Desenvolvimentos recentes na interface humano-robô baseada em linguagem falada do robô Carl

Liliana Ferreira[1], Pedro Soares[1], Mário Rodrigues, Luís Seabra Lopes e António Teixeira

Resumo **- O desenvolvimento de sistemas robóticos úteis, com hardware e inteligência desenvolvida para aplicações específicas, isto é, o desenvolvimento de robôs capazes de aceitar instruções em termos de conceitos familiares ao utilizador, é ainda um desafio. Assim, para que surjam robôs que em vez de serem programados da forma clássica sejam capazes de aceitar instruções descritas em termos familiares para o utilizador humano, é essencial que sejam desenvolvidas interfaces de linguagem natural, pois esta é considerada a única interface aceitável para uma máquina que visa ter um elevado grau de interactividade com o Homem. Apresenta-se neste artigo a evolução recente dos módulos de geração de linguagem natural, reconhecimento de voz, e compreensão de linguagem do robô Carl. O novo módulo de reconhecimento de voz e parte do módulo de compreensão de linguagem natural já foram integrados. As restantes experiências devem ser interpretadas como exploratórias, tendo como objectivo o desenvolvimento de uma interface de linguagem natural robusta.**

Abstract **- The development of robots capable of accepting instructions in terms of familiar concepts to the user is still a challenge. For these robots to emerge it's essential the development of natural language interfaces, since this is regarded as the only acceptable interface for a machine which intents to have a high level of interaction with Humans. In this paper we present the recent evolution of several parts of the spoken language interface of the robot Carl: speech recognition, natural language understanding and natural language generation. The new speech recognition module and part of the natural language understanding were already integrated in Carl. The other experiments reported must be regarded as exploratory, aiming the development of a robust spoken language interface.**

I. INTRODUÇÃO

A. O Projecto Carl

O projecto CARL – "Communication, Action, Reasoning and Learning in robotics" – começou em Julho de 1999 como um projecto da FCT e é agora continuado como uma área de investigação transversal no Instituto de Engenharia Electrónica e Telemática de Aveiro (IEETA). O projecto CARL visa o desenvolvimento de um robô capaz de entender, através de um interface amigável, instruções expressas em termos de conceitos familiares ao utilizador humano.

Carl é o nome do robô do projecto CARL. O Carl, apresentado na Figura 1, é um protótipo de um robô inteligente, desenvolvido tendo em vista tarefas tais como servir comida numa recepção [1] ou servir como recepcionista numa organização [2].

Actualmente, o Carl utiliza a plataforma móvel PIONEER 2-DX da ActivMedia Robotics, com duas rodas motrizes e uma direccional. Esta inclui um computador baseado no processador Pentium com o sistema operativo Linux instalado, 16 sonares repartidos em dois conjuntos (um conjunto à frente e outro atrás), pára-choques com sensores de colisão, uma placa de som e uma câmara Sony EVI D31 *pan-tilt-zoom*.

Por cima desta plataforma, foi montada uma estrutura em fibra de vidro que aumenta a altura do Carl para aproximadamente 110 cm (ver Figura 1). Esta estrutura em fibra de vidro transporta um microfone direccional Voice Tracker da Acoustic Magic, bem como uma coluna de som. Na sua posição normal, de pé próximo do robô, a boca de um adulto encontra-se à distância de 1 m do microfone. Esta distância é suficiente para permitir o reconhecimento de voz num ambiente silencioso. Esta foi realmente a principal razão para a adição da estrutura de fibra. De forma a obter-se uma navegação robusta, foi adicionado à estrutura de vidro um conjunto de 10 sensores de infravermelhos para a detecção de obstáculos. Recentemente, foi também adicionado um computador portátil para suportar a interacção táctil, bem como a introdução de texto e a apresentação de uma cara artificial animada computacionalmente em simultâneo com o processo de síntese de voz [3-6].

[1] Uma parte do trabalho apresentado neste artigo foi desenvolvido no âmbito do estágio final da licenciatura em Matemática Aplicada à Tecnologia da Faculdade de Ciências da Universidade do Porto. Os nossos agradecimentos às Prof. Ana Paula Rocha e Teresa Mendonça.

Figura 1 - Fotografia do Carl, o robô do projecto CARL, na sua recente participação no "International Cleaning Robots Contest" em Lausanne, Suiça, Outubro 2002

O sistema de execução e controlo do Carl baseia-se num conjunto de processos executados sobre o sistema operativo Linux. Para o reconhecimento de voz existem dois processos e para síntese outro processo. Mais processos encarregam-se do processamento de outros tipos de informação vinda do exterior através dos sonares, dos sensores infravermelhos, da visão e dos sensores de colisão. Estes processos foram desenvolvidos em linguagem de programação C.

O esquema de integração dos vários domínios do Carl é representado pela figura seguinte:

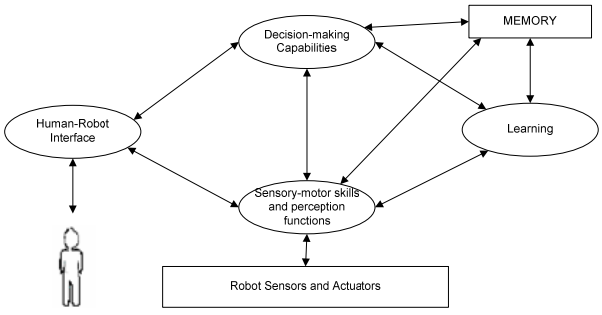

Figura 2 - Esquema de integração dos vários domínios do Carl.

B. Descrição da Interface de Linguagem natural.

Uma interface tem por função permitir a comunicação entre o Homem e a máquina. Como a principal forma de comunicação do Homem é o discurso falado, as interfaces de voz utilizando linguagem natural são as mais intuitivas para os humanos.

Pode dividir-se a arquitectura de uma aplicação por voz nos seguinte blocos principais: Reconhecimento de Voz, Compreensão de Linguagem Natural, Gestão de Diálogo, Geração de Linguagem Natural e Síntese de Voz [7].

O bloco de Reconhecimento converte o sinal acústico (sinal de voz) numa sequência de fones que formam cada palavra e posteriormente a frase. Para uma interpretação adequada da frase reconhecida é necessária a extracção da estrutura sintáctica e, posteriormente, do significado desta. Para tal existe o módulo de Processamento de Linguagem Natural. Este módulo estrutura a frase reconhecida de modo que esta seja interpretável pelo Gestor de Dialogo. A actividade do Gestor de Dialogo consiste em interpretar a mensagem do utilizador e reagir em conformidade. Para isso, o Gestor de frases, após receber a mensagem que o robô quer transmitir ao utilizador, constrói uma frase que a represente. Como tal mensagem deve ser transmitida por voz, o bloco de Síntese gera um sinal acústico a partir desse texto. Note-se que este sinal acústico tem de ser perceptível pelo ser humano.

Neste artigo apresentam-se os últimos melhoramentos efectuados nos módulos de Reconhecimento de Fala (secção II), Compreensão de Linguagem Natural (secção III) e Geração de Linguagem Natural (secção IV).

II. RECONHECIMENTO DE FALA

Quando alguém imagina a capacidade de falar com um computador, a primeira imagem que surge é, usualmente, o reconhecimento de voz, isto é, a conversão de um sinal acústico para uma sequência de palavras. O bloco de reconhecimento é crítico pelo facto de estar sujeito a um conjunto relevante de condicionantes externas, como é o caso do ruído ambiente, do qual são exemplo as conversas paralelas à frase a ser reconhecida e o próprio ruído dos motores do robô. As mudanças repentinas do contexto pelo orador e a qualidade do microfone em utilização também são factores relevantes para a qualidade do reconhecimento.

A. Versão anterior

O módulo de reconhecimento de voz implementado inicialmente no Carl (1999/2000) foi baseado no grapHvite da Entropic. Posteriormente foi utilizado o ViaVoice da IBM para Linux, em modo de gramática [8][9]. A utilização de gramáticas acarreta várias desvantagens: limita as frases possíveis, o não reconhecimento de uma palavra implica o não

reconhecimento de toda a frase, etc. Houve, assim, a necessidade de evolução deste módulo de forma a possibilitar o reconhecimento de mais frases, bem como a melhoria da taxa de reconhecimento e a possibilidade de treino/adaptação dos modelos acústicos.

B. IBM Viavoice em modo ditado

A vantagem da implementação de um reconhecedor em modo de ditado passa pelo facto de não se impor limitação sobre as palavras a serem reconhecidas. No entanto, tal implica a ocorrência de uma elevada percentagem de erros no reconhecimento. Deste modo foi necessária a activação de alguns mecanismos de correcção de erros disponibilizados pelo ViaVoice, tais como a apresentação das diversas alternativas às palavras reconhecidas pelo sistema e a apresentação do grau de confiança que este estabelecia em cada uma das anteriores. Assim, os resultados obtidos consistiam num conjunto de palavras e de valores numéricos, representativos das opções de reconhecimento e do grau de confiança destas, respectivamente.

Foi então necessária a utilização de uma ferramenta de construção, combinação, optimização e procura em máquinas de estado finitas com transições pesadas, desenvolvida pelos Laboratórios da AT&T e denominada Finite-State Machine (FSM) Toolkit [10]. No caso particular do reconhecimento de voz, cada um dos estados da máquina representa uma palavra e o peso da transição entre estados é obtido através do grau de confiança fornecido para cada palavra. Deste modo, e após a utilização desta ferramenta, obteve-se a combinação pesada das diversas alternativas de reconhecimento, sob a forma de uma lista das n melhores frases, isto é, as n frases com menor custo total associado.

No entanto, os resultados obtidos com esta ferramenta não tinham, na maior parte das vezes, significado linguístico. Assim, e para evitar que tal acontecesse foi instalada e analisada uma ferramenta para a produção e experimentação de modelos de linguagem baseados em estatísticas, desenvolvida pelo Speech Technology and Research Laboratory da SRI International e denominada SRILM [11]. Esta ferramenta cria um modelo de linguagem *n-gram* a partir de dados de treino e aplica as estatísticas obtidas a conjuntos de teste de modo a obter probabilidades, perplexidades, minimização de erros, entre outros resultados. Infelizmente, durante a tentativa de aplicar esta ferramenta aos resultados obtidos com o *FSM Toolkit* surgiu uma dificuldade: os programas que mais interessavam para a obtenção dos resultados desejados necessitavam de um ficheiro contendo informações sobre as classificações acústicas/linguísticas das n melhores frases reconhecidas que não era fornecido pelo ViaVoice.

Desta forma, foi descontinuada a utilização do ViaVoice, quer pelos motivos apresentados (não suporta n-grams), quer pelo interesse no desenvolvimento de um novo módulo.

C. Nova versão : Utilização do Nuance 8

O novo sistema de reconhecimento de voz eleito foi o da ***NUANCE*** na sua versão 8.0 [12].

O processo de reconhecimento de voz da Nuance baseia-se numa arquitectura cliente/servidor. O cliente do reconhecimento é responsável pela aquisição e pelo pré-processamento da entrada áudio, enquanto que o servidor de reconhecimento é o responsável pela transformação da onda sonora em texto, isto é, pelo reconhecimento de voz propriamente dito. Este último recebe como entrada o sinal de voz e utiliza três componentes para realizar o reconhecimento: os modelos acústicos que são fornecidos pelo sistema da Nuance e que são usados no reconhecimento de fonemas, os ficheiros de dicionários que contêm as descrições das pronunciações fonéticas das palavras definidas na gramática e que estão também incluídos no sistema, e por último as gramáticas de reconhecimento que definem o conjunto de frases que pode ser reconhecido. Uma vez que as palavras que constituem cada frase são construídas a partir de modelos de fonemas, é importante que o processamento acústico seja bastante eficiente. O sistema da Nuance utiliza modelos de Markov não observáveis como modelos acústicos na associação entre a onda sonora e a respectiva sequência de fonemas [13].

Uma outra característica importante do sistema de reconhecimento de voz da Nuance é o facto de permitir a utilização de modelos de linguagem para restringir aquilo que o utilizador pode dizer, permitindo, assim, que este se expresse de uma forma natural. Foi criado um modelo *bigram* [13] baseado num conjunto de frases representativo das que são passíveis de serem proferidas no contexto do projecto CARL. É ainda relevante mencionar que este reconhecedor suporta a criação de listas de n alternativas à frase reconhecida bem como a definição do grau de confiança a partir do qual todas as frases são rejeitadas. Deste modo, foi definido que eram apresentadas, no máximo cinco alternativas e que toda a frase a que esteja associado um grau de confiança inferior a 45% fosse rejeitada. Estes valores foram escolhidos de modo a que frases mal reconhecidas devido a interferências não fossem aceites mas que também não provocassem uma rejeição à mínima interferência.

D. Avaliação de desempenho

Foram realizados dois testes semelhantes ao reconhecedor construído através do sistema de

reconhecimento da Nuance cujos resultados são apresentados de seguida. A diferença entres os testes consiste no facto de um ter sido desenvolvido num ambiente silencioso (um laboratório de investigação com vários computadores a funcionar), enquanto que o segundo tenta simular o ambiente real de uma demonstração do robô através da existência de ruído de fundo (semelhante ao ambiente silencioso, onde a diferença reside na presença de várias pessoas falando num tom consideravelmente elevado).

A Tabela 1 apresenta os resultados do teste realizado em ambiente com ruído. Os números que estão entre parêntesis à frente das percentagens correspondem ao respectivo número de palavras.

Os testes não foram exaustivos por duas ordens de razões: primeiro, a habitual falta de tempo; segundo, interessa-nos essencialmente o desempenho do sistema na sua globalidade, não nos interessa concentrar-nos em optimizar apenas um dos módulos. Os resultados foram satisfatórios, permitindo a substituição do ViaVoice pelo novo módulo de Reconhecimento de Voz da Nuance.

	Alternativa 1		Alternativa 2		Alternativa 5	
Nº total de frases apresentadas	67		63		33	
Frases correctas	47,76%	(32)	7,94%	(5)	0%	(0)
Nº total de palavras	323		308		185	
Palavras bem reconhecidas	257		210		134	
Palavras substituídas (S)	13%	(42)	24,03%	(74)	19,46%	(36)
Palavras Inseridas (I)	1,55%	(5)	2,92%	(9)	5,41%	(10)
Palavras Removidas (R)	7,43%	(24)	7,79%	(24)	8,11%	(15)
Palavras correctas	79,57%		68,18%		72,43%	
Taxa de erro de palavras (WER)	20,43%		31,82%		27,57%	
Percentagem Total ((T-S-I-R)/T)	78,02%		65,26%		67,03%	

Tabela 1: Resultados dos teste de avaliação do reconhecedor da Nuance num ambiente ruidoso.

Verificou-se que o erro mais frequente, quer no teste com ruído quer no teste sem ruído, corresponde à substituição de palavras. Este tipo de erro reflecte o número de palavras mal reconhecidas, ou seja, a quantidade de palavras cujo reconhecedor não conseguiu identificar correctamente. Em ambos os testes este valor é reduzido quando comparado com a percentagem total de palavras bem reconhecidas o que revela um bom desempenho do reconhecedor.

A substituição mais frequentemente corresponde à substituição da palavra *is* pela palavra *you*. A justificação para esta substituição está na frequente má pronunciação por parte dos portugueses da palavra *is*. Este problema poderia ser minorado através de treino dos modelos. No entanto, note-se que ao treinar os modelos para a pronúncia portuguesa está-se a "destreiná-los" para as restantes pronúncias, o que poderia ter consequências não desejadas. Outra palavra frequentemente substituída é a palavra *the* pela palavra *a* e vice-versa. Esta substituição, apesar de originar uma frase mal reconhecida, não influencia o sentido desta, fazendo com que se torne de menor relevância.

A causa da inserção de palavras já é diferente: esta pode ser forçada quer por restrições impostas pelo modelo de linguagem em utilização, quer pela existência de ruído na altura em que o orador diz a palavra a ser reconhecida. Por exemplo, no seguinte caso em que o orador diz: *Where are you* e o robot reconhece: *Where are you <u>eating</u>*, a inserção da palavra *eating* na frase reconhecida foi provavelmente causada pela existência de ruído na altura da realização do teste. É de notar que as frases reconhecidas estão sujeitas ainda a um grau de confiança e por isso uma frase como a do exemplo só seria apresentada caso o grau de confiança das alternativas fosse menor do que o grau de confiança desta. Tal justifica o facto de a percentagem de inserções nos testes realizados ser reduzida.

É ainda importante analisar as possíveis causas das palavras removidas. Estas, à semelhança das palavras inseridas, podem ter origem em restrições impostas pelo modelo de linguagem. O problema pode também surgir no alinhamento das palavras, isto é, pode não existir uma correcta identificação de onde termina uma palavra e começa a outra. Isto deve-se ao facto de muitas vezes ao falar não se fazerem as devidas pausas entres palavras.

Finalmente, é relevante mencionar os nomes próprios em Português, como por exemplo *Teixeira*, que são frequentemente mal reconhecidos. Mais uma vez o problema poderia ser resolvido através do treino dos modelos acústicos. No entanto, pode acontecer que ao treinar o modelo com palavras deste tipo se esteja a alterar a pronúncia de palavras inglesas que contenham fonemas iguais aos destas. Uma melhor solução passaria por alargar o conjunto base de fonemas.

III. PROCESSAMENTO DE LINGUAGEM NATURAL - ANÁLISE SINTÁCTICA E SEMÂNTICA

A Compreensão de Linguagem Natural (NLU, do inglês *Natural Language Understanding*) é uma área que procura criar modelos computacionais de linguagem, que possibilitem a compreensão a computadores (ou outras máquinas) de informação em linguagem natural. A grande motivação é a das capacidades de processamento de linguagem natural revolucionarem a forma como os computadores são usados. Dado que na prática a grande maioria do conhecimento humano está guardado sob a forma de linguagem natural, os computadores que compreendessem linguagem natural teriam acesso a toda essa informação. Outra vantagem seria que, através de interfaces de linguagem natural, os sistemas computacionais complexos seriam acessíveis a todos, e ainda, esses sistemas seriam mais inteligentes e flexíveis do que é possível com a tecnologia computacional actual.

A compreensão de textos é efectuada combinando o conhecimento que temos de pequenas parcelas de texto. O objectivo da teoria linguística é mostrar como a combinação dessas parcelas constrói o texto integral. Esta tarefa é efectuada recorrendo a uma gramática. A linguística computacional tenta então implementar este processo de uma forma eficiente. Tradicionalmente divide-se esta tarefa em sintaxe e semântica. A sintaxe descreve como os elementos de uma frase podem ser combinados; a semântica como a interpretação é calculada. Na maioria das aplicações de compreensão de linguagem a gramática é separada dos componentes que efectuam o processamento de informação. A gramática é constituída por um léxico e por regras que sintáctica e semanticamente combinam palavras em frases. Assim, é através da aplicação do conhecimento existente na gramática que o texto é interpretado, sendo dissecado pelas regras que esta contenha.

Na versão anterior, a análise sintáctica era efectuada usando o CPK [14]. Isto era possível pela utilização de gramáticas no reconhecedor de fala. Com o novo reconhecedor mais flexível e livre chegam à análise sintáctica não frases. Para lidar com esta situação nova foi explorada uma nova abordagem, baseada em algoritmos de aprendizagem baseados em memória, que aprendem através de exemplos. Foram implementados módulos para: separação em frase/não frase do material que chega ao módulo de processamento de linguagem natural vindo do reconhecedor de palavras e para outras tarefas como a classificação em tipo de frase.

A. Problema Concreto

O módulo de compreensão em funcionamento no Carl é composto por um reconhecedor de voz (Nuance), que transforma o discurso humano dirigido ao robô em texto, e por uma aplicação (ALE – Atribute Logic Engine [15]) que, analisando as frases gramaticalmente correctas, retira a sua informação semântica. Contudo, o reconhecimento de voz efectuado pelo reconhecedor nem sempre é interpretável nem, na maioria das vezes, as frases que este retorna são correctas gramaticalmente. Aliás, este problema é crítico dentro do projecto dado um dos objectivos ser o do Carl poder funcionar e interagir com pessoas nos mais diversos ambientes. Logo, sendo muito difícil o reconhecimento em salas com algum ruído, será complicada a interacção com os humanos. Supondo que o ALE consegue retirar informação de todas as frases gramaticalmente correctas, torna-se necessário encontrar outra forma de interpretar frases incompletas ou mal reconhecidas, dado que cada frase que não seja analisada, será informação que não estará disponível para o Carl. Assim, o objectivo que é desejado alcançar é o do melhoramento da interpretação semântica, nomeadamente ao nível do processamento de frases estropiadas ou sem uma estrutura gramaticalmente correcta. Foi então pensada a criação de um módulo que trataria destes problemas.

O ALE tem um funcionamento baseado em gramáticas, sendo assim óbvia a razão pela qual não é possível interpretar frases gramaticalmente incorrectas. Sendo esta abordagem deficiente para os objectivos já expostos, foi necessária a procura de uma resolução com outras características. Assim, foi sugerido que se adoptasse pela Aprendizagem Baseada em Memória (Memory-Based Learning – MBL). A opção justifica-se pela possibilidade de utilização de um ficheiro de treino, que seria criado para servir as nossas necessidades. Assim, a cada frase retira-se informação que será analisada. Utilizou-se um programa de classificação (o TiMBL – Anexo A) para, com os exemplos do ficheiro de treino, podermos classificar a frase correctamente.
A tarefa de classificação é facilitada consoante a informação existente sobre cada frase. Assim, foram utilizadas também duas ferramentas com o objectivo de aumentar a informação sobre cada frase. A primeira destas ferramentas baseia-se na atribuição a cada palavra de uma etiqueta com a sua categoria gramatical (Part-of-Speech Tagging – PoS Tagging – Anexo B), e a segunda ferramenta baseia-se na segmentação sintáctica da frase, isto é, divide a frase em segmentos de palavras agrupados segundo o sintagma ao qual elas pertencem (Chunking).

B. Experiências / Resultados

1. Experiência 1 – Tipo de Frase

Para uma primeira adaptação ao *TiMBL* e ao *Brill's Tagger* [16], a primeira experiência que se realizou foi a tentativa de, automaticamente, determinar o tipo de frase que estava a ser analisada. A frase poderia ser de cinco tipos: declarativa, imperativa, interrogativa, pergunta de resposta afirmativa ou negativa, ou falsa (caso não pudesse ser considerada uma frase). Começou-se por criar

um ficheiro de treino com 20 frases, e testou-se 100 frases novas. O resultado foi surpreendente: 98% de eficácia. Apenas duas frases tinham sido mal classificadas. Depois de analisados os erros cometidos, concluiu-se que o erro foi devido a uma má classificação do *tagger*, que classificou *move* como um nome.
O ficheiro de treino consistia em vectores de informação com as *tags* (etiquetas) das palavras constituintes das frases. A cada frase corresponde um vector de informação (uma linha do ficheiro). Cada vector tinha 13 posições, a que se chamou características (designadas internacionalmente por *features*). As características eram preenchidas apenas com as *tags* das palavras. Assim, a característica 1 era preenchida com a *tag* da palavra 1, a característica 2 preenchida com a *tag* da palavra 2, e assim sucessivamente. As características que permanecessem vazias eram preenchidas com "-". A 13ª "característica" era a classificação a dar à frase. Um exemplo de linhas do ficheiro de treino encontra-se a seguir:

NNP,VBP,IN,NNP,-,-,-,-,-,-,-,-,0
DT,NN,VBZ,IN,NN,-,-,-,-,-,-,-,-,1

Os resultados obtidos permitiram a criação de boas perspectivas quanto à utilização destas ferramentas.

2. Experiência 2

O objectivo era preparar um programa para filtrar as frases que saem do reconhecedor de voz. Assim, de entre as hipóteses dadas pelo reconhecedor, teria de se escolher as frases gramaticalmente correctas. Para a resolução deste problema foram desenvolvidos dois programas. A diferença entre eles está na ferramenta utilizada para retirar a informação da frase. O primeiro programa usa apenas o *tagger* e o segundo usa o *tagger* e o *chunker*. O ficheiro de treino que o *TiMBL* usa é idêntico nos dois casos. A formação dos vectores de informação do ficheiro de treino é o seguinte: as duas primeiras características são a primeira palavra da frase e respectiva *tag*; a terceira e quarta posição são a segunda palavra da frase e respectiva *tag*, e assim sucessivamente; a 19ª posição é a classificação da frase. Com um conjunto de treino de 762 exemplos foram obtidas mais de 90% de decisões correctas. Aumentando o conjunto de treino, 3000 exemplos foram suficientes para aumentar a percentagem de decisões correctas para 95%.

3. Experiência 3

O módulo de Inteligência Artificial (IA), na altura, só aceitava informação do ALE. Assim, foi necessário efectuar a reconstrução de frases, quando nenhuma das alternativas que o reconhecedor apresentava era correcta. Então, foi acrescentado ao programa, que filtra as frases que o reconhecedor envia, uma função que tenta efectuar a reconstrução de frases. Não sendo obtida nenhuma frase gramaticalmente correcta pelo reconhecedor de voz, após a utilização da função encarregada da reconstrução é criado um novo conjunto de frases onde será procurada uma frase passível de ser analisada pelo ALE. Esta procura é efectuada pelo TiMBL.
Foi detectado um erro corrente nas frases reconhecidas: as palavras *in*, *is* e *you*, são frequentemente confundidas entre si. Assim, o primeiro passo da função de reconstrução é procurar estas palavras na frase, e criar três novas frases, substituindo uma palavra pelas outras. O passo seguinte é a criação de novas frases com combinações das palavras da frase inicial.

4. Experiência 4

Outro programa construído teve em vista a reprodução de uma experiência relatada por Walter Daelemans. Nesta experiência utilizou-se o TiMBL, um tagger e o LTChunk para a identificação do sujeito e objecto de uma frase. A importância deste programa reside em não ser necessário que uma frase esteja gramaticalmente correcta para se retirar a informação contida. Este programa foi testado, mas não existem, ainda, os ficheiros de treino necessários.

5. Experiência 5

O ALE tem uma função que permite recuperar frases incorrectas. Esta função tenta preencher os espaços em branco que a frase contenha, de forma a reestruturá-la correctamente. Assim, foi criado um programa que envia para o ALE um ficheiro com vários conjuntos de palavras, para que este reconstrua a frase. Os conjuntos são formados por combinações das palavras mais importantes de cada *chunk*. Ainda não é possível apresentar resultados desta experiência, dado que este programa ainda não foi implementado em conjunto com o ALE. Este facto deve-se ao grande esforço para a implementação em curso dos diversos módulos, o que ainda não permitiu a realização desta experiência.

Em paralelo com cada experiência foram ainda realizados programas que facilitam a criação de ficheiros de treino.

IV. Geração de Frases e Síntese de Voz

A Geração de Linguagem Natural (NLG, do inglês *Natural Language Generation*) é o processo através do qual o pensamento é transformado em linguagem [13][17]. Esta questão será, aqui, analisada de um ponto de vista computacional. Assim, a geração automatizada de linguagem natural é uma das áreas de estudo da Linguística Computacional e da Inteligência Artificial, dedicando-se a estudar e a simular a produção de discurso (escrito ou falado). Deste ponto de vista o gerador– que é o equivalente a uma pessoa com algo a dizer– é um programa de computador, e tem como objectivo a produção de um texto compreensível numa língua humana (no nosso caso o Inglês), a partir de uma forma de

representação de informação/conhecimento não linguística.

Hoje em dia, um sistema de NLG é visto como um módulo de uma aplicação, que tem como objectivo permitir uma transmissão agradável e eficiente de informação para o utilizador. A representação de informação contida numa aplicação computacional contém significado por si só, mas poderá não ser de fácil compreensão para uma pessoa normal. Assim, o trabalho de um gerador começa com a intenção inicial de comunicar, escolhendo o conteúdo do que será dito, seleccionando as palavras e a organização de ideias. Conclui criando uma frase gramaticalmente correcta ou formando textos em linguagem natural. Hoje em dia , as funções executadas por um gerador vão desde criar palavras ou frases para responder a uma questão, criação de legendas para diagramas, etc., até à criação de páginas inteiras de informações sobre determinados assuntos, dependendo da capacidade e dos objectivos da aplicação em que está inserido.

Apesar de não ser um assunto pacifico, existe dentro da comunidade científica internacional algum consenso sobre os processos que são necessários executar no módulo de geração. As tarefas encontram-se divididas em três processos principais como é explicitado na Figura 3.

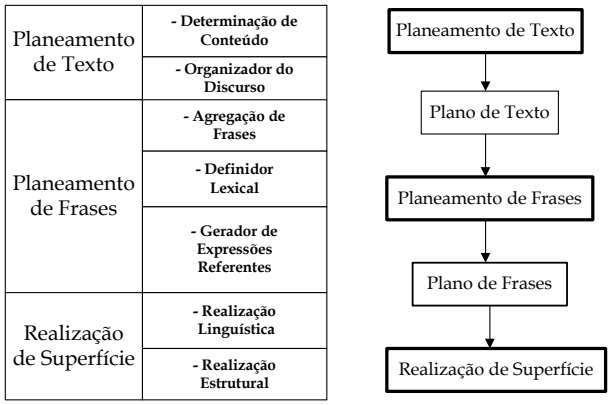

Figura 3 - Estrutura de um sistema de NLG e a forma de transmissão de informação entre os diferentes processos.

A. Qual o estado da arte?

O desenvolvimento de sistemas de NLG está intimamente ligado às aplicações em que serão inseridos, e às necessidades que estas apresentem. Assim, existem sistemas criados apenas para algumas das tarefas do processo de geração, sendo poucos os que ambicionam realizar todas as tarefas.

Os geradores de texto mais difundidos actualmente são os *Canned Text Systems*, ou seja, sistemas que têm frases pré definidas, e são usados na maioria do software: o sistema imprime simplesmente frases sem nenhuma mudança (mensagens de erro, avisos, etc.). Os *Template-Based Systems* são o nível seguinte de sofisticação. Utilizam estruturas fixas de texto em que apenas pequenas alterações são efectuadas. São exemplos deste tipo de sistemas as cartas com estrutura pré-definida em que apenas necessitamos de dar indicação sobre o destinatário, ou ainda, o gerador ANA (1983) que cria relatórios do mercado de conservas. Outro tipo de sistemas são os *Phrase-Based Systems*, que utilizam o que pode ser visto como estruturas generalizadas, quer ao nível da frase (em que as estruturas se assemelham a regras gramaticais estruturais de frases) ou ao nível do discurso (em que são frequentemente são denominados como planos de texto). Os sistemas deste tipo podem ser poderosos e robustos, mas são muito difíceis de construir quando é necessário uma grande variação de discurso, dado que todos os tipos de frases têm de estar previamente definidos. Um exemplo sofisticado é o *MUMBLE*, construído na universidade de *Massachusetts*, *Amherst*. Os *Feature-Based Systems* representam o estado da arte da geração. Neste tipo de sistemas cada alternativa possível é representada por uma única característica, por exemplo: uma frase é POSITIVA ou NEGATIVA, está no PRESENTE ou no PASSADO, etc. Os defeitos deste tipo de sistemas está na complexidade de informação que poderá ser necessária como entrada. Nesta classe de geradores encontram-se os sistemas mais avançados, como são exemplo o PENMAN/KPML e FUF (mais informação em [13]).

B. Que limitações existem?

É razoavelmente fácil construir um gerador simples para utilizar numa aplicação nossa, ou com mais alguma dificuldade adaptar um gerador existente. Contudo, não é possível ainda construir um gerador de frases generalizado ou um gerador de planos de texto com alguma capacidade de adaptação. Diversos problemas significativos permanecem ainda sem soluções suficientemente gerais, tais como a selecção do léxico, o planeamento da frase, a criação da estrutura do discurso, etc.. A dificuldade destas tarefas é crescente quando procuramos uma maior variedade e adaptação a diferentes circunstâncias. Só para dar um pequeno exemplo da dificuldade que pode surgir na escolha do léxico imaginemos que era preciso dar uma informação sobre o carro do João, as possibilidades de escolha são: o carro do João, o automóvel do João, o veículo do João, etc.. Todas essas possibilidades podem ser válidas consoante a situação, o nível de linguagem a ser utilizado, o contexto, etc.. Assim, é facilmente perceptível que esta é ainda uma área com bastante trabalho pela frente...

C. Problema Concreto

A arquitectura que foi apresentada anteriormente referia-se a um sistema de geração de linguagem natural em geral. Na adaptação de um módulo de NLG ao projecto Carl, vários processos não são necessários, dado que o nosso objectivo é a realização de uma conversa, na qual, em princípio, apenas será necessária a geração de frases.

Assim, o funcionamento do módulo de NLG no Carl será o seguinte:

- O módulo de Inteligência Artificial (IA) fornecerá ao gerador a ideia a transmitir. A ideia já estará dividida em blocos de informação que representam frases.
- O gerador criará uma ou mais frases em linguagem natural que consigam transmitir cada uma das ideias.
- O módulo de NLG fornecerá ao módulo de síntese as frases a serem convertidas em discurso.

Assim, o módulo de geração necessário ao projecto CARL apenas terá como objectivo efectuar a realização de superfície, tendo três componentes: a componente sintáctica, a componente morfológica e a componente ortográfica, sendo esta última de relevância diminuta.

D. Trabalho efectuado

Um dos objectivos que se queria que o Carl atingisse era o de responder a perguntas sobre o IEETA e sobre ele próprio, sendo também criada uma base de dados com a informação necessária para ele estar à altura destas solicitações. Assim, a construção do programa de ligação entre o módulo de IA e o ASTROGEN também era o responsável pelo acesso à informação da base de dados.
A base de dados foi construída em Prolog para facilitar a interacção com os outros programas. A informação que contém está armazenada sob a forma de predicados, existindo informação sobre os atributos de entidades (laboratórios, projectos, professores e o próprio Carl), sobre as relações entre as diversas entidades e sobre o mapa interno do IEETA. Também é definido o tipo de todas as entidades.

A Tabela 2 contém a informação que é disponibilizada pela base de dados para a geração de frases. Os nomes de investigadores encontram-se em minúsculas, dado que não o sintetizador não dá relevância a esse pormenor.
Os predicados principais são:

- **type**(entity, type)
- **relation**(relation,entity1,entity2)
- **attribute**(entity, attribute type, attribute)
- **cabinet**(entity, place)
- **place**(cabinet, localization)

Projectos	Coordenador	Investigadores	Tipo de Projecto	Relação com o IEETA
Projecto Carl (carl_proj)	seabra lopes	seabra Lopes antonio Teixeira	Projecto de Pesquisa (research_proj)	Unidade de Investigação (r_unit)
Laboratório de Sistemas de Informação de Telemática (lab_sit)	sousa pereira	sousa pereira marques silva manuel ramos manuel cunha	Laboratório de Pesquisa (research_lab)	Basic Research Group (b_r_group)
Laboratório de Processamento de Sinal (lab_ps)	paulo ferreira	paulo ferreira ana tome armando pinho francisco Vaz	Laboratório de Pesquisa (research_lab)	Basic Research Group (b_r_group)
Laboratorio de Sistemas Computacionais (lab_sc)	antonio ferrari	antonio ferrari antonio borges valery sklyarov	Laboratório de Pesquisa (research_lab)	Basic Research Group (b_r_group)
Laboratório de Sistemas (lab_se)	estima oliveira	estima oliveira alexandre mota ernesto martins jose fonseca	Laboratório de Pesquisa (research_lab)	Basic Research Group (b_r_group)
Projecto F. C. Portugal (fcportugal_proj)	nuno lau	nuno lau luis reis	Projecto de Pesquisa (research_proj)	Unidade de Investigação (r_unit)

Tabela 2: Informação da base de dados disponível para geração

Lista M	Frase Gerada
[responsible(carl_proj)]	Professor seabra is the coordinator.
[projects]	Projects are: carl project and f.c. Portugal.
[researchers]	researchers are: seabra lopes antonio teixeira.
[relation(researcher, carl_proj, X)]	Professor antonio teixeira is a researcher of the carl project.
[cabinet('antonio teixeira', X)]	Professor antonio teixeira is in the second floor in the cabinet 210.
[type('seabra lopes', X)]	Professor seabra lopes is a researcher.
[relation(researcher, carl_proj, Y), relation(researcher, carl_proj, X)]	professor antonio teixeira and professor seabra lopes are researchers of the carl project.
[cabinet('antonio teixeira', X), cabinet('seabra lopes', Y)]	professor antonio teixeira is in the second floor in the cabinet 210 and professor seabra lopes is in the first floor in the cabinet 111.
[type('seabra lopes', X), type('antonio teixeira', Y)]	Professor antonio teixeira and professor seabra lopes are researchers.
[relation(researcher, carl_proj, Y), cabinet(Y,X)]	professor seabra lopes is in the first floor in the cabinet 111 and professor seabra lopes is a researcher of the carl project.
[type('carl', X), relation(researcher, carl_proj, Y)]	carl is a robot and professor seabra lopes is a researcher of the carl project.

É com a utilização destes atributos que o módulo de IA especifica qual a informação a transmitir. O modo de chamar a geração de frases *gerar(M,L)*, onde M é a lista de informação a transmitir, e na variável L será recebida a frase que será sintetizada. A lista M poderá ser da forma:

- Combinação do predicados **relation**, **cabinet** e **type**.
- [researchers(labs or project]
- [labs]
- [projects]
- [responsible(lab or project)]

E. Resultados

Na Tabela 3 estão alguns resultados de frases geradas pelo módulo de NLG. Os resultados são quase todos relativos ao projecto CARL, contudo, informação idêntica, relativa ao outro projecto ou laboratórios, também é processada da mesma forma.

Este facto será um dos pontos que terá de ser corrigido.

F. Síntese de voz

A conversão da frase em onda sonora é actualmente efectuada pelo sistema de síntese de voz IBM Viavoice TTS.
Recentemente, foi efectuada um experiência exploratória com um domínio de síntese limitado [18][19], utilizando o Festival. Como o vocabulário utilizado pelo Carl para transmitir informações é limitado, é possível, com uma pequena quantidade de gravação e processamento, desenvolver uma nova voz com uma maior qualidade natural.

Tabela 3: Exemplos de frases geradas pelo módulo de NLG.

V. PARTICIPAÇÃO DO CARL NO ROBOT CONTEST .

O Carl participou recentemente na primeira edição do *International Cleaning Robots Contest* que se realizou em Lausanne na Suiça. Este evento foi integrado na *2002 IEEE/RSJ International Conference on Intelligent Robot Systems (IROS'2002)*, uma das mais importantes conferências na área da robótica inteligente, onde foi apresentada uma comunicação sobre o Carl.

O concurso incluiu três modalidades de competição: *floor cleaning*, *window cleaning* e *housekeeping ideas*. O Carl participou na terceira destas modalidades desempenhando o papel de mordomo, demonstrando elevadas capacidades de interacção humano-robô.

Em Lausanne, o Carl demonstrou as suas capacidades durante a recepção de abertura do IROS'2002, provocando grande curiosidade entre os participantes da conferência. O único rival existente na modalidade de *housekeeping* foi um robô que regava plantas. Ambos foram premiados.

Neste concurso o Carl usou a nova versão dos módulos de reconhecimento e compreensão de linguagem natural que identifica frases e não frases descritos neste artigo.

VI. DISCUSSÃO

A. Reconhecimento

O trabalho descrito tinha como objectivos principais a implementação de um novo sistema de reconhecimento de voz. Para tal, foram analisadas as capacidades quer do modo de ditado do reconhecedor em utilização, quer do sistema de reconhecimento de voz da Nuance. O primeiro reconhecedor avaliado permite concluir que o

desejo de que o utilizador se exprima de uma forma totalmente livre e natural é demasiado ambicioso. Isto porque quando não se definem quaisquer limitações sobre o que o reconhecedor pode apresentar, então os resultados tornam-se menos correctos.
Ao analisar os resultados obtidos a partir do sistema de reconhecimento de voz da Nuance observa-se que, apesar de ter sido imposta uma limitação na linguagem utilizada através de um modelo de linguagem, os utilizadores não necessitam de se restringir a um conjunto limitado de comandos, continuando a obter-se resultados satisfatórios. É de notar, no entanto, que ao impor um modelo linguístico deste tipo se restringe o vocabulário que pode ser reconhecido, isto é, impõem-se limitações na quantidade de palavras passíveis de serem reconhecidas.
Os problemas que surgiram com a utilização do Nuance estavam ligados, com uma relativa frequência, à substituição de palavras. Tal como foi dito, este tipo de problemas poderão no futuro ser combatidos e possivelmente minorados através do treino dos modelos acústicos de modo a adaptá-los às características dos utilizadores portugueses.

Assim, o futuro do desenvolvimento do módulo de reconhecimento de fala passa pela implementação da capacidade de adaptação dos modelos aos utilizadores e ao ambiente.

B. Processamento de Linguagem Natural

A introdução de novas técnicas de processamento de linguagem no projecto CARL, terá, a curto prazo, um papel importante a desempenhar. A grande vantagem desta abordagem é a liberdade que nos dá, impensável com a utilização de gramáticas. Com um melhoramento do módulo de IA, será possível aumentar o raio de acção actual no projecto.
Apesar de os resultados da **Experiência 2** (classificação em frase ou não frase) parecerem, à primeira vista, pouco favoráveis, é importante relembrar que o processo construção de ficheiros de treino é extremamente importante nesta tecnologia. Este factor é dificilmente ultrapassável, pois não temos, actualmente, no projecto, nenhum *corpus* disponível para a criação de ficheiros de testes. A não existência de um *corpus* também torna complicado a realização de testes para o cálculo do desempenho, pois será sempre uma tarefa algo empírica.

A possibilidade de utilizar programas que retiram informação de ficheiros de treino, permite também que possamos moldar o processamento aos nossos objectivos. A utilização do programa que reproduz a experiência de Daelemans permitirá, depois da criação dos ficheiros de treino, reconhecer o sujeito e o objecto de uma frase. Com a adaptação do módulo de IA, este programa poderá assumir as funções que o ALE tem.

Esta abordagem ganha força quando se torna necessária a reconstrução de frases, pois é uma tarefa extremamente complicada. Aliás, não foi encontrado nenhum artigo que focasse este assunto. Assim, se tivermos um programa que permita uma compreensão baseada apenas em algumas palavras da frase, será mais fácil contornar os erros, e retirar informação da frase.
Este trabalho já foi entretanto continuado sendo os resultados publicados em [4-6].

C. Geração de frases

O processamento de informação do módulo de NLG depende em grande parte do módulo que o comanda, no nosso caso o módulo de IA. A interacção entre estes dois módulos é fundamental para que a geração seja conseguida de uma forma capaz. Este foi um dos problemas que se encarou na realização desta tarefa. A quantidade de informação que o *Astrogen* necessita, para fazer a geração, é superior à que o módulo de IA disponibiliza. Este problema foi apenas minorado com o programa que liga os dois módulos. Apesar de ir buscar muita da informação que utiliza à base de dados, ainda carece alguma adaptação do módulo de IA.

Um outro problema, possivelmente mais grave, foi a dificuldade que existiu na adequação do *Astrogen* às nossas necessidades. Tornou-se complicado adaptá-lo à informação que era necessário que processasse, em grande parte devido à deficiente documentação disponível, e também devido aos erros que o programa tem. O *Astrogen* apenas está preparado para alterações ao léxico e à gramática que não alterem os esquemas de frases que traz de origem, o que o torna demasiado limitativo. Existiu uma necessidade de acrescentar novas palavras, e consequentemente, novas regras gramaticais, e, por consequência, o tratamento interno do *Astrogen* a esta nova informação foi deficiente.

Apesar dos problemas, o módulo de NLG criado é adaptável ao Carl, realizando a tarefa pretendida. A geração, não sendo a desejável, é a suficiente para os moldes de funcionamento do módulo de IA.

VII. REFERÊNCIAS

[1] Seabra Lopes, L. (2001). Carl, a learning robot, serving food at the AAAI reception. In: *Proc .of the AAAI Mobile Robot Competition and Exhibition Workshop*. Seattle, WA, USA

[2] Seabra Lopes, L. (2002). Carl: from situated activity to language level interaction and learning. In: *Proc. IEEE IROS*. pp. 890-896.

[3] Seabra Lopes, L., A. Teixeira, D. Gomes, C. Teixeira, J. Girão and N. Sénica (2003). A friendly high-level human-robot interface. In: Encontro Científico do 3º Festival Nacional de Robótica.

[4] Seabra Lopes, L., Teixeira, A., Rodrigues, M,. Gomes, D., Teixeira, C., Ferreira, L, Soares, P., Girão, J., Sénica, N. (2003). Towards a Personal Robot with Language Interface. In: Proc. 8th European Conference on Speech Communication na Technology (Eurospeech), Geneva, Suiça, Setembro.

[5] Seabra Lopes, L., Teixeira, A.J.S., Rodrigues, M., Gomes, D., Girão, J., Teixeira, C., Sénica, N., Ferreira, L., Soares, P. (2003). A Robot with Natural Interaction Capabilities. In: 9th IEEE Internacional Conference on Emerging Technologies and factory Automation (EFTA), Lisboa, Portugal, Setembro.

[6] Teixeira, A., Seabra Lopes, L., Ferreira, L., Soares, P., Rodrigues, M. (2003). Recent Developments on the Spoken Language Human-Robot Interface of the Robot Carl, Robótica 2003 – 3ª Festival Nacional de Robótica, Maio.

[7] Seabra Lopes, L. and A.J.S. Teixeira (2000b). Teaching behavioral and task knowledge to robots through spoken dialogues. In: My Dinner with R2D2: Proc. AAAI Spring Symposium on Natural Dialogues with Practical Robotic Devices.

[8] Seabra Lopes, L. and A. Teixeira (2000a). Human-robot interaction through spoken language dialogue. In: *Proc. IEEE/RSJ IROS.*

[9] Ferreira, N.M., Ferreira, N.F., Rodrigues, M., Teixeira, A., Vaz, F., Seabra Lopes, L. (2001). Interacção com um Robô através de Linguagem Natural Falada, Revista do departamento de Electrónica e Telecomunicações da Universidade de Aveiro, Vol. 3, N.º 3.

[10] http://www.research.att.com/sw/tools/fsm

[11] http://www.speech.sri.com/projects/srilm

[12] Nuance (n.d.). *Nuance Speech Recognition System, version 8.0.: Introduction to the Nuance System.* And the other manuals.

[13] Jurafsky, D. and J.H. Martin (2000). *Speech and Language Processing*. Prentice Hall

[14] Brendsted, T. (1999). The CPK NCP Suite for Spoken Language Understanding, In: Proc. Eurospeech .

[15] Carpenter, B. , Penn, G. (2001), ALE The Attribut Logic Engine User's Guide Version 3.2.1.

[16] Brill, E (2000). Part-of-Speech Tagging. In: *Handbook of Natural Language Processing* (Robert Dale, Hermann Moisi and Harold Somers, Eds.). Marcel Dekker.

[17] Becker, Tilman (1999). Natural Language Generation: some examples.

[18] Black, A. and K. Lenzo (2000). Limited domain synthesis. In: *Proc. ICSLP*. Beijing China.

[19] Pereira, P. Barros, J. (2002). Síntese de Voz para Aplicações Específicas, Relatório Final da Disciplina de Projecto, LEET, Dep. Electrónica e Telec, Universidade de Aveiro, 2002.

[20] Daelemans, W., J. Zavrel, K, van der Sloot and A. van den Bosh (2001). *TiMBL: Tilburg Memory Based Learner Reference Guide, v. 4.1*.

[21] http://www.ltq.ed.ac.uk/software/pos/

[22] Dalianis, H. (1999). A general validation tool by natural generation for the STEP/EXPRESS Standard.

[23] http://www.dsv.sv.se/~hercules/ASTROGEN/ASTROGEN.html

VIII. ANEXOS

A. TiMBL

O *TiMBL* (*Tilburg Memory-Based Learner*) [20] é um programa que tem implementados vários tipos de algoritmos de *Memory Based Learning* (MBL). Com a criação do grupo de pesquisa *ILK* (*Induction of Linguistic Knowledge*) em 1997, e com o uso crescente da tecnologia de MBL, tornou-se necessário a criação de uma ferramenta que englobasse os algoritmos mais utilizados. Assim, o *TiMBL* é o resultado da combinação de várias ideias vindas de implementações de MBL. Desde a sua criação o *TiMBL* tem-se vindo a afirmar como uma ferramenta importante no Processamento de Linguagem Natural, sendo um dos seus maiores impulsionadores *Walter Daelemans*.

B. PoS Tagging

A etiquetagem de cada palavra de um texto (ou frase) com a sua categoria sintáctica é um processo designado por *PoS Tagging*. A investigação na realização automática desta tarefa teve origem na década de 1950.

Aparentemente simples, esta tarefa é bastante complicada devido às ambiguidades que as palavras podem ter. Um exemplo disso é a palavra rio, que pode ser considerada um nome ou a utilização da primeira pessoa do pretérito perfeito do verbo rir conforme a frases onde esteja inserida. Assim, a tarefa de um *tagger* (que é o programa que efectua etiquetagem), também é a de contemplar o contexto em que a palavra está inserida, de forma a etiquetá-la correctamente. O *tagger* começa por analisar quais as etiquetas possíveis de cada palavra, para depois escolher a que melhor se adequa ao contexto [21].

Entre as tarefas para as quais a etiquetagem das palavras com sua categoria sintáctica é importante encontram-se a tradução automática e extracção automática de informação. Na tradução automática, a probabilidade de uma palavra na língua de origem ser correctamente traduzida, é altamente dependente da etiqueta que lhe é atribuída. Para a extracção de informação é usual recorrer ao *tagger* para melhor localizar o assunto pretendido.

Uma das principais técnicas de etiquetagem utilizada são os modelos de Markov não observáveis. Outras técnicas mais recentes são a Aprendizagem Baseada em Transformações (Transformation-Based Learning – TBL) e a utilização de estimadores de Máxima Entropia.

Alguns exemplos de etiquetagem de palavras encontram-se na Tabela 5, onde se pode ver a etiqueta correspondente a cada frase ligada a esta por uma "/".

Esta é a forma usual em que os *taggers* apresentam os resultados.

Frase Original	Frase Etiquetada
My name is Pedro.	My/**PRP$** name/**NN** is/**VBZ** Pedro/**NNP** ./**.**
Where is the robot?	Where/**WRB** is/**VBZ** the/**DT** robot/**NN** ?/**.**
I am in Aveiro.	I/**PRP** am/**VBP** in/**IN** Aveiro/**NNP** ./.

Tabela 4: Exemplos de frases em que as palavras estão etiquetadas com a categoria sintáctica.

C. ASTROGEN

O *ASTROGEN – Aggregated deep and Surface naTuRal language GENerator* – é um gerador de linguagem natural escrito em Prolog [22][23]. Efectua a geração de superfície e também a agregação de frases. O objectivo do seu desenvolvimento foi a conversão de descrições existentes sob a forma do protocolo *STEP/EXPRESS* para linguagem natural, de forma a permitir a compreensão destas descrições a utilizadores comuns. Estava inserido no projecto VOLVEX: *Validation Of Specifications by Natural Language Generation for VOLVO expressed in STEP/EXPRESS*. O *STEP – STandard for the Exchange of Product data* – é uma forma de representação de dados sobre trocas de produtos e os produtos em si, que foi criada para as normas de estandardização ISO 10303. O *EXPRESS* é a linguagem utilizada pelo *STEP*.

O *ASTROGEN* consiste, basicamente, em dois módulos o gerador profundo, que efectua a agregação, e o gerador de superfície. A geração de superfície é baseada em gramática. A estrutura do ASTROGEN encontra-se na Figura 4.

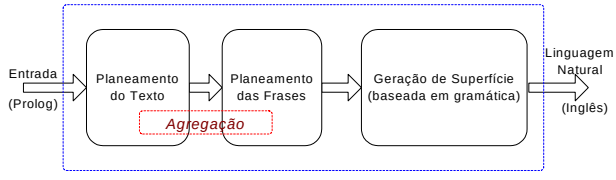

Figura 4: Arquitectura computacional do Astrogen.

O módulo de planeamento de texto é algo débil e quase inexistente, deixando para o módulo de planeamento de frases as funções de agregação. A informação do plano de texto que é transmitida pelo módulo de planeamento de texto necessita de ser, quase na totalidade, introduzida no *ASTROGEN*. O módulo de planeamento de frases consiste numa sequência de processos que levam a cabo a agregação. Cada um deste processos funciona independentemente dos outros e a sua interligação é similar a uma cadeia de montagem. Os processos são os seguintes:

- Agregação sintáctica
- Pronominalização.
- Delimitação de frases.

O resultado da utilização de cada um destes processos no texto em linguagem natural encontra-se exemplificado na Tabela 4.

Geração de Superfície …	John is a subscriber and John is idle and Mary is a subscriber and Mary is idle and Paul is busy.
… + Agregação Sintática …	John and Mary are subscribers and idle and Paul is busy.
… + Pronominalização …	John and mary are subscribers and they are idle and Paul is busy.
… + Delimitação de Frases	John and Mary are subscribers and they are idle. Paul is busy.

Tabela 5: Exemplo da utilização de cada um dos processos anteriores.

O módulo de geração de superfície apenas utiliza uma gramática para a geração. O procedimento baseia-se na correspondência entre as estruturas de informação da entrada com as estruturas de linguagem natural da gramática.