

Uma perspectiva técnica da Inteligência Artificial

Trabalho efectuado por PEDRO MANUEL CABRAL DA FONSECA

Aluno da turma C, para a cadeira de Filosofia da Linguagem

Introdução.....	3
Problemas específicos da IA	3
Inteligência e inteligência artificial	3
As coisas que um computador não pode fazer	4
Estrutura do programa.....	5
Representação do conhecimento	5
Métodos de pesquisa heurística de soluções	6
Tipos de pesquisa heurística	6
Pesquisa em extensão ou profundidade.....	6
Nível de imitação do processo humano de resolução de problemas	8
Redes neurais	8
Redes de Hopfield.....	9
Perceptrões	11
Identificação da linguagem oral	12
Sistemas de processamento de linguagem natural.....	13

NOTA: A abreviatura *AI* refere o livro *Artificial Intelligence*, de Elaine Rich e Kevin Knight, na 2ª edição de 1991. A abreviatura *IntAr*, refere o livro *Inteligência Artificial, guia para o programador*, de Piero Scaruffi (© de 1987), na tradução portuguesa da Editorial Presença, edição de 1990.

Introdução

O que permite distinguir um programa de Inteligência Artificial da panóplia de *software* actualmente disponível para computadores? A resposta mais geral é a de que um programa de IA deve manifestar inteligência. Esta resposta (em princípio correcta) é insatisfatória já que não nos diz o que é a inteligência¹ nem de que forma ela se deve manifestar. Em relação a este último ponto podemos encontrar na literatura sobre a IA (implícita ou explicitamente) pelo menos três critérios: os problemas que o programa consegue resolver, a estrutura interna do programa e o modo como o processo de resolução de problemas imita o do homem.

Problemas específicos da IA

Inteligência e inteligência artificial

Uma máquina capaz de calcular a raiz cúbica de um número de 10 dígitos em menos de 1 segundo é mais ou menos inteligente (e em que grau) que outra que passa 1 minuto a tentar compreender (i.e. pôr em contexto assimilável) uma frase óbvia com quatro palavras? Posta desta maneira a questão é irrespondível. A noção de inteligência que possuímos é demasiado nebulosa para proporcionar uma resposta consensual a esta questão.

No entanto, tradicionalmente, considera-se que certo tipo de aplicações não pertencem ao domínio da IA. Um programa de IA deve ser capaz de imitar a actuação humana sendo independente deste. Um programa de contabilidade tem um grau de independência limitado visto que não tem meios para considerar se as operações que faz sobre os números introduzidos são adequadas ao fim estipulado. Isto é, as funções utilizadas pelo programa não lhe dizem o que a contabilidade pretende ou se é normal que um estabelecimento comercial com duas máquinas de tirar café gaste apenas 10.000\$00 de electricidade por mês.² Para que o computador pudesse realizar tarefas deste tipo é necessário introduzir-lhe uma quantidade bastante grande de informação. A dificuldade da tarefa aliada à quase execrável qualidade dos computadores actuais (pelo menos relativamente aos fins em vista) e a ausência de grandes investimentos na área de investigação em IA levaram a que pouco se tivesse evoluído na construção deste tipo de sistemas.³ É, por isso, justificável que se diga que a área de IA se restringe à construção de sistemas que

¹ As várias noções clássicas de inteligência, forjadas a partir da observação da actividade de pessoas consideradas 'inteligentes', não se adaptam bem à classificação dos sistemas artificiais. Parte da razão é o facto de o significado da palavra 'inteligência' ser um pouco difuso e dependa mais da *imagem* global (coisas que vão desde o olhar aos valores éticos) que temos da pessoa a que o aplicamos do que de um critério específico que nos permita distinguir as pessoas inteligentes das não inteligentes. Por outro lado a 'inteligência' tem sido, desde a Idade Média, utilizada como propriedade que distingue homens de animais. (Este mito tem sido em grande parte o responsável pela caça a muitas espécies [por exemplo a dos gorilas] cuja capacidade em comunicar, abstrair, rir, tem sido sucessivamente comprovada.) Nesta perspectiva, criar máquinas inteligentes seria quase como assumir o papel de Deus, seria criar máquinas vivas. Não vou falar neste trabalho das várias noções de inteligência. Sigo a perspectiva mais vulgar no nosso século que é a de considerar que a inteligência não faz parte do mundo supra-sensível que dá ao homem a possibilidade de uma conduta ética mas que ela é apenas uma propriedade de alguns sistemas 'naturais' que o homem também partilha.

² Isto, é claro, não é impossível, mas é altamente improvável porque indicia a presença de um contador de mais de 15 amperes (aluguer mais elevado) e a saída de muitas 'bicas' (o que aumenta o consumo de electricidade).

³ As dificuldades derivam sobretudo da dificuldade em extrair e introduzir a informação e os 'motores de inferência' necessários à criação de programas de IA. Isto fez com que a maior parte dos programas fosse construído segundo moldes clássicos pelo que a necessidade de recorrer ao processamento paralelo foi em grande parte encoberta (com excepção da área dos chamados 'super-computadores'). É plausível que a vulgarização dos CD-ROM venha a tornar possível a introdução de programas cada vez mais próximos dos de IA e que isso vulgarize a utilização das *motherboard* de processamento paralelo para PCs (já

compreendam frases e se retire do seu domínio programas que resolvam equações de qualquer nível de complexidade. Isto, é claro, não nos permite dizer que um desses programas seja mais inteligente que o outro. Não podemos, portanto definir a IA como uma disciplina que tenta produzir sistemas inteligentes. Em vez disso podemos dizer apenas que ela tenta resolver problemas que, por enquanto, ainda não são bem resolvidos pelas máquinas. (Nesta perspectiva é difícil dizer se um jogo de xadrez é (actualmente) um programa de IA ou não. E essa resposta será ainda mais difícil de dar se o computador que está a ser desenvolvido pela IBM (capaz de considerar um bilião de jogadas por segundo) conseguir vencer Kasparov no torneio mundial de Xadrez.)

Pode dividir-se a actividade da IA em quatro grandes áreas:

- Execução de tarefas formais, – que são as que exigem (comparativamente) menos quantidade de conhecimento – e que incluem jogos e programas que actuam (tentando descobrir regras ou inconsistências) no domínio da lógica, matemática, etc.
- Emulação de peritos, que inclui a engenharia, medicina, química ou economia.
- Execução de tarefas triviais, – esta é a actividade que envolve o maior grau de conhecimento – o que inclui o processamento da linguagem natural, percepção, compreensão do senso comum e reconhecimento da linguagem falada.
- Outras, como o controlo de robots e a construção de sistemas que desenvolvam sistemas de IA.

As coisas que um computador não pode fazer

As dificuldades levantadas à introdução e manipulação (pelo *software* e *hardware*) de enormes quantidades de conhecimento fizeram com que muitos projectos que inicialmente se consideravam fáceis de alcançar (como a tradução automática, a compreensão do discurso ou o raciocínio de senso comum (*common sense reasoning*)) se revelassem como os objectivos mais difíceis a que a IA poderia aspirar. Estas dificuldades levaram muitos pensadores a restringir o domínio que os computadores seriam capazes de tratar. Esse domínio pode então ser identificado com o núcleo da inteligência humana ou daquilo que distingue o homem das máquinas⁴. Estas restrições parecem-nos erradas (muitas foram desmentidas pela experiência) ou prematuras (não há ainda suficientes dados disponíveis para considerar o que um computador por princípio pode ou não fazer).⁵

No domínio conceptual creio que os computadores podem emular com bastante perfeição os seres humanos. As dificuldades que se têm vindo a encontrar não dependem de qualquer característica intrínseca ao computador mas da perícia necessária aos seus criadores humanos para o porem a funcionar. As dificuldades que nós temos vindo a encontrar para os pôr a ouvir, a ler, a falar e interactuar com o mundo são as mesmas que teríamos se tivéssemos de ensinar um recém nascido que não aprendesse por ostensão. Temos não só que ensinar ao computador que uma cadeira é algo em que as pessoas se sentam mas também o que é que quer dizer ‘algo’, pessoa, quais as limitações de uma definição dada através da função, quais as propriedades de um objecto físico e as características que as pessoas têm. Uma criança de três anos já

disponíveis no mercado há algum tempo). Parece-nos que esta vulgarização é indispensável ao desenvolvimento rápido da IA.

⁴ Para uma lista de várias posições adoptadas ver Turing, *Can a Machine Think?*.

⁵ Isto, é claro, aplica-se tanto aos apoiantes da IA como aos cépticos. Assim parece-me correcta a crítica da visão segundo a qual um sistema de processamento simbólico tem consciência. Turing por exemplo de fende que se um computador pudesse responder como se fosse humano não teríamos qualquer razão para dizer que ele não tinha consciência. porque, se o fizéssemos, diz Turing, nenhum critério (ou, talvez, nenhum critério empírico) poderia impedir a queda numa posição solipsista. Neste aspecto penso que a crítica de Searle é interessante, na medida em que parece apontar para uma dimensão ‘semântica’ que tem de existir como suporte de um sistema meramente simbólico ou ‘sintáctico’, dimensão essa que a IA simbólica não prova poder ser criado a partir de um sistema simbólico. É evidente que a crítica de Searle também não prova que os computadores (ou, na verdade, alguém com a excepção de "mim") tenham ou não tenham consciência. Mas mostra que há algo (a mente ou ‘semântica’) que não se segue das *descrições dadas* dos sistemas de IA e, portanto, que o argumento de Turing é insuficiente. (Para uma posição contrária ver Ned Block, “Mental Pictures and Cognitive Science”, in *Mind and Cognition*)

aprendeu respostas parciais para todas estas questões, no entanto, utilizamos esse ‘conhecimento’ sem saber qual ele é e, na maior parte dos casos, sem saber sequer que o utilizamos.⁶ A dificuldade que se coloca então ao ensino dos computadores não deriva apenas do volume de informação que tem de se introduzir mas também da dificuldade de determinar os dados necessários à execução de uma determinada tarefa.

Outra dificuldade corresponde ao facto de os computadores clássicos (segundo o modelo de von Newman) executarem as instruções de forma sequencial. Embora certos problemas (como os da aritmética elementar) não possam ser resolvidos de forma substancialmente mais rápida pelo processamento paralelo (visto que cada operação depende do valor obtido da operação anterior) para outros isso é praticamente indispensável. É relativamente simples fazer com que um programa de IA tradicional possa utilizar múltiplos processadores de forma a que dez CPUs resolvam um problema dez vezes mais rapidamente do que se utilizássemos apenas um processador. Com o recente interesse da indústria na criação de computadores chamados ‘massivamente paralelos’ com mais (mas, no futuro, muitíssimo mais) de 200 processadores parece legítimo esperar que a IA dê um salto quantitativo tal como, pelo aumento do conhecimento disponível, poderá dar um salto qualitativo.

Por outro lado, os computadores são capazes de aprender e extrair informação automaticamente. Essa capacidade está no entanto, fortemente dependente da quantidade de conhecimento que um computador tem à partida. Um computador capaz de aprender a partir de uma enciclopédia tem já de possuir uma quantidade de informação avassaladora. No entanto, depois de possuir uma base de dados suficientemente possante (o que pode levar ainda algumas décadas a conseguir) o computador terá eventualmente armazenado suficiente informação para aprender a partir de enciclopédias, conversas, etc. Assim, o maior volume de informação associado a *hardware* mais apropriado e com ‘motores de inferência’ mais aperfeiçoados irá certamente contribuir para melhorar quer o número quer a exactidão com que certos problemas até agora reservados quase exclusivamente ao homem podem ser resolvidos por computadores.

Estrutura do programa

Dissemos já implicitamente que um programa de IA não tem a mesma estrutura que um programa convencional. As características de um programa de IA estão fortemente dependentes do tipo de problema que ele deve resolver. No entanto podemos considerar que qualquer programa simbólico de IA utiliza duas componentes fundamentais: a representação de conhecimento e um método de pesquisa heurística (*heuristic search*).

Representação do conhecimento

A primeira função do conhecimento é descrever o problema a resolver. Por exemplo para criar um programa que jogue xadrez precisamos de criar uma descrição do tabuleiro de xadrez das peças e do seu valor e das jogadas válidas. Depois atribuímos a cada disposição possível das peças no tabuleiro um determinado valor baseado na diferença entre o valor das nossas peças e o das do adversário. Ao computador resta apenas calcular as posições possíveis com base na disposição actual das peças no tabuleiro e nas jogadas válidas. Depois disso escolhe a jogada que conduz à disposição de maior valor.

Para além de nunca tentar fazer xeque (normalmente, nos livros sobre xadrez, não é dado valor ao rei) este programa já seria capaz de jogar xadrez. No entanto seria extraordinariamente lento e ineficaz. Considerar para cada jogada possível todas as jogadas possíveis do adversário torna-se uma tarefa praticamente impossível para projectar mais de três lances. Por outro lado este programa é incapaz de

⁶ Na verdade, é forçoso que só tenhamos consciência de uma ínfima parte dos processos que utilizamos para compreender o que nos rodeia. Nisto, os computadores são ainda piores do que nós visto que, num programa não implantado de forma paralela, só uma decisão pode ser tomada de cada vez. Os homens, pelo contrário, podem ter consciência de uma situação envolvendo vários aspectos (o que pode ser explicado pela sua capacidade de processar informação em paralelo) pelo que a sua análise da situação é geralmente, em situações que envolvem um grande número de factores, mais rápida do que a dos computadores. Para o computador o ‘inconsciente’ é a quase totalidade do seu programa (e dados).

considerar factores essenciais para fazer um bom jogo de xadrez como o controlo do centro do tabuleiro, a protecção do rei ou a colocação das peças em jogo.

Uma das maneiras de fazer com que o computador jogue melhor seria, por exemplo, atribuir valores pelo número de casas centrais que se domina (isto é, que se pode atacar se invadidas pelo adversário). Isto levaria a que o computador tentasse dominar o centro. Outra solução seria limitar o número de jogadas consideradas informando o computador das jogadas possivelmente mais interessantes e fazendo-o considerar essas em primeiro lugar. Estas duas soluções aprofundam o grau de conhecimento que o computador possui do problema no entanto têm objectivos distintos. Em geral, quando se tenta aumentar a qualidade com que um programa é capaz de analisar uma situação aumenta-se o tempo de análise necessário para resolver o problema. Por outro lado, a introdução de conhecimento para acelerar a procura de possíveis soluções pode fazer com que possíveis soluções sejam ignoradas.

Métodos de pesquisa heurística de soluções

Como já vimos, uma característica de qualquer programa de IA é o de ter de solucionar problemas cuja implantação em computador tem sido até agora difícil. Parte da razão é a de que não há um caminho claro a seguir para solucionar o problema. Por exemplo, quando queremos fazer uma soma de dois números específicos sabemos que o computador encontrará uma solução num certo número de passos. Poucos problemas de IA permitem este tipo de previsão. Em vez disso o computador tem uma representação de uma situação que a apresenta como um conjunto (finito ou infinito) de possibilidades. chamaremos a esse conjunto espaço de pesquisa. O problema consiste em passar de um estado inicial a um estado final através de um conjunto limitado de regras. Mas o caminho que tem de ser percorrido do estado inicial até ao estado final não é dado. O computador tem de o descobrir por meio de tentativas sucessivas, voltando atrás sempre que um caminho se revele infrutífero. Num jogo de xadrez este processo manifesta-se quando o programa tenta encontrar uma boa jogada. No caso da compreensão da linguagem natural manifesta-se quando tenta encontrar uma representação simbólica adequada à frase expressa em linguagem natural.

Tipos de pesquisa heurística

Um dos maiores problemas que se põem no desenho de um programa de IA é exactamente o de saber que tipo de pesquisa se deve utilizar para solucionar um certo tipo de problemas. O tipo de pesquisa utilizado difere grandemente de programa para programa e depende muito do problema a ser tratado. Por outro lado tem também de se controlar o caminho escolhido para efectuar a pesquisa. Da boa orientação do programa depende o tempo despendido na procura, a qualidade da solução alcançada e mesmo a possibilidade de se encontrar qualquer tipo de solução.

Uma das condições iniciais para se proceder a uma busca é definir em que condições uma dada regra é aplicável. O computador utilizará assim em cada estágio da busca de uma solução apenas as regras adequadas a esse ponto da situação. Por exemplo, o roque, no xadrez, só é possível quando nenhuma casa entre o rei e a torre está ocupada ou ameaçada. Enquanto tal não se verificar o computador não considerará o roque como uma jogada possível. Portanto, a pesquisa que fará das suas jogadas futuras não o incluirão. No caso do xadrez a aplicabilidade das regras é bastante simples. Noutros casos é mais complexa e chega mesmo a constituir o núcleo do programa.

Por exemplo (o exemplo é tirado do livro AI, p.188) o programa ELIZA, quando confrontado com a frase 'Eu sei que toda a gente se ri de mim' tem várias regras possíveis de aplicação. Por exemplo, o programa poderia responder 'Fala-me mais sobre isso' o que é aplicado a qualquer frase introduzida, ou, devido à frase utilizar a palavra chave 'eu', poderia responder 'Tu dizes saber que toda a gente se ri de ti' ou ainda, utilizando a palavra chave 'toda a gente' diria 'Em quem estás a pensar especificamente?'. A resposta escolhida por ELIZA seria, neste caso, a última visto que a expressão 'toda a gente' (*everybody*) foi considerada mais significativa do que 'eu'. Como se vê, o sucesso deste programa depende sobretudo da sua capacidade de encontrar as regras adequadas a uma dada situação. Encontrar a regra mais adequada depende apenas da definição dada previamente do valor semântico de cada palavra chave e não da consideração do valor dos resultados alcançados por cada uma das regras possíveis (este programa não tem capacidade para avaliar a qualidade das suas respostas). Este exemplo mostra também que muitas vezes o conhecimento que um sistema tem do objecto em estudo muitas vezes está imbuído no seu método de escolha de regras a utilizar.

Pesquisa em extensão ou profundidade

A partir do momento em que se conhecem as regras que se podem aplicar em cada estágio da resolução do problema, podemos delinear o tipo de procura que o programa deve utilizar para descobrir a solução para um determinado problema. Há, basicamente, dois grandes tipos de pesquisa que um computador pode efectuar no espaço de pesquisa. Ele pode efectuar uma pesquisa em extensão (*breadth-first search*) ou em profundidade (*depth-first search*). A pesquisa em extensão considera todas as regras aplicáveis ao estado inicial do problema (por exemplo, na primeira jogada de um jogo de xadrez, as regras que definem os movimentos possíveis dos peões e dos cavalos). Depois aplica cada regra ao estado inicial e memoriza o estado daí resultante. Para cada um dos estados resultantes volta a aplicar todas as regras possíveis e a memorizar os resultados da sua aplicação. O que quer dizer que para um problema que tenha em cada passo apenas seis possibilidades, o número de posições a ser encontrado irá ser igual a 6 elevado ao número de passos efectuado.

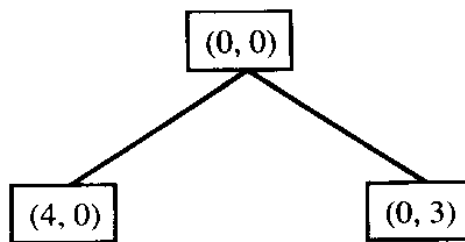


Figure 2.5: One Level of a Breadth-First Search Tree

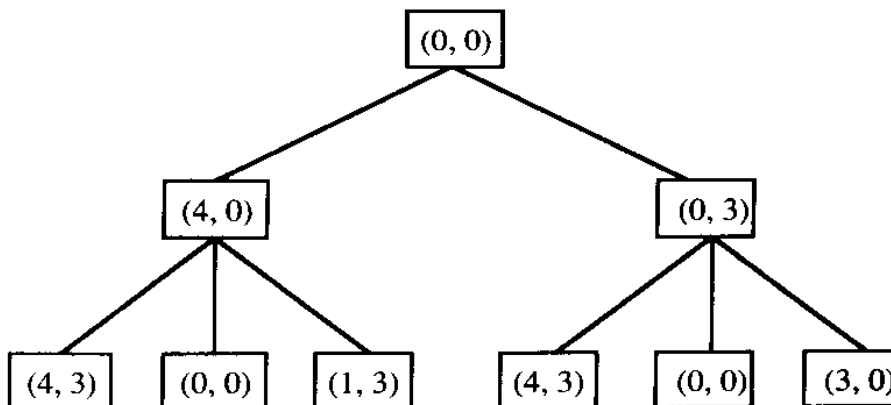


Figure 2.6: Two Levels of a Breadth-First Search Tree

Representação gráfica de uma pesquisa em extensão
(retirado de IA, p.38)

Por outro lado, a pesquisa em profundidade analisa apenas um ramo de cada uma das possibilidades até atingir o estado final ou até que não haja mais regras aplicáveis. Como é evidente tem de se juntar algumas condições perante as quais um caminho é abandonado. Por exemplo, se ele repetir um estado já dado previamente (se o programa não terminasse a pesquisa desse caminho ingressaria num looping rumo à eternidade).

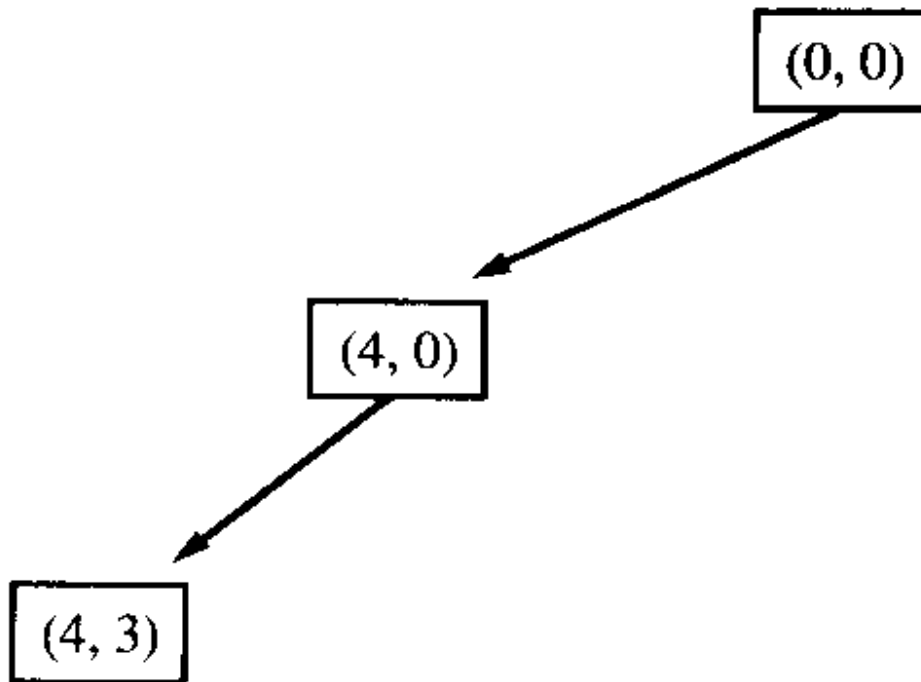


Figure 2.7: A Depth-First Search Tree

Representação gráfica de uma pesquisa em profundidade
(retirado de *AI*, p.39)

A maior parte dos programas de IA utiliza uma conjugação destes dois métodos. Para isso é necessário recorrer a um volume maior de conhecimento de modo a saber reconhecer os bons caminhos dos maus e a gerar uma pesquisa que assegure que não menosprezamos uma solução eficaz ou que não é desperdiçado mais que o tempo necessário em caminhos sem saída. Abordaremos com mais pormenor os métodos de representação de conhecimento e de pesquisa heurística no decorrer do nosso trabalho.

Nível de imitação do processo humano de resolução de problemas

Os computadores, com o desenvolvimento de linguagens de programação de alto nível, tornaram-se um importante instrumento de trabalho para psicólogos, sociólogos tal como para todos os investigadores do comportamento de sistemas complexos. No entanto, no domínio da disciplina de IA a imitação do comportamento humano só é justificada se ela ajudar a aperfeiçoar o programa quer porque a análise dos métodos humanos permite uma melhor compreensão do problema e métodos de solução, quer porque ela é indispensável a compreensão de certas situações (por exemplo, no caso da emulação de raciocínio de senso comum). De modo geral os programas que têm métodos de resolução de problemas semelhantes ao que as pessoas utilizam são preferíveis porque permite explicar mais facilmente aos seus utilizadores os pressupostos e raciocínios utilizados pelo computador para justificar a conclusão a que chegou permitindo assim que estes analisem a validade das conclusões atingidas. Os programas que têm como objectivo imitar o modo como os homens realizam determinadas tarefas estão já fora do domínio da IA. No entanto eles são certamente úteis para a compreensão do modo como o homem realiza certas tarefas.

Redes neurais

Outro nível em que diversos investigadores tentaram simular o homem foi ao nível do *hardware*. A tentativa de simular neurónios foi uma das primeiras linhas de investigação adoptadas pela IA e precocemente abandonadas. A investigação de redes neurais só foi retomada há relativamente pouco tempo

e, embora as opiniões ainda se dividam quanto à sua utilidade futura, a verdade é que já conseguiram alguns sucesso no campo do tratamento de imagens e reconhecimento do discurso.

A principal desvantagem das redes neurais é que não nos ajudam a compreender o modo como um problema pode ser resolvido por homens. Paradoxalmente, as redes neurais são sem dúvida os modelos funcionais mais parecidos aos cérebros de que dispomos neste momento. De facto, as redes neurais possuem algumas características que as tornam aparentemente parecidas ao modo com funciona o nosso cérebro. Em primeiro lugar podem sofrer estragos de alguma importância e continuarem a trabalhar sem perda assinalável de qualidade. Nas redes neurais quer a memória quer os procedimentos utilizados na transformação de padrões encontram-se espalhados por toda a rede e podem ser reconstruídos a partir apenas de um fragmento. Visto que o processamento de informação é feito por toda a rede em simultâneo o processamento da informação é feito em paralelo e não em série, pelo que certos problemas como o reconhecimento de imagens podem ser feitos muito rapidamente. Por outro lado as redes neurais aprendem por ostensão. Normalmente estas redes não são programadas para fazer reconhecimento de palavras ou o que quer que seja (além disso seria demasiado difícil – na prática é impossível – estar a calcular os valores que as várias ligações entre as células deveria ter para se atingir o resultado pretendido). O que se faz é apresentar a uma rede uma série de dados (vozes, imagens, etc) diferentes mas com certas características parecidas. Depois dizemos à rede, para cada dado emitido, se ela falhou ou não na sua classificação. No caso de ter falhado, a própria rede faz os ajustes necessários de modo a aproximar-se mais de uma identificação correcta. Depois de um determinado período de aprendizagem esperamos que ela retorne o mesmo sinal sempre que depare com um dado que possua certas características essenciais. É claro que não precisamos de saber como é que uma determinada rede reconhece os seus objectos (na verdade duas redes inicialmente iguais que aprendam a reconhecer as mesmas situações podem funcionar de maneira muito diferente).

Embora o funcionamento das redes neurais nos possa ajudar a ter uma ideia sobre o modo como o nosso cérebro funciona dificilmente nos pode ajudar a resolver problemas conceptuais ou sequer a perceber como é que nós resolvemos problemas num plano conceptual.⁷ No entanto, é muito plausível que as redes neurais venham a ter um papel muito mais importante no futuro. Em primeiro lugar porque proporcionam um objecto de estudo apaixonante, por outro lado elas parecem ser mais apropriadas para certas tarefas (sobretudo na área da percepção) do que as aproximações simbólicas em último lugar porque o desenvolvimento da tecnologia dos computadores de processamento paralelo favorece a simulação rápida de redes neurais.⁸ Visto que não voltarei a falar delas no decurso deste trabalho, farei uma breve descrição do modo de funcionamento dos sistemas mais simples.

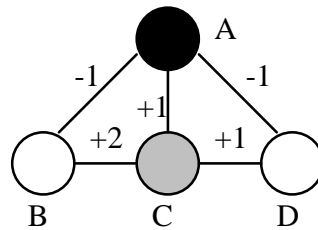
Redes de Hopfield

Uma rede de Hopfield é constituída por um conjunto de elementos ligados entre si. Estas ligações têm um determinado peso expresso por um número que pode assumir valores negativos ou positivos. Se duas unidades estiverem ligadas por uma ligação de peso negativo têm tendência a desactivarem-se mutuamente,

⁷ Seria mais ou menos como tentar explicar como é que funciona uma colónia de formigas baseados nas suas propriedades atómicas ou químicas. Mesmo que o conseguíssemos, a imagem daí resultante não nos permitiria ver as complexas relações que se estabelecem entre os vários membros da colónia e a importância de certos rituais para a sua sobrevivência. Para conseguirmos isso é necessário compreender que causas diferentes podem gerar os mesmos efeitos. Mesmo que duas formigas tenham uma estrutura atómica radicalmente diferente se as compararmos átomo por átomo, elas responderão de maneira praticamente idêntica face a situações idênticas. São as características abstraídas deste comportamento idealmente uniforme que permite explicar a sobrevivência da colónia.

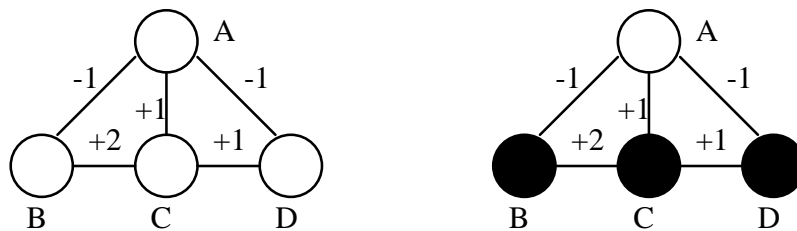
⁸ As redes neurais são, normalmente, simuladas em computadores tradicionais. Não tenho conhecimento de nenhuma rede que tenha tido existência real. Parece ser crível que uma rede neural real seja muitas vezes mais rápida e eficiente do que uma rede simulada. Uma notícia recente numa conhecida revista de PCs dizia que em menos de um ano serão comercializadas as primeiras redes neurais – implantadas em *hardware* – para PC. Estas redes serão comercializadas pela Intel e terão um custo inicial de duas ou três centenas de contos. No entanto não sei exactamente como é que o *hardware* implanta a rede neural.

uma ligação positiva tende a ligá-las. No entanto só as unidades activas podem exercer influência sobre as unidades adjacentes. Assim, numa rede de Hopfield em que todas as unidades estão desactivadas não é possível haver uma alteração do estado inicial. No entanto, se uma unidade for activada ela irá activar todas as unidades adjacentes com as quais tenha uma ligação positiva. Por exemplo, se, na seguinte figura, activarmos a unidade A isso irá provocar a activação da unidade C:



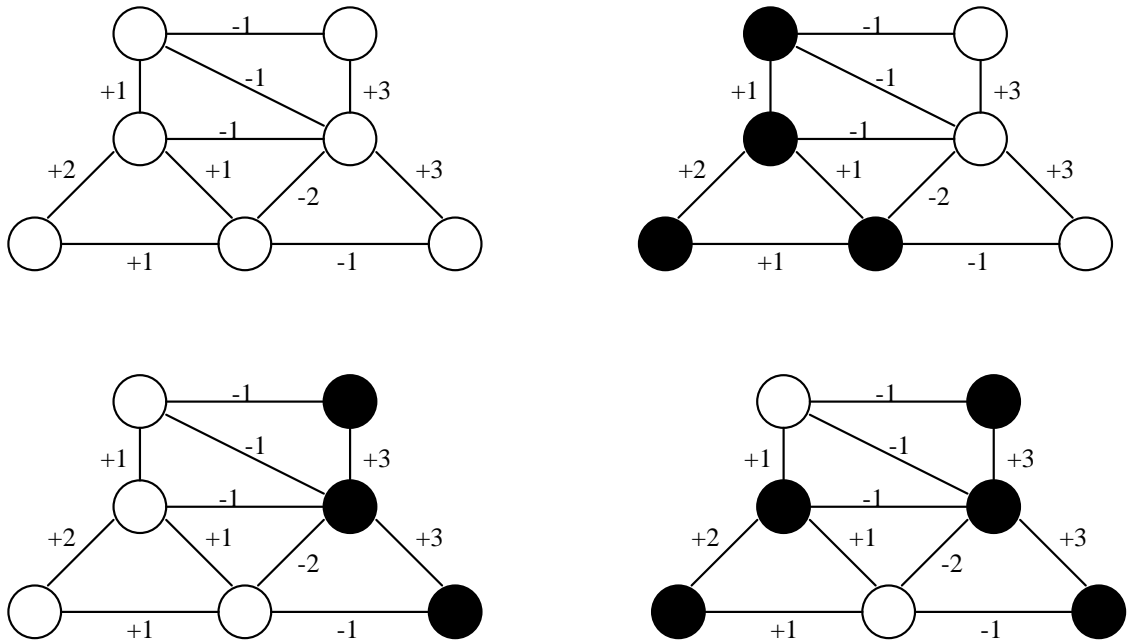
Activação da unidade C pela unidade A

Por outro lado, A exerce uma pressão inibitória sobre as unidades B e D. Essa força impede C (agora activa) de activar D uma vez que a soma das ligações AD (-1) e CD (+1) não é um número positivo. No entanto C activará B uma vez que a soma das ligações AB (-1) e BC (+2) resulta num número positivo. A configuração resultante irá então desactivar A uma vez que essa unidade terá de ter em conta o peso das relações BA e CA (só contam os valores das células activas). A desactivação de A faria com que a unidade D deixasse de receber qualquer influência inibidora pelo que seria activada por C. Assim, a activação de A produziria, ao fim de 4 passos uma configuração em que A é a única unidade desactivada. Na verdade, este sistema só pode ter dois estados pelo que a activação de qualquer unidade leva necessariamente ao estado final descrito.



Os dois estados possíveis do sistema descrito

Em sistemas mais complexos é possível haver várias configurações dependendo das unidades inicialmente activadas. Uma característica importante destes sistemas é que, se considerarmos que um determinado padrão corresponde a uma característica de um objecto basta apenas fornecermos à rede uma parte do padrão para que ela reconstitua o resto a partir de ligações predefinidas. Mas as mesmas ligações podem conter ainda muitos outros padrões diferentes. A dificuldade está em atribuir às ligações os pesos certos de modo a que possam recordar os padrões certos.

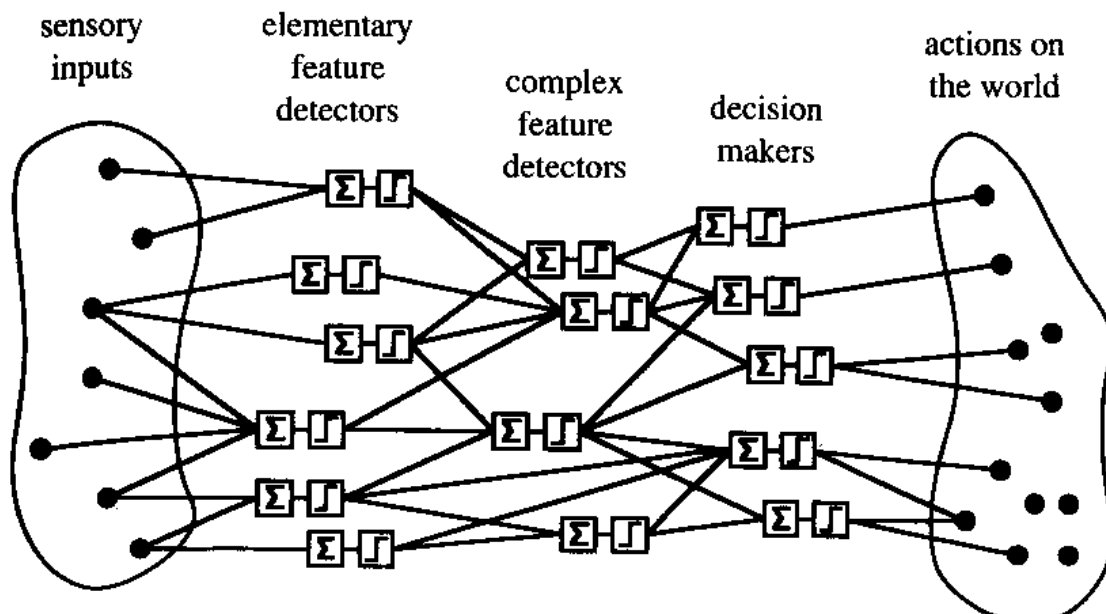


Os quatro estados estáveis de uma rede de Hopfield particular
(exemplo de AI, p.491)

Perceptrões

A ideia de que certo tipo de redes pode realizar actividades cognitivas apareceu já em 1943,⁹ muito antes do aparecimento do projecto de IA ter aparecido (Minsky e McCarthy foram os seus fundadores oficiais numa conferência realizada em 1959). Rosenblatt foi um dos mais famosos membros da corrente conexionista pela invenção do conceito de perceptrão. O funcionamento de um perceptrão é algo análogo ao de um neurónio. Possui várias entradas que podem assumir vários valores. Cada entrada tem um peso específico. O valor introduzido em cada entrada de um perceptrão é multiplicado pelo peso de cada entrada respectiva. Os valores, que podem ser negativos ou positivos, são então comparados com um certo número limite. Se o somatório dos números for superior a esse limite o perceptrão emite um sinal, se for inferior não emite. A vantagem dos perceptrões é que podem aprender facilmente a processar informação sem necessidade de serem previamente programados para realizarem funções. Em vez disso os perceptrões são treinados para realizarem certas funções. Rosenblatt julgava ser possível construir um sistema formado por várias camadas de perceptrões *previamente* ensinados a efectuar certas tarefas. Um sistema completo de perceptrões poderia assim simular um comportamento humano tal como se mostra na seguinte figura:

⁹ De acordo com o Heinz R. Pagels, *Os Sonhos da Razão*, Gradiva, 1988 (1ª ed. portuguesa de 1990), p.147.



(Imagem retirada de *AI*, p.495)

No entanto, as investigações sobre as propriedades das redes neurais tornaram-se praticamente inexistentes na década de 70 devido principalmente à análise que Minsky e Papert fazem das propriedades dos perceptrões no seu livro de 1969, *Perceptrons*. De facto, os dois autores conseguiram mostrar que os perceptrões eram incapazes de efectuar funções simples como a função *ou* exclusiva (equivalente à negação da função \Rightarrow). Tais funções só seriam possíveis em sistemas mais complexos, onde as saídas dos perceptrões se juntassem às entradas de outros formando assim um só sistema funcional. Mas, julgou-se na altura que um sistema composto de múltiplas camadas não poderia ser treinado da mesma forma que poderíamos treinar um perceptrão individual. Só depois de mais de uma década se descobriram formas de ensinar redes constituídas por várias camadas de unidades.

Este obstáculo, se fosse inultrapassável, teria proibido completamente a utilização de redes neurais. De facto, relembrando o que já dissemos, é na prática impossível programar uma rede calculando os valores que cada unidade deve ter com as restantes para alcançar o fim previsto. Esse cálculo é incrivelmente complexo e não temos qualquer maneira de o fazer. Se as redes devem funcionar então é absolutamente indispensável que aprendam por elas mesmas. O algoritmo através do qual redes com várias camadas podem aprender é demasiado complexo para ser aqui tratado. No entanto o processo é análogo ao realizado por um perceptrão.¹⁰

Identificação da linguagem oral

O problema de transformar em símbolos linguagem falada tem sido abordado com mais ou menos sucesso por vários programas. o livro *IA* cita seis sistemas de reconhecimento do discurso o mais recente dos quais, o SPHINX de 1988, pode reconhecer cerca de mil palavras, em tempo real e com discurso contínuo.

¹⁰ Por exemplo (exemplo adaptado de *IA*, p.495), imaginemos que queremos que um perceptrão responda com um sinal positivo de cada vez que se introduzir um valor positivo na entrada E_1 tudo o que temos de fazer é fazer com que o peso dado a E_1 seja maior quando o perceptrão não consegue reconhecer o sinal, ou diminuir o valor das entradas E_n quando haja disparos quando $E_1=0$. Para problemas mais complexos joga-se com o peso do número de entradas correspondente ao número de vectores a considerar para uma identificação correcta e com o valor fronteira a partir do qual o perceptrão reage ou não reage.

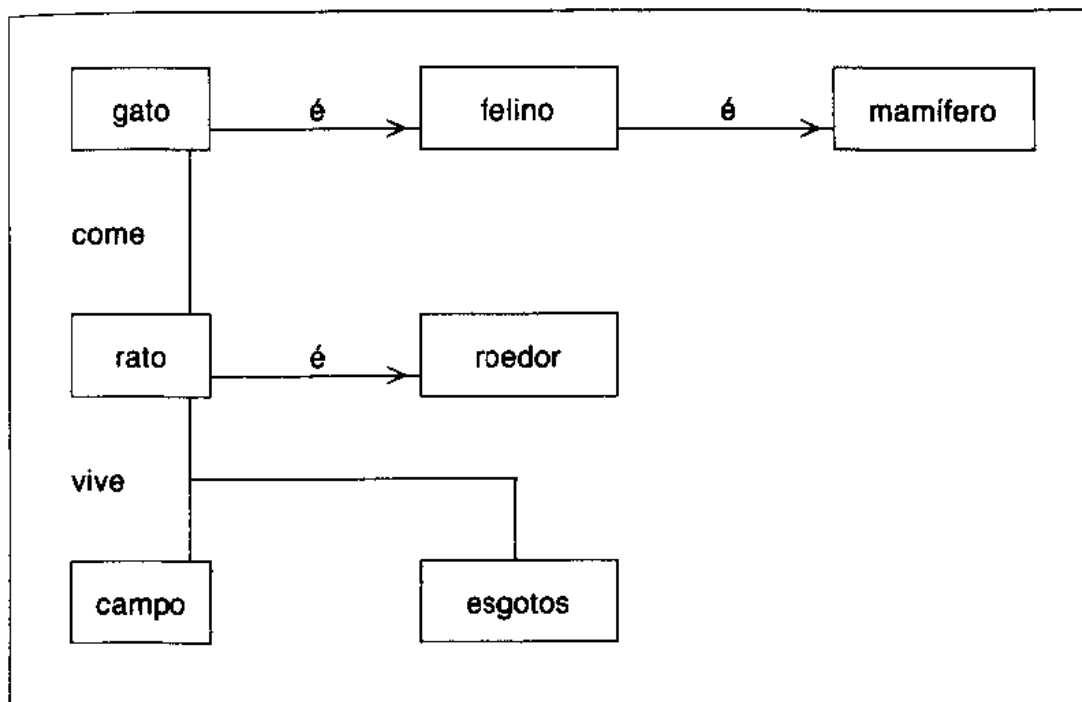
São várias as dificuldades que se colocam à compreensão da fala. Em primeiro lugar não há uma correspondência única entre cada palavra e cada som. Para dizer o mesmo fonema são utilizados sons muito diferentes por pessoas diferentes. Além disso, a hora do dia ou outros aspectos, como a arquitectura da sala, o facto da pessoa estar constipada, e outros factores parecidos podem fazer modificar muito as frequências que deveriam corresponder a cada palavra mesmo tendo em conta o mesmo falante. É claro que não precisamos de nos basear numa frequência precisa, podemos, por exemplo, decidir que fonema se está a utilizar partindo do *tipo* de curva traçada num plano de dois vectores – frequência-tempo. No entanto estas curvas também não têm um traçado completamente regular. Isto deve-se ao modo como juntamos os fonemas quando falamos. A junção de dois fonemas faz com que o primeiro influencie a dicção do segundo. E o mesmo acontece com as palavras. Problemas adicionais surgem quando consideramos a entoação ou a pronúncia do falante. Um programa de reconhecimento do discurso tem portanto de identificar as características básicas da voz do seu interlocutor, depois tem de tentar distinguir, no seio de um som contínuo, o local preciso onde se encontra cada palavra e, finalmente proceder à identificação de cada fonema da palavra.

Perante o que foi dito, a tarefa de decifrar a linguagem falada parece ser praticamente impossível de realizar. No entanto há várias técnicas que possibilitam um grau relativo de sucesso (o sistema SPHINX tem um grau de exactidão de 96%). A principal razão para este sucesso é o facto de o programa ter diversas informações de ordem morfológica, sintáctica semântica e estatística¹¹ que permitem reduzir o número de palavras confrontadas com a informação sonora. A identificação dos padrões adequados aos som é então feita por tentativa e erro.

Sistemas de processamento de linguagem natural

A compreensão da linguagem natural é um dos problemas mais difíceis que se podem colocar à IA, quer seja resolvido por programas simbólicos ou redes conexionistas. A estratégia normalmente utilizada para abordar o problema consiste em criar uma estrutura de representação de conhecimento que relaciona vários elementos da linguagem entre si. É assim possível, com base apenas numa palavra, recordar muitas características que lhe estão associadas e que lhe atribuem sentido.

¹¹ Na verdade o sistema SPHINX parece utilizar apenas uma gramática que diz quais são as palavras que podem legalmente aparecer ao lado de uma dada palavra adjacente. Além disso a informação estatística informa o programa da probabilidade de determinadas palavras aparecerem dentro de um certo contexto.



Representação gráfica de uma rede semântica simplificada
(retirado de *IntAr*, p.33)

Para além das redes semânticas (e das *frames* que são redes semânticas mais complexas que podem descrever objectos e estão ligadas entre si) necessitamos ainda de processos de inferência que possibilitem a criação de informação a partir das relações entre palavras que o programa possui. Este tipo de motor pode, por exemplo, abstrair informações do género de tipo silogístico (com premissas maiores e menores) ou utilizando conhecimento acessório sobre uma dada relação entre conceitos. (por exemplo, 'Se X é meu primo o pai de X é meu tio'). No entanto, mesmo possuindo este conhecimento sobre o mundo seremos ainda incapazes de tratar certas frases. Para isso precisamos de ter ainda uma descrição de processos comuns. Esta descrição é designada por *script*.

Por exemplo (o exemplo é adaptado de AI, pp. 14-22) consideremos a frase

'Mary went shopping for a new coat. She found a red one she really liked.
When she got it home, she discovered that it went perfectly with her favorite dress.'

Para responder à questão '*What did Mary go shopping for?*' não precisamos de um programa que compreenda do que está a falar (à semelhança do computador puramente sintáctico de Searle). Basta-nos ter um conhecimento de certas expressões chave (neste caso '*What did x y*') que retribuam uma expressão do tipo $x y z$. Basta tentar encontrar $x y$ no texto – '*shopping for*' – e somar-lhe z – '*a new coat*'. Para saber quantas palavras z terá de incluir será talvez necessário alguma função que leve em conta a estrutura da frase (por exemplo o número de letras que nos separa do ponto final) mas nada é necessário de um ponto de vista da compreensão do sentido do texto.

No entanto, se colocássemos a questão '*What did Mary find that she liked?*' teríamos muita dificuldade em responder à questão com um programa tradicional. Já que não há nenhuma estrutura que se repita na questão e na citação.

Para poder responder a perguntas que envolvam pronomes é preciso especificar o conteúdo do texto em forma simbólica. A representação estruturada da frase '*She found a red one she really liked.*' pode ser descrita por uma *frame* da seguinte forma (imagem retirada de AI, p.17):

Event2

instance : Finding
tense : Past
agent : Mary
object : Thing1

Thing1

instance : Coat
color : Red

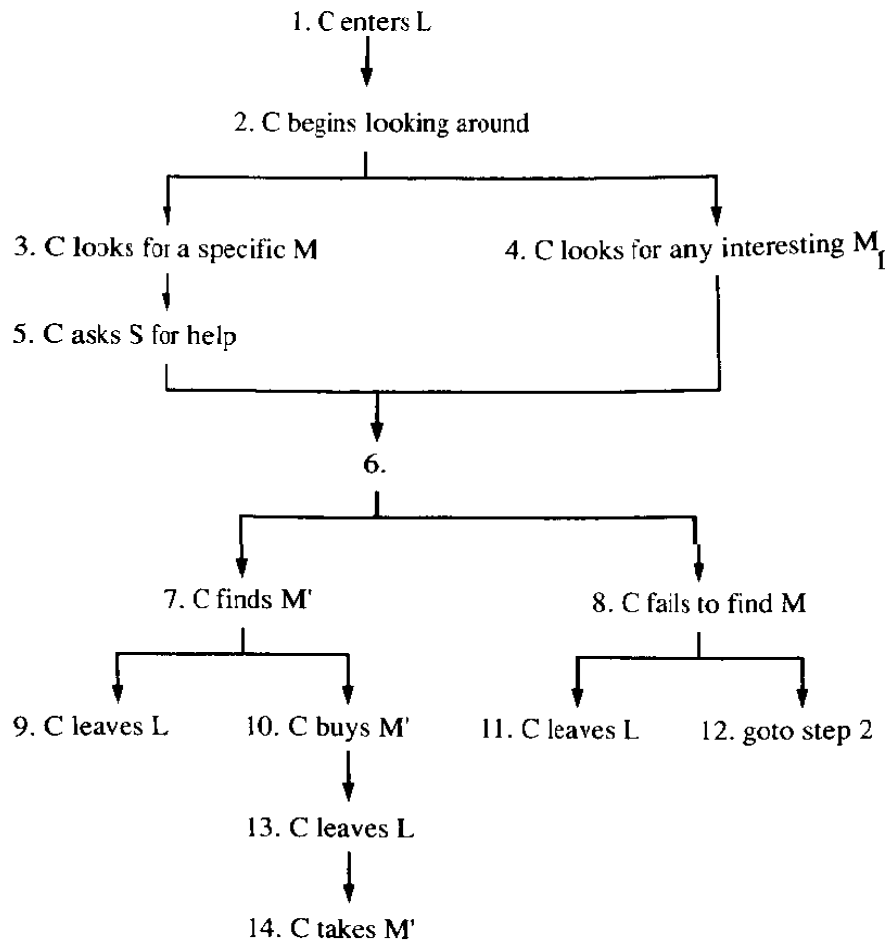
Event2

instance : Liking
tense : Past
modifier : Much
object : Thing1

O significado de cada um dos termos da frame é dado pela ligação de cada termo a outros termos e a um conjunto de procedimentos. Temos ainda de possuir um mecanismo que nos possibilite a passagem das frases em linguagem natural para linguagem estruturada e vice-versa (se quisermos uma resposta). No entanto, mesmo com o acesso a este tipo de informação o programa não será capaz de responder à questão *'Did Mary buy anything?'*. A resposta a esta questão não é dada directamente pelo texto. Para a extrairmos do texto é necessário saber uma série de factos sobre o mundo. O *script*, de que falámos há pouco proporciona precisamente essa informação suplementar. Ele constitui uma descrição dos procedimentos adoptados em determinadas situações. Um *script* simplificado para o que se faz numa loja poderia assumir a seguinte forma (a figura é da p.20):

Shopping Script:

roles: C (customer), S (salesperson)
props: M (merchandise), D (dollars)
location: L (a store)



Na posse desta informação o programa seria agora capaz de responder à terceira questão. Visto que, através da última frase do texto apresentado, o programa sabe que o passo 14 do *script* foi realizado. E pode portanto responder que Mary comprou um casaco vermelho (e, idealmente, acrescentar) a não ser que o tenha roubado ou que a tenham deixado experimentar em casa.

O problema de desambiguar o sentido de uma palavra de uma frase ou de um texto pode ser resolvido se acrescentarmos a esta estrutura um método de busca do tipo tentativa e erro. Quando se alcança uma frase que pode ter mais do que um sentido o programa interpreta-a de todas as maneiras possíveis. Cada interpretação gera então um conjunto de expectativas¹² que o programa deve satisfazer. Qualquer método

¹² Pode também utilizar-se o raciocínio contrário. Isto é, gerar previamente uma série de restrições baseadas no contexto tentando encontrar uma interpretação que as respeite. Se se encontrar mais de uma interpretação possível tenta-se encontrar restrições adicionais (por exemplo, continuando a ler o texto). Se

de procura que permita a rectificação (monotónico), quer se aproxime mais da pesquisa em extensão ou em profundidade, será então capaz de escolher a interpretação mais adequada.

Esta exposição está bastante longe de ter descrito eficientemente os aspectos essenciais dos vários processos utilizados em IA (por exemplo, a distinção entre conhecimento e procedimentos heurísticos é muito mais difusa do que indiquei aqui). No entanto penso ter mostrado que: (1) Há computadores reais que fazem coisas como o reconhecimento do discurso ou a compreensão da linguagem natural. (2) Os processos utilizados pela IA envolvem a manipulação de uma quantidade incrível de conhecimento e de processos que relacionam todos os dados entre si resultando daí a possibilidade de se reconhecer as propriedades de uma dada estrutura semântica de uma frase através da análise das suas palavras.

No entanto creio que seria irrealista dizer para já que a IA (seja simbólica ou conexionista) é capaz de fazer tudo o que o homem é capaz de fazer (o sonho de Turing) ou descobrir nela alguma limitação. A ideia que fica da leitura de livros sobre a engenharia de projectos de IA é que ainda ninguém sabe muito bem o que se pode e não pode fazer e cada projecto novo aumenta significativamente a capacidade de visualizar novos rumos de acção para toda a comunidade científica. Parece que a IA está ainda longe de atingir o seu paradigma 'definitivo'.

nenhuma interpretação for possível tenta-se rever o conjunto de restrições a que chegámos (por exemplo, tentando reinterpretar o texto já lido).