE).

	Decisão					
Verdade	"achieve"	"ask"	"ask_if"	"tell"	sem decisão	total
"achieve"	5	0	0	0	0	5
"ask"	0	13	0	0	40	53
"ask_if"	0	0	9	0	43	52
"tell"	0	0	0	10	53	63
total	5	13	9	10	136	173

Tabela 4.10: Resultados da precisão, *recall* e *F-measure* para as frases com WER inferior a 50%.

(a) Novo sistema.

Tipo de Frase	**Precisão**	*Recall*
"achieve"	100% (5/5)	100% (5/5)
"ask"	100% (23/23)	43% (23/53)
"ask_if"	78% (43/55)	83% (43/52)
"tell"	87% (47/54)	75% (47/63)
todos	86,1% (118/137)	68,2% (118/173)
F-measure	76,1%	

(b) Sistema implementado no Carl (ALE).

Tipo de Frase	**Precisão**	*Recall*
"achieve"	100% (5/5)	100% (5/5)
"ask"	100% (13/13)	25% (13/53)
"ask_if"	100% (9/9)	17% (9/52)
"tell"	100% (10/10)	16% (10/63)
todos	100% (37/37)	21,4% (37/173)
F-measure	35,2%	

precisão de 100% e obtém um *recall* e uma *F-measure* de 21,4% e 35,2%, respectivamente. A acentuada quebra no *recall*, quando comparada com o obtido para um WER inferior a 25% (de 66,0% para 21,4%), e por consequência da *F-measure*, deve-se à incapacidade do ALE extrair qualquer informação de frases com erros. Assim, e como se pode reparar por comparação das Tabela 4.7(b) e 4.9(b), verificamos que não foram analisadas quaisquer frases com um WER entre os 25% e os 50%. Na realidade, neste teste o sistema em uso no Carl apenas conseguiu extrair informação de frases com um WER de 0%.

Tabela 4.11: Decisões tomadas sobre os tipos de frases com WER inferior a 75%.

(a) Novo sistema.

	Decisão					
Verdade	"achieve"	"ask"	"ask_if"	"tell"	sem decisão	total
"achieve"	5	0	0	0	0	5
"ask"	1	23	6	14	37	81
"ask_if"	3	0	53	4	11	71
"tell"	0	0	10	55	14	79
total	9	23	69	73	62	236

(b) Sistema implementado no Carl (ALE).

	Decisão					
Verdade	"achieve"	"ask"	"ask_if"	"tell"	sem decisão	total
"achieve"	5	0	0	0	0	5
"ask"	0	13	0	0	68	81
"ask_if"	0	0	9	0	62	71
"tell"	0	0	0	10	69	79
total	5	13	9	10	199	236

Por consulta às Tabelas 4.12(a) e 4.12(b), verifica-se que para o conjunto de frases com um WER inferior a 75%, o novo sistema teve uma precisão de 78,2%, um *recall* de 57,6% e uma *F-measure* de 66,3%. Com o mesmo conjunto de frases, o sistema actual do Carl teve uma precisão de 100%, um *recall* de 15,7% e uma *F-measure* de 27,1%.

Foi também verificada a distribuição das frases sem decisão em função do WER. Os resultados estão ilustrados na Figura 4.5. O gráfico mostra que à medida que o WER aumenta, o número de frases sem decisão também aumenta. Este resultado era esperado uma vez que quantos mais erros uma frase tiver, mais difícil é para o sistema conseguir extrair informação coerente desta.

Teste 3: qualidade da informação semântica extraída

O último teste efectuado incidiu sobre as relações semânticas extraídas. Para verificar se o comportamento do sistema desenvolvido é conforme a especificação inicial, foram primeiro analisadas as relações extraídas das frases reconhecidas correctamente pelos reconhecedores de fala. Os resultados são apresentados na Tabela 4.13.

Tabela 4.12: Resultados da precisão, *recall* e *F-measure* para as frases com WER inferior a 75%.

(a) Novo sistema.

Tipo de Frase	**Precisão**	*Recall*
"achieve"	56% (5/9)	100% (5/5)
"ask"	100% (23/23)	28% (23/81)
"ask_if"	77% (53/69)	75% (53/71)
"tell"	75% (55/73)	70% (55/79)
todos	78,2% (136/174)	57,6% (136/236)
F-measure	66,3%	

(b) Sistema implementado no Carl (ALE).

Tipo de Frase	**Precisão**	*Recall*
"achieve"	100% (5/5)	100% (5/5)
"ask"	100% (13/13)	16% (13/81)
"ask_if"	100% (9/9)	13% (9/71)
"tell"	100% (10/10)	13% (10/79)
todos	100% (37/37)	15,7% (37/236)
F-measure	27,1%	

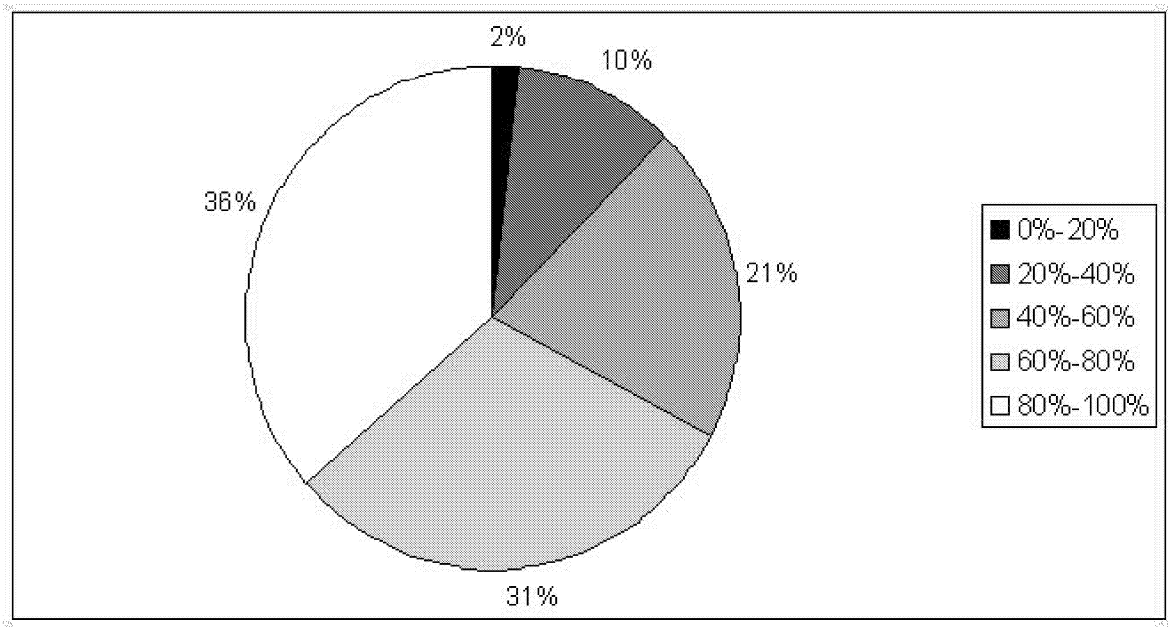


Figura 4.5: Distribuição em função do WER das frases para as quais não foram tomadas decisões.

Tabela 4.13: Quantidade de relações semânticas extraídas das frases bem reconhecidas.

	Relações Certas	**Relações Erradas**
análise ideal	111	0
novo sistema	107	1
sistema do Carl (ALE)	111	0

Como se pode observar, o sistema desenvolvido errou uma relação e não conseguiu extrair 3 (111-(107+1)). Uma análise mais cuidada ao sucedido revelou que estas falhas foram motivadas por uma incorrecta avaliação da categoria sintáctica de certas palavras, ou seja erros provocados pelos anotadores morfossintácticos. Este resultado indica que o sistema acertou em 96,4% (107/111) das relações quando as fases eram correctas.

Foi também estudada a variação das relações semânticas extraídas em função do WER. Como se observa na Figura 4.6 até um WER de 40%-60%, o sistema consegue extrair mais relações semânticas certas que erradas. A partir deste limiar são extraídas sensivelmente tantas relações semânticas certas como erradas.

Uma observação aos ficheiros de resultados revela que é vulgar as relações semânticas extraídas, para frases com um WER entre 20% e 60%, não conterem por si só uma mensagem completa, ou seja, uma mensagem com informação suficiente para o robô decidir o que fazer. Contudo a informação contida nas relações semânticas obtidas nestes casos, são suficientes para permitir ao robô fazer perguntas acerca do que foi dito e assim completar a informação em falta. É mais adequado o robô fazer perguntas para tentar perceber a mensagem que a pessoa lhe está a transmitir que simplesmente não reagir.

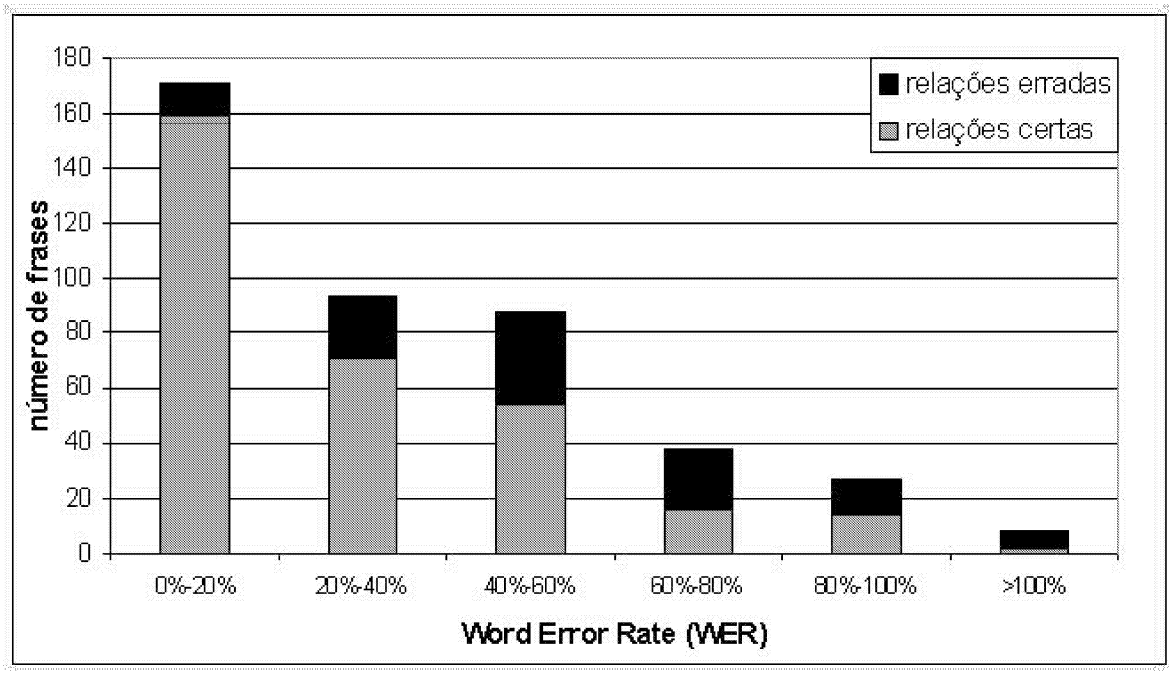


Figura 4.6: Histograma das relações semânticas extraídas em função do WER das frases.

4.3 Discussão

Conjugando os resultados das duas secções anteriores e tendo apenas em conta o reconhecedor com melhor desempenho e o novo sistema de NLU e comparando com o sistema actualmente em utilização, temos o seguinte:

Tabela 4.14: Resumo dos resultados obtidos com três configurações representativas.

	Sistema Inicial	**Evolução 1**	**Evolução 2**
Reconhecimento de Fala	Nuance $\overline{WER}$: 79,1%	Nuance $\overline{WER}$: 79,1%	Sphinx2 com Treino $\overline{WER}$: 38,3%
Compreensão de Linguagem Natural	ALE frases analisadas: 5% (7/128)	Sistema Híbrido frases analisadas: 30% (38/128)	Sistema Híbrido frases analisadas: 76% (97/128)

Os resultados mostram claramente que houve uma evolução a dois níveis: reconhecedor e compreensão e que quando utilizados em conjunto os efeitos são cumulativos. Os resultados também mostram vantagem do novo sistema de compreensão mesmo se aplicado a um sistema de reconhecimento com um WER elevado, como se comprova pelos resultados da coluna intitulada "Evolução 1".

Capítulo 5

Conclusões

5.1 Resumo do Trabalho Apresentado

Neste documento foi proposto um sistema híbrido de compreensão de linguagem natural falada. O objectivo deste sistema é aumentar a capacidade de compreensão de informação de um robô. A abordagem seguida até à data no robô não permitia compreender parcialmente ou totalmente uma frase, sempre que esta apresentasse qualquer erro sintáctico. Como se sabe, é frequente que as hipóteses de frase vindas de reconhecedores de fala contenham erros devidos a dificuldades inerentes ao reconhecimento de fala, o que provoca que o robô ignorasse frequentemente as frases que lhe eram dirigidas. Com o intuito de resolver este problema foi concebido um novo módulo de compreensão de linguagem natural tolerante a erros nas frases e, posteriormente, foi melhorado o desempenho do reconhecimento da fala.

O enquadramento feito pela evolução da interface de fala do robô Carl pretende ajudar a compreender melhor o problema a resolver: como entender um utilizador quando frequentemente apenas se obtêm fragmentos do que foi dito? Se pretendermos utilizar interfaces de fala em qualquer ambiente, o módulo de compreensão das frases tem forçosamente de lidar com apenas fragmentos de informação, à semelhança dos seres humanos quando inseridos em ambientes ruidosos. Em virtude disto, o novo módulo de compreensão de linguagem utiliza soluções das duas principais filosofias adoptadas na área: orientada a dados e orientada ao conhecimento. A ideia central da abordagem proposta é aproveitar as vantagens inerentes a cada filosofia para compensar as falhas que cada uma delas possui isoladamente. Assim, o sistema híbrido obtido consegue extrair informação detalhada de frases correctas e alguma informação de frases com erros de estrutura, onde apenas se encontram fragmentos de informação.

O desempenho do reconhecimento de fala foi melhorado através do treino dos modelos acústicos do reconhecedor de fala. Os modelos acústicos ficam assim melhor adaptados a um utilizador português a falar a língua inglesa. O treino efectuado foi apenas para um utilizador devido a não existirem à data mais pessoas disponíveis, mas revela que é uma solução válida para melhorar o desempenho desta ferramenta e que o treino deve ser alargado a um maior

grupo de utilizadores.

Foi também feito um esforço para avaliar quantitativamente e com objectividade o desempenho da solução proposta, quer o módulo de compreensão de linguagem natural falada quer os novos modelos acústicos para o reconhecedor de fala. Estes desempenhos foram comparados com o desempenho da solução anterior de modo a se poder verificar se o sistema proposto é mais adequado à compreensão de linguagem natural falada.

5.2 Resultados Mais Importantes

O sistema desenvolvido, mesmo sem informação lexical definida especificamente para a aplicação de teste, conseguiu extrair 96.4% das relações possíveis (107 em 111), sempre que a frase estava bem reconhecida tendo apenas falhado 1 relação. Existiram frases bem reconhecidas que o novo sistema conseguiu processar e que o sistema "ideal" não conseguiu. Como não existia termo de comparação estas frases não foram consideradas para o teste, mas pela sua observação verifica-se que a análise obtida é válida. Isto aconteceu devido a um lapso na definição lexical do sistema ideal, sendo precisamente um dos pontos fortes do novo sistema: tolerância a falhas do programador, característica herdada da abordagem baseada em dados. Neste caso específico são os anotadores morfossintácticos que possibilitam ao sistema saber a categoria sintáctica de qualquer palavra, permitindo a sua inclusão na análise das frases.

Outro resultado importante foram os casos em que o sistema em utilização no Carl, um sistema baseado em conhecimento, não conseguiu obter resultados ao invés do sistema em estudo. Isto verificou-se sempre que a frase não era totalmente bem reconhecida. O sistema desenvolvido conseguiu obter bons resultados para frases com um WER até 50%: pelo menos 75% das relações extraídas são correctas. Repare-se que um WER de 50% implica que apenas 50% das palavras presentes na frase é que foram efectivamente pronunciadas. Apesar de algumas análises não conseguirem retirar informação que possibilite ao módulo de inferência lógica tomar decisões, pelo menos fornecem pistas sobre a informação da frase, permitindo ao robô um eventual esclarecimento nas seguintes iterações do diálogo.

Podemos concluir, mesmo sem o sistema contar com o nível de confiança do reconhecedor de fala na frase, que os resultados obtidos são bons. O sistema teve um comportamento ao nível do sistema baseado no ALE quando as frases eram correctas do ponto de vista gramatical, mostrando também que consegue retirar informação correcta de frases com erros de estrutura. Este avanço permitiu compreender, considerando frases com um WER até 50%, mais 402.6% de frases que anteriormente. Ao nível do reconhecimento de fala, também foram obtidas melhores taxas de WER devido a algum trabalho ao nível do treino dos modelos acústicos.

O sistema de compreensão de linguagem natural apresentado é funcional e está presentemente em uso no robô Carl, tendo substituído o sistema anterior baseado no ALE. É de referir que este sistema apenas pressupõe como fonte de conhecimento uma pequena gramática baseada em etiquetas morfossintácticas e alguns exemplos de frases para treinar um classificador de dados. Apesar do sistema ter sido testado para o Inglês, não é definida na

arquitectura qualquer informação para aquela língua, permitindo assim a migração do sistema para outras línguas como o Português. Para colocar o sistema a funcionar é necessário um etiquetador (pelo menos) para a língua desejada, construir uma pequena gramática e obter alguns exemplos de frases com a análise sintáctica desejada, para além de um reconhecedor de fala.

5.3 Trabalho Futuro

Um aspecto que deverá ser afinado no actual sistema é a anotação morfossintáctica. Uma vez que uma fonte importante de informação do sistema são os anotadores morfossintácticos, estes deverão ser treinados ou ajustados melhor ao contexto do Carl. Como exemplo, o erro cometido na obtenção da relação de uma frase correcta derivou de uma incorrecta anotação morfossintáctica. Uma vez que o sistema já está a ser utilizado no Carl, podem ser recolhidas frases para treinar os anotadores morfossintácticos.

Outra tarefa a executar deverá ser a simplificação da estrutura global do sistema. Testes efectuados mostram que tanto o algorimto IB-1 do TiMBL como o LCFlex têm contributos importantes na análise sintáctica (Rodrigues *et al.*, 2004), mas não é tão claro que o algoritmo IGTREE do TiMBL tenha um papel importante ao decidir se a frase é gramatical ou não. Esta solução foi herdada de contribuições anteriores, mas com o evoluir do sistema poderá ter perdido a sua relevância. Acontece que a análise proveniente do *parser* é quantificada, podendo a frase ser encaminhada ou não para uma análise mais conservadora, não se justificando assim uma decisão sobre a hipotética validade da análise do *parser* antes desta etapa. Inclusivamente já foram efectuados alguns testes preliminares que revelam que o desempenho do sistema não é significativamente alterado com a remoção do IGTREE. No entanto são necessários testes mais exaustivos antes de se poder tomar qualquer decisão.

Este trabalho incidiu pouco sobre a representação semântica das frases pois, para poder ser integrado no robô, esta teve de ser idêntica à utilizada pelo sistema anterior. Contudo, um trabalho porventura interessante no futuro seria mudar a representação semântica para um formato mais difundido como por exemplo lógica de primeira ordem, em Inglês First Order Logic (FOL). A FOL é adoptada por vários sistemas que necessitam de representação semântica. Por ser difundida, ao invés do formato presentemente utilizado no Carl, existem ferramentas disponíveis livremente baseadas em FOL que poderiam ser úteis no desenvolvimento do robô. Ferramentas que permitem construir modelos de conhecimento (em Inglês *model builder*) como o MACE-2.0 (McCune, 2001) e o Paradox 1.0 (Claessen e Sörensson, 2003) e demonstrar teoremas (em Inglês *theorem prover*) como o Otter-3.3 (Kalman, 2001; McCune, 2003) e o Bliksem (de Nivelle, 2003).

Bibliografia

J. Allen (1994). *Natural Language Understanding*. The Benjamin/Cummings Publishing Company, Inc., 390 Bridge Parkway, Redwood City, CA 94065, segunda edição.

J. R. Bellegarda e K. E. A. Silverman (2003). Natural language spoken interface control using data-driven semantic inference. *IEEE Transactions on Speech and Audio Processing*, 11(3):267–277.

R. Bod (1998). Spoken dialogue interpretation with the dop model. In C. Boitet e P. Whitelock (editores), *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, páginas 138–144. Morgan Kaufmann Publishers, San Francisco, California.

R. Bod (2000). Parsing with the shortest derivation. In *Proceedings COLING'2000*. Saarbrcken, Germany.

R. Bod e R. Scha (1996). Data-oriented language processing: An overview. Relatório técnico, Department of Computational Linguistics, ILLC, University of Amsterdam.

J. Bos, E. Klein, e T. Oka (2003). Meaningful conversation with a mobile robot. In *Proceedings of the EACL*.

E. Brill (1995). Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*.

T. Brøndsted (1999). The CPK NLP suite for spoken language understanding. In *Eurospeech, 6th European Conference on Speech Communication and Technology*, páginas 2655–2658.

G. Bugmann, S. Lauria, T. Kyriacou, E. Klein, J. Bos, e K. Coventry (2001). Using verbal instruction for route learning: Instruction analysis. Relatório Técnico UMC-01-4-1, Department of Computer Science, Manchester University.

B. Carpenter e G. Penn (2001). ALE - the attribute logic engine user's guide. Relatório Técnico Version 3.2.1, SpeechWorks Research and Department of Computer Science, University of Toronto.

J. Carroll e T. Briscoe (1993). Generalized probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19(1):25–59.

D. Chiang (2000). Statistical parsing with an automatically-extracted tree adjoining grammar. In *38th Annual Meeting of the Association for Computational Linguistics*, páginas 456–463.

D. G. Childers (1997). *Probability and Random Processes*. McGraw-Hill.

K. Claessen e N. Sörensson (2003). New techniques that improve MACE-style finite model finding. *Model Computation*.

P. R. Clarkson e R. Rosenfeld (1997). Statistical language modeling using the CMU-cambridge toolkit. In *Proceedings ESCA Eurospeech*.

R. A. Cole, J. Mariani, H. Uszkoeit, A. Zaenen, V. Zue, G. Varile, e A. Zampolli (editores) (1995). *Survey of the State of the Art in Human Language Technology*, capítulo 3. Cambridge University Press.

D. V. Compernolle (1999). Speech recognition by goats, wolves, sheep and ... non-natives. In *Multi-lingual Interoperability in Speech Technology*.

W. Daelemans, J. Zavrel, A. van den Bosch, e K. van der Sloot (2002). *TiMBL: Tilburg Memory-Based Learner Reference Guide, version 4.2*. Induction of Linguistic Knowledge, Computational Linguistics, Tilburg University, P.O. Box 90153, NL-5000 LE, Tilburg, The Netherlands.

H. de Nivelle (2003). *Bliksem 1.10 User Manual*.

S. J. DeRose (1988). Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1):31–39.

L. Ferreira, P. Soares, M. Rodrigues, L. Seabra Lopes, e A. Teixeira (2002). Desenvolvimentos recentes na interface humano-robô baseada em linguagem falada do robô Carl. *Revista do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro*.

N. M. Ferreira, N. F. Ferreira, M. Rodrigues, A. Teixeira, F. Vaz, e L. Seabra Lopes (2001). Interacção com um robô através de linguagem natural falada. *Revista do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro*, 3(3).

N. M. F. Ferreira e N. F. P. da Silva Ferreira (2000). Interface humano máquina do CARL. Relatório final de projecto 5^o ano, Departamento de Electrónica e Telecomunicações da Universidade de Aveiro, Instituto de Engenharia Electrónica e Telemática de Aveiro.

S. Fujie, D. Yagi, Y. Matsusaka, H. Kikuchi, e T. Kobayashi (2004). Spoken dialogue system using prosody as para-linguistic information. In *International Conference on Speech Prosody (SP2004)*.

M. R. Genesereth e S. P. Ketchpel (1994). Software agents. *Communications of the Association for Computing Machinery*, 37(7):48–53.

Godot (2003). Godot, the robot - school of informatics, university of edimburgh. http://www.ltg.ed.ac.uk/godot/.

E. B. Gouvêa (2000). http://www.speech.cs.cmu.edu/sphinx/models/hub4opensrc_-jan2002/INFO_ABOUT_MODELS.

J. Henderson e P. Lane (1998). A connectionist architecture for learning to parse. In *COLING-ACL*, páginas 531–537.

J. B. Henderson (2000). A neural network parser that handles sparse data. In *Proceedings of the 6th International Workshop on Parsing Technologies*. Trento, Italy.

IBL (2004). IBL: Instruction-based learning for mobile robots. http://www.tech.plym.ac.uk/soc/staff/guidbugm/ibl/index.html.

Insite (2001). Insite soluções internet - processamento de linguagem natural. http://linguistica.insite.com.br/nlp.phtml.

Jurafsky e Martin (2000). *Speech and Language PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.

J. A. Kalman (2001). *Automated Reasoning with OTTER*. Rinton Press.

P. Lamere, P. Kwok, W. Walker, E. Gouvêa, R. Singh, B. Raj, e P. Wolf (2003). Design of the CMU sphinx-4 decoder. In *Eurospeech*.

A. Lavie (1996). *GLR*: A Robust Grammar-Focused Parser for Spontaneously Spoken Language*. Tese de Doutoramento, School of Computer Science, Carnegie Mellon University.

L. Lee (2003). I'm sorry Dave, I'm afraid I can't do that. *National Research Council study on the Fundamentals of Computer Science*. To appear.

D. L. Martin, A. J. Cheyer, e D. B. Moran (1999). The open agent architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, 13(1-2):91–128. OAA.

T. Matsui, H. Asoh, J. Fry, Y. Motomura, F. Asano, T. Kurita, I. Hara, e N. Otsu (1999). In *Proceedings of AAAI*.

W. McCune (2001). *MACE 2.0 Reference Manual and Guide*. Argonne National Laboratory - Mathematics and Computer Science Division.

W. McCune (2003). *OTTER 3.3 Reference Manual*. Argonne National Laboratory - Mathematics and Computer Science Division.

H. Meinedo (2000). *Utilização de Informação Silábica no Reconhecimento de Fala Contínua*. Tese de Mestrado, Instituto Superior Técnico, Universidade Técnica de Lisboa.

F. Michaud e D. A. Gustafson (2002). The hors d'oeuvres event at the aaai-2001 mobile robot competition. *AI magazine*, 23(1):31–35.

A. Mikheev (1994). Automatic rule induction for unknown word guessing. *Computational Linguistics*, 16(1).

L. Màrquez, L. Padró, e H. Rodríguez (2000). A machine learning approach to pos tagging. *Machine Learning*, 39(1):59–91. Kluwer Academic Publishers.

G. V. Noord (1997). An efficient implementation of the head-corner parser. *Computational Linguistics*, 23(3):425–456.

Nuance (2001a). *Application Developer's Guide: Nuance Speech Recognition System Version 8.0*. Nuance.

Nuance (2001b). *Grammar Developer's Guide: Nuance Speech Recognition System Version 8.0*. Nuance.

Nuance (2001c). *Nuance Platform Integrator's Guide: Nuance Speech Recognition System Version 8.0*. Nuance.

J. Odell, D. Wood, D. Kershaw, D. Ollason, V. Valtchev, e D. Whitehouse (1998). *the hapi book: a description of the HTK application programming interface, version 1.3*. Entropic.

T. Peijnenburg (2003). Philips CFT robocup team description: preliminary proceedings 2003 robocup conference. http://borneo.ais.fraunhofer.de/robocup/msl2003/QualificationMSL2003/tdp_philips.pdf.

Philips (2002). Philis CFT - centre for industrial technology. http://www.cft.philips.com/robocup/.

K. Power, R. Morton, C. Matheson, e D. Ollason (1998). *the graphvite book (for version 1.1)*. Entropic.

Priberam (2004). Priberam informática - língua portuguesa on-line: Dicionário. http://www.priberam.pt/dlpo/dlpo.aspx.

J. G. Proakis e D. G. Manulakis (1996). *Digital Signal Processing: Principles, Algorithms, and Applications*, páginas 29–33. Prentice Hall.

L. Rabiner e B.-H. Juang (1993). *Fundamentals of Speech Recognition*. Prentice Hall.

L. R. Rabiner (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Robita (2002). Project ROBITA (Real-world Oriented BI-modal Talking Agent). http://www.pcl.cs.waseda.ac.jp/robita/.

M. Rodrigues (2000). Colaboração no desenvolvimento de uma interface de voz. Relatório de estágio, Departamento de Matemática Aplicada da Universidade do Porto, Instituto de Engenharia Electrónica e Telemática de Aveiro.

M. Rodrigues, L. Seabra Lopes, e A. Teixeira (2002). Interacção humano-robô ao nível da linguagem. In *Robotica 2002*.

M. Rodrigues, A. Teixeira, e L. Seabra Lopes (2004). An hybrid approach for spoken natural language understanding applied to a mobile intelligent robot. In *Natural Language Understanding and Congnitive Science, Proceedings of the 1st International Workshop*.

C. P. Rosé (1997). *Robust Interactive Dialogue Interpretation*. Tese de Doutoramento, Graduate Faculty of the Language Technologies Institute, Carnegie Mellon University.

C. P. Rosé e A. Lavie (1998). *LCFlex: An Efficient Robust Left-Corner Parser*.

C. P. Rosé e A. Lavie (2001). Balancing robustness and efficiency in unification-augmented context-free parsers for large pratical applications. In J.-C. Junqua e G. van Noord (editores), *Robustness in Language and Speech Technology*. Kluwer Academic Publishers.

L. Seabra Lopes e A. Teixeira (2000). Teaching behavioral and task knowledge to robots through spoken dialogues. In S. LuperFoy e D. Miller (editores), *My Dinner with R2D2: Natural Dialogues with Practical Robotic Devices*, AAAI-2000 Spring Symposium Series, páginas 35–43. Stanford University, Stanford, California.

L. Seabra Lopes, A. Teixeira, M. Rodrigues, D. Gomes, J. Girão, C. Teixeira, N. Sénica, L. Ferreira, e P. Soares (2003a). A robot with natural interaction capabilities. In *9th IEEE International Conference on Emerging Technologies and Factory Automation*.

L. Seabra Lopes, A. Teixeira, M. Rodrigues, D. Gomes, C. Teixeira, L. Ferreira, P. Soares, J. Girão, e N. Sénica (2003b). Towards a personal robot with language interface. In *Eurospeech*.

SMAPI (2000a). *SMAPI Reference: IBM ViaVoice Software Developer's Kit*. IBM.

SMAPI (2000b). *SMAPI User's Guide: IBM ViaVoice Software Developer's Kit*. IBM.

P. Soares (2002). Comando e ensino de um robô através de linguagem natural falada. Relatório de estágio, Departamento de Matemática Aplicada da Universidade do Porto, Instituto de Engenharia Electrónica e Telemática de Aveiro.

SPASS (2004). http://spass.mpi-sb.mpg.de/download/index.html.

P. A. Taylor, A. Black, e R. Caley (1998). The architecture of the festival speech synthesis system. *The Third ESCA Workshop in Speech Synthesis*, páginas 147–151.

A. Teixeira, L. Seabra Lopes, L. Ferreira, P. Soares, e M. Rodrigues (2003). Recent developments on the spoken language human-robot interface of the robot Carl. In *Robotica 2003*.

A. J. S. Teixeira e F. Vaz (2004). Acetatos das aulas de processamento digital de voz. http://www.ieeta.pt/~ajst/pdv/. Universidade de Aveiro.

ViaVoice (2000). *ViaVoice^{TM}Speech Developer's Tools Guide*. IBM, primeira edição.

ViaVoiceTTs (2000). *ViaVoice^{TM}TTS API Reference*. IBM, sixt edição.

A. Voutilainen (1997). EngCG tagger, version 2. In T. Bronsted e I. Lytje (editores), *Sprog og Multimedier*. Aalborg Universitetsforlag.

Apêndice A

Aprendizagem Automática

A aprendizagem automática (em Inglês *machine learning*) é o estudo de algoritmos para computadores que têm a capacidade de melhorarem o seu desempenho através da experiência. Várias técnicas de *machine learning* têm sido aplicadas a diversas áreas da NLU. Nesta secção vão ser referidas duas técnicas. Uma delas é utilizada no novo sistema para *parsing* sintáctico, a MBL, sendo a outra utilizada para a anotação morfossintáctica, a TBL.

A.1 Aprendizagem Baseada em Memórias

A aprendizagem baseada memórias, em Inglês Memory Based Learning (MBL), tem sido utilizada em várias tarefas de NLU. Experiências com esta técnica já tinham sido feitas com bons resultados no Carl (Teixeira *et al.*, 2003; Seabra Lopes *et al.*, 2003b), encorajando a sua utilização no módulo em desenvolvimento.

Os algoritmos de MBL provêem directamente do algoritmo k-NN, conhecido por ser muito eficaz a classificar dados numéricos. Baseiam-se na hipótese que um comportamento inteligente pode ser obtido utilizando raciocínio por analogias, em alternativa à aplicação de "regras mentais" abstractas como acontece no processamento baseado em regras. Mais concretamente, a MBL assenta na ideia que a extrapolação de experiências passadas para novas situações, baseada na semelhança entre a situação nova e as situações já vividas, é de extrema importância (Daelemans *et al.*, 2002).

Na figura A.1 está esquematizada a arquitectura de um sistema de MBL. A componente de aprendizagem adiciona instâncias de treino à memória. Este processo é usualmente designado de aprendizagem preguiçosa (em Inglês *lazy learning*) uma vez que é feito sem qualquer tipo de abstracção ou reestruturação dos dados. Uma instância é um vector de comprimento fixo com n pares característica/valor mais um campo com a classe atribuída a cada vector de pares característica/valor.

A componente de classificação é o produto da componente de aprendizagem. A sua função é calcular a saída correspondente a cada entrada do sistema, ou seja classificar os dados de entrada. Este processo pode ser efectuado quer para uma ocorrência conhecida, presente no

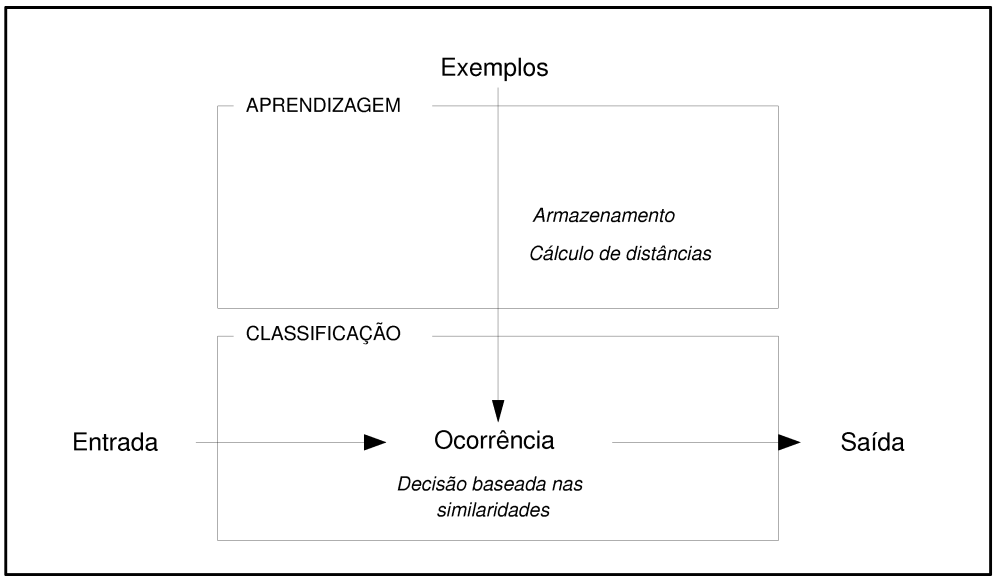


Figura A.1: Arquitectura geral de um sistema de aprendizagem baseada em memória

conjunto de treino, quer para uma ocorrência desconhecida. A similaridade entre a ocorrência X a classificar e os exemplos Y contidos na memória do sistema é calculada utilizando uma métrica $\Delta(X, Y)$. A extrapolação é efectuada pela atribuição da categoria mais frequente nos k exemplo(s) mais similar(es), designados também de vizinhos, à ocorrência a classificar.

A.2 Aprendizagem Baseada em Transformações

Em Jurafsky e Martin (2000) está uma analogia atribuída a Terry Harvey que facilita a compreensão do paradigma da TBL. Imaginemos um artista a pintar uma imagem de uma casa branca com estores verdes sob um céu azul. Suponhamos que a maioria da área do quadro é ocupada pelo céu azul. O pintor poderia começar a pintar toda a tela de azul com um pincel muito largo. Depois poderia pegar um pincel um mais estreito e pintar a casa toda de branco, sem se preocupar com as cores do tecto castanho ou dos estores verdes. De seguida pegaria num pincel ainda mais estreito para pintar o tecto de castanho. Finalmente pegaria no pincel mais estreito para colorir o estore.

Na analogia, o pintor inicia a pintura com um pincel largo que cobre muita tela mas que pinta áreas que têm de ser repintadas. Na iteração seguinte, a pintura da casa, cobre menos tela mas comete menos erros. Em cada nova iteração é utilizado um pincel mais estreito que corrige menos a imagem mas que também comete menos erros. Na TBL são aplicadas uma sucessão de regras. A primeira regra a ser aplicada é a mais lacta, a que descreve melhor o maior número de casos. Depois é aplicada uma sucessão de regras cada vez mais específicas (Jurafsky e Martin, 2000, p. 307).

A arquitectura base de um sistema baseado em TBL encontra-se esquematizada na figura A.2. Primeiro os exemplos são classificados por um classificador inicial. A complexidade do

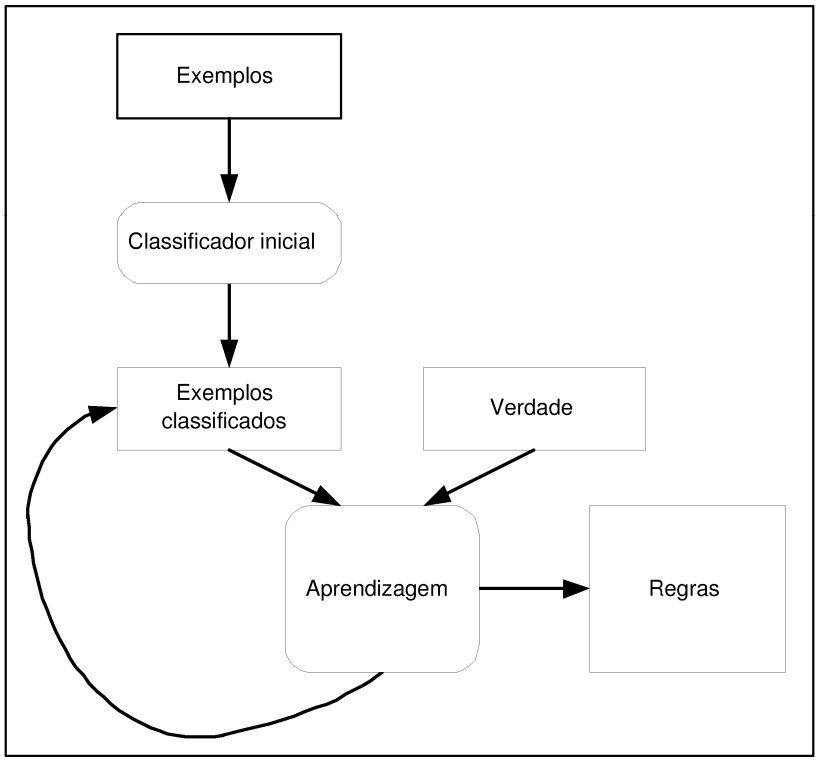


Figura A.2: Esquema da TBL.

classificador inicial poderá variar desde uma ferramenta muito simples até uma bastante sofisticada. Depois os exemplos são comparados à verdade, uma lista com os exemplos e respectivas classificações. Finalmente são aprendidas iterativamente as transformações necessárias para corrigir os erros do classificador inicial. Na construção do classificador são definidas as estruturas das transformações possíveis, por exemplo para o caso da anotação morfossintáctica: mudar da etiqueta x para a etiqueta y quando a palavra ou etiqueta anterior for z. Em cada iteração da aprendizagem, verifica-se qual das transformações da lista é a conduz a uma maior diminuição dos erros. Essa transformação é aplicada e acrescentada à lista de transformações aprendidas. A aprendizagem termina quando não for possível melhorar o desempenho do anotador (Brill, 1995).

Para se proceder à classificação dos exemplos, estes são primeiro classificados pelo classificador inicial, que tem de ser o mesmo do utilizado para a aprendizagem as regras. Depois aplicam-se as transformações pela ordem em que foram aprendidas até se acabar a sequência completa de regras ou não existir mais nenhuma regra válida a aplicar ao caso em questão.

Apêndice B

Informação Utilizada no LCF$_{\text{LEX}}$

B.1 Léxico

```
; definição do léxico do LCFlex
;
; exemplo:
; definir que a etiqueta cc é uma conjunção do tipo coordenativo,
; atribuindo-lhe a etiqueta *tag_cc*

(:morph cc
 :syntax ((cat conj) (type coord) (tag *tag_cc*))
 :semantics nil)

(:morph cd
 :syntax ((cat cardinal) (tag *tag_cd*))
 :semantics nil)

(:morph dt
 :syntax ((cat det) (tag *tag_dt*))
 :semantics nil)

(:morph ex
 :syntax ((cat there) (tag *tag_ex*))
 :semantics nil)

(:morph fw
 :syntax ((cat foreign) (type word) (tag *tag_fw*))
 :semantics nil)

(:morph in
```

```
 :syntax ((cat prep) (tag *tag_in*))
 :semantics nil)

(:morph jj
 :syntax ((cat adj) (tag *tag_jj*))
 :semantics nil)

(:morph jjr
 :syntax ((cat adj) (type comp) (tag *tag_jjr*))
 :semantics nil)

(:morph jjs
 :syntax ((cat adj) (type sup) (tag *tag_jjs*))
 :semantics nil)

(:morph jjp
 :syntax ((cat adj) (type sup) (tag *tag_jjp*))
 :semantics nil)

(:morph ls
 :syntax ((cat marker) (tag *tag_ls*))
 :semantics nil)

(:morph md
 :syntax ((cat verb) (form finite) (tense aux) (tag *tag_md*))
 :semantics nil)

(:morph nn
 :syntax ((cat noun) (type common) (mode mass) (agr sing) (tag *tag_nn*))
 :semantics nil)

(:morph nns
 :syntax ((cat noun) (type common) (mode count) (agr plu) (tag *tag_nns*))
 :semantics nil)

(:morph nnp
 :syntax ((cat noun) (type proper) (agr sing) (tag *tag_nnp*))
 :semantics nil)

(:morph nnps
 :syntax ((cat noun) (type proper) (agr plu) (tag *tag_nnps*))
 :semantics nil)
```

```
(:morph pdt
 :syntax ((cat predet) (tag *tag_pdt*))
 :semantics nil)

(:morph prp
 :syntax ((cat pronoun) (type pers) (tag *tag_prp*))
 :semantics nil)

(:morph prp$
 :syntax ((cat pronoun) (type poss) (tag *tag_prp$*))
 :semantics nil)

(:morph rb
 :syntax ((cat adv) (tag *tag_rb*))
 :semantics nil)

(:morph rbr
 :syntax ((cat adv) (type comp) (tag *tag_rbr*))
 :semantics nil)

(:morph rbs
 :syntax ((cat adv) (type sup) (tag *tag_rbs*))
 :semantics nil)

(:morph rp
 :syntax ((cat particle) (tag *tag_rp*))
 :semantics nil)

(:morph to
 :syntax ((cat prep) (tag *tag_to*))
 :semantics nil)

(:morph uh
 :syntax ((cat interj) (tag *tag_uh*))
 :semantics nil)

(:morph vb
 :syntax ((cat verb) (form baseform) (tense pres) (tag *tag_vb*))
 :semantics nil)

(:morph vbd
```

```
 :syntax ((cat verb) (form finite) (tense past) (tag *tag_vbd*))
 :semantics nil)

(:morph vbg
 :syntax ((cat verb) (form gerund) (tense pres) (tag *tag_vbg*))
 :semantics nil)

(:morph vbn
 :syntax ((cat verb) (form finite) (tense past) (tag *tag_vbn*))
 :semantics nil)

(:morph vbp
 :syntax ((cat verb) (form finite) (tense pres) (tag *tag_vbp*))
 :semantics nil)

(:morph vbz
 :syntax ((cat verb) (form finite) (tense pres) (tag *tag_vbz*))
 :semantics nil)

(:morph wdt
 :syntax ((cat wh) (type det) (tag *tag_wdt*))
 :semantics nil)

(:morph wp
 :syntax ((cat wh) (type pronoun) (tag *tag_wp*))
 :semantics nil)

(:morph wp$
 :syntax ((cat wh) (type poss) (tag *tag_wp$*))
 :semantics nil)

(:morph wrb
 :syntax ((cat wh) (type adv) (tag *tag_wrb*))
 :semantics nil)
```

B.2 Gramática

```
; definição da gramática do LCFlex
;
; exemplo 1:
; definir que a frase começa com a regra <s>
```

```
(<start> <--> (<s>)
  ((x0 = x1)))

; exemplo 2:
; definir que <s>, uma frase, é composta por um <np>,
; sintagma nominal (noun phrase), seguido de um <vp>,
; sintagma verbal (verb phrase).
; Neste caso o tipo de frase é tell.

(<s> <--> (<np> <vp>)
  (((x0 vp) = x2)
   ((x0 np) = x1)
   ((x0 stype) = tell)))

(<s>  <--> (<np> <vp> <ap>)
  (((x0 modifier) = x3)
   ((x0 vp) = x2)
   ((x0 np) = x1)
   ((x0 stype) = tell)))

(<s> <--> (<vp_ask> <np>)
  (((x0 np) = x2)
   ((x0 vp) = x1)
   ((x0 stype) = ask_if)))

(<s> <--> (<verb> <np> <vp> <np>)
  (((x1 tense) = aux)
   ((x0 object) = x4)
   ((x0 vp) = x3)
   ((x0 np) = x2)
   ((x0 aux) = (x1 tag))
   ((x0 stype) = ask_if)))

(<s> <--> (<verb> <np> <ap>)
  (((x0 modifier) = x3)
   ((x0 np) = x2)
   ((x0 vp verb) = (x1 tag))
   ((x0 stype) = ask_if)))

(<s> <--> (<verb> <np>)
  (((x1 form) = baseform)
```

```
   ((x0 np) = x2)
   ((x0 vp verb) = (x1 tag))
   ((x0 stype) = achieve)))

(<s> <--> (<verb> <pp>)
  (((x1 form) = baseform)
   ((x0 pp) = x2)
   ((x0 vp verb) = (x1 tag))
   ((x0 stype) = achieve)))

(<s> <--> (<verb> <np> <np>)
  (((x1 form) = baseform)
   ((x0 object) = x3)
   ((x0 np) = x2)
   ((x0 vp verb) = (x1 tag))
   ((x0 stype) = achieve)))

(<s> <--> (<verb> <ap>)
  (((x1 form) = baseform)
   ((x0 modifier) = x2)
   ((x0 vp verb) = (x1 tag))
   ((x0 stype) = achieve)))

(<s> <--> (<wh> <vp> <np>)
  (((x0 np) = x3)
   ((x0 vp) = x2)
   ((x0 prep) = (x1 tag))
   ((x0 stype) = ask)))

(<s> <--> (<wh> <verb> <np> <vp>)
  (((x2 tense) = aux)
   ((x0 vp) = x4)
   ((x0 np) = x3)
   ((x0 aux) = (x2 tag))
   ((x0 prep) = (x1 tag))
   ((x0 stype) = ask)))

(<vp> <--> (<verb> <np>)
  (((x0 np) = x2)
   ((x0 verb) = (x1 tag))))

(<vp> <--> (<verb> <pp>)
```

```
  (((x0 pp) = x2)
   ((x0 verb) = (x1 tag))))

(<vp> <--> (<verb>)
  (((x0 verb) = (x1 tag))))

(<vp_ask> <--> (<verb> <np>)
  (((x1 form) = finite)
   ((x0 np) = x2)
   ((x0 verb) = (x1 tag))))

(<vp_ask> <--> (<verb> <pp>)
  (((x1 form) = finite)
   ((x0 pp) = x2)
   ((x0 verb) = (x1 tag))))

(<vp_ask> <--> (<verb>)
  (((x1 form) = finite)
   ((x0 verb) = (x1 tag))))

(<ap> <--> (<adj>)
  (((x0 adj) = (x1 tag))))

(<ap> <--> (<adv> <adj>)
  (((x0 adv) = (x2 tag))
   ((x0 adj) = (x1 tag))))

(<pp> <--> (<prep> <np>)
  (((x0 np) = x2)
   ((x0 prep) = (x1 tag))))

(<pp> <--> (<prep> <noun>)
  (((x2 type) = common)
   ((x0 cnoun) = (x2 tag))
   ((x0 prep) = (x1 tag))))

(<np> <--> (<np> <pp>)
  (((x0 pp) = x2)
   ((x0 np) = x1)))

(<np> <--> (<det> <noun>)
  (((x2 type) = common)
```

```
   ((x0 cnoun) = (x2 tag))
   ((x0 det) = (x1 tag))))

(<np> <--> (<pronoun> <noun>)
  (((x1 type) = poss)
   ((x2 type) = common)
   ((x0 cnoun) = (x2 tag))
   ((x0 pronoun) = (x1 tag))))

(<np> <--> (<pronoun>)
  (((x1 type) = pers)
   ((x0 pronoun) = (x1 tag))))

(<np> <--> (<noun>)
  (((x1 type) = proper)
   ((x0 pnoun) = (x1 tag))))

(<np> <--> (<noun> <noun>)
  (((x1 type) = common)
   ((x2 type) = proper)
   ((x0 pnoun) = (x2 tag))
   ((x0 cnoun) = (x1 tag))))

(<np> <--> (<cardinal> <noun>)
  (((x2 type) = common)
   ((x2 mode) = count)
   ((x0 cnoun) = (x2 tag))
   ((x0 number) = (x1 tag))))
```