



UNIVERSIDADE FEDERAL DA BAHIA
Escola Politécnica
Instituto de Matemática
Programa de Pós-Graduação em Mecatrônica

Ricardo André Oliveira dos Reis

**UMA INTERFACE ROBÓTICA BASEADA EM LINGUAGEM
NATURAL**

Salvador

2006

Ricardo André Oliveira dos Reis

UMA INTERFACE ROBÓTICA BASEADA EM LINGUAGEM NATURAL

Trabalho apresentado ao Programa de Pós-graduação em Mecatrônica, programa conjunto entre o Departamento de Engenharia Mecânica e o Departamento de Ciência da Computação, da UNIVERSIDADE FEDERAL DA BAHIA como requisito parcial para obtenção do grau de Mestre em Mecatrônica.

Orientador: Prof. Dr. *Leizer Schnitman*

Salvador

2006

A meus pais, José Antonio e Ivonice pelo apoio incondicional, pelo amor, doação e educação dados ao filho que os ama muito.

A meus familiares, em especial meus irmãos Fábio e Cláudio, pela cooperação e privação das horas de convívio tão importantes para mim em decorrência da dedicação a meus estudos.

Com muito amor, à minha noiva e futura esposa Elaine, a quem sempre posso recorrer nos momentos de dificuldade, que esteve o tempo todo ao meu lado me incentivando e me dando forças.

AGRADECIMENTOS

À FAPESB, pela bolsa de mestrado concedida durante o curso.

Ao pessoal do projeto SGPA da UFBA, pela oportunidade de pesquisa em sua parceria com a Petrobras.

A Fábio Reis e Vitor Leão, pelo apoio no projeto.

A meu orientador, Prof. Dr. Leizer Schnitman, pelo incentivo, paciência, ensinamento, dedicação e tempo dispensados no criterioso trabalho de acompanhamento e orientação deste trabalho.

Aos professores e funcionários do Mestrado Acadêmico em Mecatrônica da Universidade Federal da Bahia, pelo apoio à realização deste curso.

A todos que direta ou indiretamente me apoiaram para a realização deste trabalho.

“Não são apenas as estrelas no universo que fascinam o homem com o seu impressionante número. Em um outro universo, o nosso universo biológico interno, uma gigantesca ‘galáxia’ com centenas de milhões de pequenas células nervosas que formam o cérebro e o sistema nervoso comunicam-se umas com as outras através de pulsos eletroquímicos para produzir atividades muito especiais: nossos pensamentos, sentimentos, dor, emoções, sonhos, movimentos, e muitas outras funções mentais e físicas, sem as quais não seria possível expressarmos toda a nossa riqueza interna e nem perceber o nosso mundo externo, como o som, cheiro, sabor, e também luz e brilho, inclusive o das estrelas...”

—SILVIA HELENA CARDOSO (1998)

RESUMO

As interfaces entre os seres humanos e as máquinas tornam-se mais amigáveis a cada dia. A evolução tecnológica destas interfaces tende para formas mais humanas de comunicação. Máquinas com interfaces mais amigáveis têm o intuito de facilitar a sua utilização por usuários sem treinamentos técnicos específicos.

O objetivo deste trabalho é a proposição de uma interface entre seres humanos e robôs utilizando a linguagem natural escrita no idioma português como protocolo de comunicação. Neste contexto, enfatizou-se uma abordagem para a utilização de técnicas da Descoberta de Conhecimento em Textos baseada em Conceitos, cujo processo visa extrair padrões interessantes e não triviais ou conhecimento de documentos textuais.

Apesar dos recentes avanços e da diversidade de artigos e pesquisas na área, aplicações das técnicas de processamento de linguagem natural ainda se encontram em estágio incipiente de desenvolvimento. O método proposto neste trabalho tem por objetivo interpretar o texto escrito e associar comandos de execução de tarefas a serem realizadas, característica que diferencia a técnica apresentada das técnicas usuais. Exemplos reais de aplicação estão ilustrados neste trabalho e um estudo de caso é enfatizado com aplicação em uma interface robótica.

Palavras-chaves: Processamento de Linguagem Natural. Descoberta de Conhecimento em Textos. Interface Robótica.

ABSTRACT

The interfaces among men and machines have become friendlier each day. The technological evolution of these interfaces heads to more human forms of communication. Machines with friendlier interfaces have the purpose of facilitating their usage by those without specific technical training.

The goal of this work is the proposal of an interface among human beings and robots by using a natural language written in Portuguese as a communication protocol. In this context, emphasis is given to an approach for the usage of techniques of the Knowledge Discovery from Text based in Concepts whose process aims to extract interesting and not trivial patterns or knowledge of textual documents.

Despite recent advances and the diversity of articles and researches in this area, the application of natural language processing techniques is still in incipient stages of development. The proposed method is applied to interpreting the written text and associating commands of execution of tasks to be accomplished. This characteristic differentiates the technique presented from the usual ones. Real examples of applications are illustrated and an application in a robotic interface is highlighted as an example.

Keywords: Natural Language Processing. Knowledge Discovery from Text. Robotic Interface.

LISTA DE FIGURAS

2.1	Estrutura em árvore de análise gramatical de uma sentença (LEVINE; DRANG; EDELSON, 1988).	15
2.2	Estrutura em forma de árvore a partir da análise <i>Top-Down</i>	16
2.3	Exemplo de gramática semântica (KNIGHT; RICH, 1993).	20
2.4	Resultado da análise com uma gramática semântica (KNIGHT; RICH, 1993).	21
3.1	Dado, Informação, Conhecimento e Inteligência (REZENDE, 2003). . .	25
3.2	Etapas do processo de KDD.	29
3.3	Método para KDT (TAN, 1999).	36
3.4	Estrutura do Módulo JARGÃO.	49
3.5	Visualização de agrupamento de poços no SGPA.	54
4.1	Diagrama eletrônico do protótipo.	56
4.2	O protótipo.	58
4.3	Exemplo de reconhecimento de ações a partir de um texto.	66
5.1	Interface para execução da tarefa em linguagem natural.	78
A.1	Quadro com cantoneiras.	96

A.2	Primeira estrutura do robô.	96
A.3	Componentes do sistema de locomoção.	97
A.4	Primeiro protótipo montado.	97
A.5	Segundo quadro do robô.	98
A.6	Estrutura montada do robô.	98
A.7	Montagem do segundo robô.	99
A.8	Possibilidades de giro do Robô.	99
A.9	Mecanismo de Visão.	100
A.10	Montagem final.	100
A.11	Posicionamento do servo relativo à largura de pulso.	101
A.12	Possibilidades de giro do Robô.	104
A.13	Imagem capturada da câmera.	106
A.14	Dados da localização do tom de cor na imagem.	107
A.15	Placa eletrônica principal do robô com suas regiões mais importantes destacadas.	108
A.16	Placa auxiliar para conexões de servos motores e sensores.	111

LISTA DE TABELAS

2.1	Alguns elementos gramaticais formadores de frase (WINSTON, 1987) (LEVINE; DRANG; EDELSON, 1988).	11
2.2	Gramática livre de contexto, definida em (LEVINE; DRANG; EDELSON, 1988).	12
2.3	Aplicação de uma gramática livre de contexto.	14
2.4	Regras de uma gramática (LEVINE; DRANG; EDELSON, 1988).	15
2.5	Gramática 2.	16
2.6	Aplicação da análise <i>Top-Down</i>	17
2.7	Aplicação da análise <i>Bottom-Up</i>	17
3.1	Distribuições de Conceitos e regras associativas encontradas em texto de psiquiatria Loh (2001a).	42
3.2	Regras do conceito Internet.	47
3.3	Alguns conceitos da engenharia de petróleo.	51
3.4	Dados de entrada da Rede SOM.	53
3.5	Conceitos identificados em alguns poços.	53
4.1	Conceitos implementados na interface para o mapeamento dos comandos.	61
4.2	Associação dos conceitos comandos com os conceitos parâmetros.	62

4.3	Conceitos parâmetros.	62
4.4	Algoritmo para o reconhecimento de conceitos.	64
4.5	Algoritmo para o reconhecimento da ação.	65
4.6	Comando Movimentar.	66
4.7	Algoritmo da ação movimentar.	67
4.8	Tom amarelo.	67
4.9	Comando Procurar.	68
4.10	Algoritmo da ação procurar.	68
4.11	Algoritmo da ação seguir.	70
4.12	Comando Movimentar câmera.	71
4.13	Algoritmo da ação movimentar câmera.	71
4.14	Comando Capturar.	72
4.15	Algoritmo da ação capturar.	72
5.1	Tarefa em linguagem natural.	73
5.2	Orações da tarefa descrita na tabela 5.1.	74
5.3	Passos para o reconhecimento de uma ação na oração “Robô, movimente-se para a frente por 5 segundos.”.	75
5.4	Passos para o reconhecimento de uma ação na oração “Logo em seguida, vire para a direita.”.	75
5.5	Passos para o reconhecimento de uma ação na oração “Depois, vá para a esquerda.”.	76
5.6	Passos para o reconhecimento de uma ação na oração “Procure a cor amarela no ambiente por 120 segundos, no máximo.”.	76

5.7	Passos para o reconhecimento de uma ação na oração “Movimente a câmera para baixo.”.	77
5.8	Passos para o reconhecimento de uma ação na oração “Capture uma imagem do ambiente e armazene.”.	77
5.9	Ações reconhecidas.	78
5.10	Exemplo de tarefa submetida à interface.	79
5.11	Ações reconhecidas na tarefa descrita na tabela 5.10.	79
5.12	Descrição de tarefa submetida à interface.	79
5.13	Ações reconhecidas na tarefa descrita na tabela 5.12.	80
5.14	Descrição de tarefa submetida à interface.	80
5.15	Ações reconhecidas na tarefa descrita na tabela 5.14.	80
5.16	Novas regras para o conceito movimentar.	82
5.17	Orações que são ignoradas pela interface.	83
5.18	Definição do conceito movimentar em diversas línguas.	85
A.1	Lista dos componentes do sistema de locomoção.	97
A.2	Estrutura do protocolo de comunicação do módulo eletrônico.	101
A.3	Posicionamento do servo relativo à largura de pulso.	102
A.4	Estrutura do <i>byte</i> utilizado para o controle do motor DC.	102
A.5	Estado do motor DC em relação ao sinal aplicado.	102
A.6	Velocidade do motor DC em relação ao sinal aplicado.	103
A.7	Direção do movimento do robô em relação ao acionamento dos motores DC.	103
A.8	Lista de componentes utilizados na placa do robô.	110

LISTA DE ABREVIATURAS

ADJ	Adjetivo
ADJS	Adjetivos
DET	Determinante
F	Frase
IA	Inteligência Artificial
KDD	Knowledge Discovery in Databases
KDT	Knowledge Discovery from Text
MEV	Modelo Espaço de Vetores
O	Oração
PLN	Processamento de Linguagem Natural
PREP	Preposição
PWM	Pulse-Width Modulation
RI	Recuperação de Informação
SGPA	Sistema de Gerenciamento de Poços Automatizados
SN	Sintagma nominal
SOM	Self-Organizing Maps
SP	Sintagma preposicional
SPS	Sintagmas preposicionais
SUB	Substantivo
SV	Sintagma verbal
V	Verbo

SUMÁRIO

Capítulo 1—Introdução	1
Capítulo 2—Compreensão da Linguagem Natural	5
2.1 Etapas do Processo	6
2.2 Análise Morfológica	7
2.2.1 Ambigüidades Léxicas	9
2.3 Análise Sintática	10
2.3.1 Gramática livre de contexto	12
2.3.2 Análises <i>Top-Down</i> e <i>Bottom-Up</i>	14
2.4 Análise Semântica	17
2.4.1 Gramática Semântica	20
2.5 Análise Pragmática	22
Capítulo 3—Descoberta de Conhecimento	24
3.1 Dado, Informação e Conhecimento	25
3.2 Descoberta Reativa X Proativa	27
3.3 Descoberta de Conhecimento em Bases de Dados	28

3.3.1	Etapas do processo de KDD	28
3.3.2	Métodos tradicionais de <i>Data Mining</i>	30
3.3.3	Áreas de Aplicações	33
3.4	Descoberta de Conhecimento em Texto	34
3.4.1	Etapas do processo de KDT	35
3.4.2	Estratégias Baseadas em Palavras e Baseadas em Conceitos	38
3.5	Métodos para KDT	39
3.5.1	Recuperação de Informação	40
3.5.2	Extração de Informação	40
3.5.3	Agrupamento	40
3.5.4	Classificação	41
3.5.5	Associação	41
3.5.6	Sumarização	42
3.5.7	Outros métodos	43
3.6	Descoberta de Conhecimento em Texto Baseado em Conceitos	43
3.6.1	Modelo Espaço de Vetores	44
3.6.2	Modelo Contextual	45
3.6.3	Exemplo de aplicação do KDT baseado em Conceitos	48
3.6.4	Sistema SGPA	48
Capítulo 4—Uma proposta para a interface robótica		55
4.1	Estrutura de <i>hardware</i>	55

4.2	Definição do contexto	57
4.2.1	Conceitos implementados na interface	59
4.3	Algoritmos implementados	63
4.3.1	Reconhecimento dos conceitos	63
4.3.2	Reconhecimento das ações	64
4.3.3	Execução das ações	65
Capítulo 5—Resultados experimentais		73
5.1	Exemplo detalhado	73
5.2	Outros exemplos	79
5.3	Limitações da interface proposta	80
5.4	Considerações Finais	84
Capítulo 6—Conclusões		87
6.1	Trabalhos futuros	88
Apêndice A—Estrutura do robô		95
A.1	Detalhamento da estrutura do robô	95
A.1.1	Mecanismo de movimentação da câmera	99
A.2	Detalhamento do módulo eletrônico	101
A.2.1	Protocolo de comunicação do módulo eletrônico	101
A.2.2	Protocolo de comunicação do <i>firmware</i> da câmera CMUCam1	104
A.2.3	Componentes eletrônicos do módulo eletrônico	107

INTRODUÇÃO

Atualmente, equipamentos eletrônicos fazem parte do cotidiano dos seres humanos. Diversas vezes, há a necessidade de se aprender formas não intuitivas de comunicação para interagir com estas máquinas e executar alguma tarefa cotidiana.

A comunicação com as máquinas pode ser estabelecida utilizando comandos específicos definidos pelo fabricante do equipamento, acessando menus através de algum dispositivo de saída de vídeo, bem como pressionando botões sequencialmente para ativar uma determinada função. No entanto, as interfaces entre os seres humanos e as máquinas tornam-se cada dia mais amigáveis e sua evolução tecnológica tende para formas mais humanas de comunicação, como a utilização de recursos de voz (AMARAL; BARRIVIERA; TEIXEIRA, 2004) e de processamento de imagem (KITANO et al., 1997).

Neste contexto, uma possível alternativa para a interface entre os seres humanos e as máquinas é dotá-las com capacidade de compreender a linguagem natural utilizada pelos seres humanos para comunicação e descrição das tarefas cotidianas. Em Levine, Drang e Edelson (1988), a criação de um sistema de comunicação entre pessoas e computadores é chamado de Processamento de Linguagem Natural (PLN). O objetivo final do PLN é permitir que o usuário interaja com o computador usando a linguagem humana, dotando o computador com recursos para estabelecer a comunicação. O PLN é um dos segmentos da Inteligência Artificial (IA), definido em Santos e Pelizzoni (2005) como uma área de trabalho situada entre a lingüística e a ciência da computação, que lida com os aspectos computacionais da capacidade humana de linguagem. O

objetivo do PLN inclui a construção de sistemas que tratem de problemas intrínsecos ao processo comunicativo.

Em Amaral, Barriviera e Teixeira (2004), o comando de voz é utilizado para a automação residencial, através de um sistema computacional desenvolvido em uma arquitetura baseada em agentes inteligentes. O objetivo desta automação residencial é auxiliar e proporcionar maior independência às pessoas portadoras de necessidades especiais. Avanços tecnológicos permitem ainda o desenvolvimento de sistemas específicos para um único locutor, ou independentes de locutores, como proposto em Borges (2001).

Diversos exemplos de interfaces amigáveis também incluem o uso de câmeras e processamento de imagens. Um desses exemplos é visto em Kitano et al. (1997), grupos de pesquisa recebem problemas encontrados em um jogo típico de futebol de robôs. Os grupos de pesquisa desenvolvem diversas técnicas de IA que são integradas aos robôs com o propósito de solucionar os problemas encontrados na competição.

No contexto de interfaces amigáveis, uma possível alternativa para a interface entre os seres humanos e as máquinas é dotá-las com capacidade de compreender a linguagem¹ natural utilizada pelos seres humanos para comunicação e descrição das tarefas cotidianas. Segundo Levine, Drang e Edelson (1988), a criação de um sistema de comunicação entre pessoas e computadores é chamado de Processamento de Linguagem Natural (PLN). O PLN é um dos segmentos da Inteligência Artificial (IA), definido em Santos e Pelizzoni (2005) como uma área de trabalho situada entre a lingüística e a ciência da computação, que lida com os aspectos computacionais da capacidade humana de linguagem. O objetivo final do PLN é permitir que o usuário interaja com o computador usando a linguagem humana, dotando o computador com recursos para estabelecer a comunicação.

Por outro lado, a utilização da língua portuguesa na construção de interfaces com-

¹O uso da palavra articulada ou escrita como meio de expressão e de comunicação entre pessoas. (FERREIRA, 1975)

putacionais com base em PLN requer muito conhecimento sobre os aspectos formadores de palavras e orações deste idioma. Entender linguagem natural constitui uma tarefa difícil, pois requer conhecimento específico da linguagem utilizada e conhecimento do contexto no qual o tema da conversa está inserido (RICH, 1986). Para a análise da linguagem propriamente dita, em Levine, Drang e Edelson (1988) o estudo é dividido em três partes básicas: análise léxica, análise sintática e análise semântica, o que traz à tona os problemas de lingüística e a complexidade computacional correspondente. Assim, apesar da diversidade de artigos e pesquisas na área, muitos trabalhos que abordam o PLN - a exemplo de Amaral, Barriviera e Teixeira (2004) - apresentam restrições quanto à interpretação do contexto, ou seja, são usualmente restritos a comandos específicos e explícitos.

Uma alternativa para o problema da complexidade de compreensão de orações da língua portuguesa é a utilização de técnicas da Descoberta de Conhecimento em Textos (KDT). Técnicas de KDT podem ser utilizadas para a criação de uma interface homem-máquina de tal modo que um ser humano possa descrever todos os passos para a execução de uma tarefa. A partir da análise do texto contendo os procedimentos, serão extraídos os conceitos relativos à tarefa a ser cumprida e, com base nestes, serão definidas as ações.

A idéia deste trabalho é a utilização da linguagem natural como interface entre seres humanos e máquinas. A interface com o equipamento utiliza técnicas de KDT como base para a análise da linguagem natural. Textos digitais compõem a entrada de dados, descrevendo detalhadamente os passos a serem executados com o objetivo de cumprir as tarefas. O processamento de imagem é utilizado na interface com o intuito de guiar e auxiliar na execução das ações necessárias à realização da tarefa. Um robô móvel é apresentado como estudo de caso.

O capítulo 2 trata de aspectos teóricos da lingüística, e tem como objetivo apresentar os fundamentos do PLN. No capítulo 3, o tema de KDT é detalhado e alguns exemplos de aplicação são ilustrados. O capítulo 4 trata do estudo de caso propri-

amente dito, onde detalhes de projeto do robô são citados e uma associação é feita entre a aplicação proposta e os aspectos teóricos apresentados nos capítulos anteriores. Os resultados de experimentos e as conclusões estão dispostas nos capítulos 5 e 6, respectivamente.

COMPREENSÃO DA LINGUAGEM NATURAL

Este capítulo traz alguns conceitos fundamentais de lingüística, com o objetivo de oferecer uma visão mais ampla do PLN. Além disso, esta seção busca prover uma noção mais clara da complexidade da implementação computacional.

A capacidade de compreender a linguagem natural é inata aos seres humanos. A linguagem natural está presente em tudo que os humanos fazem no seu dia-a-dia, sendo utilizada para interagir com outras pessoas, tomar conhecimento do que está acontecendo no mundo por meio de um jornal, acessar ou fornecer descrições de como uma tarefa deve ser executada passo a passo etc.

Enquanto a compreensão da linguagem natural é elementar e cotidiana para os seres humanos, para os computadores é um processo considerado complexo. Exige, por exemplo, conhecimentos em lingüística, que as máquinas não utilizam para seu funcionamento. O significado de uma oração pode ser entendido pela maioria das pessoas, mas a liberdade de estruturas para linguagem natural faz com que a sua compreensão pelos computadores seja muito difícil. Ao contrário da sintaxe restrita de uma linguagem de programação, as orações dos seres humanos podem ser vagas, palavras podem ser ambíguas ou a oração pode ser fragmentada (FROMM; DREWS, 1998).

Para que um computador entenda uma determinada língua, é necessário dividi-la em elementos básicos e inserir esses dados no computador, além de desenvolver programas que aceitem esses dados. Em Levine, Drang e Edelson (1988), a criação desses sistemas de comunicação entre seres humanos e computadores é chamada de

Processamento de Linguagem Natural.

Segundo Schildt (1989), muitos profissionais em IA acreditam que a mais importante tarefa que a IA pode resolver é o Processamento de Linguagem Natural. Essa opinião se deve ao fato de que o advento de máquinas com capacidade de entender e se comunicar com os seres humanos - através da linguagem com a qual estes estão bem familiarizados - possibilitará a interação entre homem e máquina de forma mais elementar.

O Processamento de Linguagem Natural (PLN) é uma área de pesquisa e aplicação que explora as maneiras como computadores podem ser usados para entender e manipular a fala ou textos em linguagem natural e, a partir do seu entendimento e interpretação, realizar ações consideradas úteis (CHOWDHURY, 2003).

Em Santos e Pelizzoni (2005), PLN é definido como uma área de trabalho situada entre a lingüística e a ciência da computação que lida com os aspectos computacionais da capacidade humana de linguagem. O objetivo desse processamento é construir sistemas que tratem de problemas intrínsecos ao processo comunicativo verbal e fazer com que o computador entenda comandos escritos em linguagens naturais para os humanos (SCHILDT, 1989).

O PLN consiste no desenvolvimento de modelos computacionais para a realização de tarefas que dependem de informações expressas em alguma linguagem natural como, por exemplo, tradução e interpretação de textos, busca de informações em documentos e interface homem-máquina (PEREIRA, 2006).

2.1 ETAPAS DO PROCESSO

Segundo Knight e Rich (1993), o Processamento de Linguagem Natural é dividido em quatro etapas:

Análise Léxica - Nesta etapa, as palavras do texto são analisadas isoladamente, e

cada uma destas é atribuída a categorias sintáticas. Os sinais presentes no texto, como a pontuação, são tratados à parte.

Análise Sintática - As seqüências de palavras são transformadas em estruturas que mostram como as palavras estão relacionadas entre si. Esta transformação utiliza regras da linguagem definidas em uma gramática.

Análise Semântica - Nesta etapa, o resultado da análise sintática recebe significado, sendo formado por estruturas de palavras. As estruturas que não tenham significado atribuído a elas geralmente podem ser rejeitadas.

Análise Pragmática - Nesta etapa, a estrutura que representa o que foi dito é reinterpretada para obter o que realmente se quis transmitir com a frase (KNIGHT; RICH, 1993). O significado da frase implica em relações com o contexto e a intenção do interlocutor (SANTOS; PELIZZONI, 2005).

Na seqüência, são destacados alguns detalhes das etapas do PLN.

2.2 ANÁLISE MORFOLÓGICA

A Análise Morfológica, também conhecida como Análise Léxica, consiste no estágio que considera a formação das palavras. Esta etapa leva em consideração, por exemplo, a classe da palavra, a flexão, os elementos mórficos, a terminação e a grafia (LEVINE; DRANG; EDELSON, 1988). Conforme estabelecido em Cegalla (1997), os elementos mórficos podem ser classificados em:

Raiz - “É o elemento originário e irredutível em que se concentra o significado das palavras, considerada do ponto de vista histórico. Geralmente monossilábica, a raiz encerra sentido lato¹ e geral, comum às palavras da mesma família etimológica²”.

¹Largo, dilatado, extenso (FERREIRA, 1975).

²Relativo à etimologia. Estudo sobre a origem das coisas (FERREIRA, 1975).

Radical - É o elemento básico e significativo das palavras, consideradas sob o aspecto gramatical.

Tema - É o radical acrescido de uma vogal. Esta vogal é conhecida como vogal temática.

Afixos - São os elementos secundários que se agregam a um radical ou tema para formar palavras derivadas. Estes elementos podem ser classificados em:

Prefixos - Elementos secundários que são antepostos ao radical ou tema.

Sufixos - Elementos secundários que são pospostos ao radical ou tema.

Desinências - São os elementos terminais indicativos das flexões das palavras, que podem ser classificados em:

Desinências nominais - Indicam as flexões das palavras quanto a gênero - masculino e feminino -, e quanto a número - singular e plural.

Desinências verbais - Indicam as flexões de número, pessoa, modo e tempo dos verbos.

Vogal Temática - É o elemento que é acrescido ao radical e forma o tema de nomes e verbos. As vogais temáticas “a”, “e” e “i” aplicadas aos verbos caracterizam, respectivamente, os verbos da 1ª, 2ª e 3ª conjugações.

Por exemplo, em Levine, Drang e Edelson (1988), as palavras “caixote” e “reluzir” são divididas da seguinte forma:

- **caixote**

- caix (raiz)

- ote (sufixo)

- **reluzir**

- re (prefixo)
- luz (tema)
- ir (desinência)

Nesta análise, geralmente todas as palavras da frase são classificadas em categorias sintáticas (KNIGHT; RICH, 1993). Por exemplo, em Knight e Rich (1993), a expressão “do Mário” é analisada da seguinte forma:

- “do” - separada na *preposição* “de” seguida do *artigo* “o”
- “Mário” - no *substantivo próprio* “Mário”

2.2.1 Ambigüidades Léxicas

Conforme visto em Krovetz e Croft (1992), as ambigüidades léxicas podem ser de dois tipos: sintática e semântica. A ambigüidade sintática ocorre quando uma palavra pode ser classificada em mais de uma categoria sintática. Por exemplo, nas frases “Eles *verão* que dará errado.” e “O *verão* é a melhor estação climática do ano.”, a palavra “*verão*” está presente nas duas frases, porém é classificada em categorias sintáticas diferentes - um verbo e um substantivo, respectivamente.

A ambigüidade semântica ocorre quando uma palavra apresenta diferentes significados, podendo ocorrer polissemia e/ou homonímia (KROVETZ; CROFT, 1992).

Polissemia - Polissemia é a propriedade que uma palavra tem de apresentar vários significados. Exemplos de Polissemia incluem:

- Ele ocupa um alto *posto* na empresa;
- Abasteci meu carro no *posto* da esquina;
- Tudo está *posto* na mesa;
- Os convites eram de *graça*;

- Os fiéis agradecem a *graça* recebida.

Homonímia - A homonímia ocorre quando duas ou mais palavras apresentam significados diferentes, porém possuem a mesma estrutura fonológica³. As palavras homônimas se apresentam de três maneiras:

Homógrafas heterofônicas ou homógrafas - São as palavras escritas do mesmo modo mas pronunciadas diferentemente. Por exemplo, as palavras *gosto* (substantivo) e *gosto* (verbo), respectivamente, em:

- O *gosto* da maçã é delicioso.
- Eu *gosto* de maçã.

Homófonas heterográficas ou homófonas - São as palavras grafadas diferentemente e pronunciadas da mesma maneira. Por exemplo, as palavras *cela* (substantivo) e *sela* (verbo), respectivamente, em:

- A *cela* da cadeia é pequena.
- Maria *sela* muito bem o cavalo.

Homófonas homográficas ou homônimos perfeitos - São as palavras escritas da mesma maneira e pronunciadas igualmente. Por exemplo, as palavras *cedo* (verbo) e *cedo* (advérbio), respectivamente, em:

- Eu *cedo* aos teus beijos.
- Famosos chegam *cedo* para a festa.

2.3 ANÁLISE SINTÁTICA

A definição de análise sintática em Levine, Drang e Edelson (1988) estabelece que a palavra é analisada com relação às outras que estão na mesma oração. Em Knight e Rich (1993), a análise sintática é descrita como aquela que explora os resultados da análise morfológica para criar uma descrição estrutural da frase, com o objetivo

³Refere-se à fonologia, estudo dos sons da linguagem (FERREIRA, 1975).

de converter a lista de palavras que formam a frase em uma estrutura que defina a unidade gramatical de cada uma dessas palavras.

A análise sintática representa um papel fundamental no processamento da linguagem natural, pois separa os componentes de uma frase que serão utilizados na análise semântica. Com isto, o processamento semântico terá um número limitado de componentes para a sua análise, reduzindo sua complexidade. A análise sintática requer menos carga computacional do que a análise semântica, pois a semântica envolve inferência sobre os componentes da frase, o que pode tornar a carga computacional elevada (KNIGHT; RICH, 1993).

Na análise semântica, é necessário que haja um glossário para que o processo possa identificar cada palavra na frase ao seu respectivo elemento gramatical. Conforme visto em Winston (1987), os lingüistas raramente escrevem nomes completos dos constituintes das frases, geralmente usam siglas ou abreviaturas. Sendo assim, alguns constituintes de frases estão dispostos na tabela 2.1, com seus nomes completos e suas abreviaturas:

Tabela 2.1. Alguns elementos gramaticais formadores de frase (WINSTON, 1987) (LEVINE; DRANG; EDELSON, 1988).

Nome Completo	Abreviatura
Frase	F
Oração	O
Sintagma ^a nominal ^b	SN
Sintagma verbal ^c	SV
Sintagma preposicional	SP
Sintagmas preposicionais	SPS
Determinante	DET
Adjetivo	ADJ
Adjetivos	ADJS
Substantivo	SUBS ou SUB
Verbo	V ou VERBO
Preposição	PREP

^aSintagma é uma unidade formada por uma ou várias palavras que desempenham uma função na frase (PINILLA; RIGONI; INDIANI, 2006).

^bÉ a parte de uma sentença com comportamento de sujeito ou objeto (GONZALEZ; LIMA, 2001).

^cÉ o conjunto de elementos que se organizam em torno de um verbo (PINILLA; RIGONI; INDIANI, 2006).

Além da existência de um glossário, é necessário que regras gramaticais sejam definidas para que o computador possa manipular as frases e compreender a linguagem natural. A utilização de gramática livre de contexto é o método clássico para incorporar restrições sintáticas e caracterizar o modo como as palavras se relacionam em uma frase. Os detalhes deste método serão expostos a seguir.

2.3.1 Gramática livre de contexto

As gramáticas livres de contexto consistem de regras independentes do contexto. Isto é, estas gramáticas⁴ são compostas por diversas regras gramaticais, onde o lado direito de cada regra consiste apenas no símbolo a ser substituído pelo lado esquerdo da regra e nada influencia no modo como a substituição é feita (WINSTON, 1987).

Formalmente, estas gramáticas são definidas como uma 4-upla (Sn, St, Rp, Si), onde: Sn é um conjunto de símbolos não-terminais; St é um conjunto de símbolos terminais; Rp é o conjunto de regras de produção; e Si, pertencente a Sn, é o símbolo inicial.

Por exemplo, em Levine, Drang e Edelson (1988), uma gramática livre de contexto foi criada para um fragmento do Português. Esta gramática é representada pelas regras expressas na tabela 2.2:

Tabela 2.2. Gramática livre de contexto, definida em (LEVINE; DRANG; EDELSON, 1988).

1 ^a	O	→	SN, SV
2 ^a	SN	→	DET, SUB
3 ^a	SN	→	SUB
4 ^a	SV	→	V, SN
5 ^a	SUB	→	menino banana
6 ^a	DET	→	o
7 ^a	V	→	comeu

A primeira regra desta gramática indica que uma oração (O) é composta de um

⁴Estudo dos fatos da linguagem e das leis naturais que a regulam. Livro onde se expõem as regras da linguagem (FERREIRA, 1975).

sintagma nominal (SN) e de um sintagma verbal (SV), representado por:

- $O \rightarrow SN, SV$

Na segunda regra, um sintagma nominal é dividido em um determinante e um substantivo ou, no caso da terceira regra, simplesmente um substantivo:

- $SN \rightarrow DET, SUB$
- $SN \rightarrow SUB$

Na quarta regra da gramática, um sintagma verbal é dividido em um verbo seguido de um sintagma nominal:

- $SV \rightarrow V, SN$

As quatro primeiras regras envolvem símbolos não terminais, formados pelos componentes gramaticais que podem estar presentes em uma frase da língua portuguesa. Já os símbolos terminais são formados pelas regras cinco, seis e sete, nas quais os símbolos não terminais localizados na parte esquerda da regra estão associados a palavras que realmente aparecem em frases da Língua Portuguesa, que aparecem na parcela direita da regra.

Por exemplo, a sentença “*O menino comeu banana*” pode ser analisada sintaticamente utilizando a gramática livre de contexto definida na tabela 2.2.

O primeiro passo é aplicar a 1ª regra, onde uma oração é derivada em um sintagma nominal e um sintagma verbal. O segundo passo é aplicar a 2ª regra pois, após a aplicação da 1ª regra da gramática, o símbolo não terminal SN foi mencionado. Ao aplicar a 2ª regra, o sintagma nominal é derivado em um determinante e um substantivo. Com a aplicação da 3ª regra, o símbolo não terminal SV é derivado em um verbo e em um sintagma nominal. E assim por diante, um passo de cada vez,

até que todos os símbolos não terminais sejam substituídos por símbolos terminais da gramática. A tabela 2.3 mostra, passo a passo, como as regras descritas na tabela 2.2 são aplicadas a uma oração, com o objetivo de saber se a oração está sintaticamente correta.

Tabela 2.3. Aplicação de uma gramática livre de contexto.

Passo 1	O
Passo 2	SN SV
Passo 3	DET SUB SV
Passo 4	DET SUB V SN
Passo 5	DET SUB V SUB
Passo 6	o SUB V SUB
Passo 7	o menino V SUB
Passo 8	o menino comeu SUB
Passo 9	o menino comeu banana

Uma outra forma de visualizar os passos para a análise sintática da mesma frase - utilizando uma gramática livre de contexto - pode ser observada na figura 2.1. Esta forma de visualização é chamada de árvore de derivação, onde cada nível da árvore é definido pela derivação dos símbolos não terminais da gramática, com a finalidade de substituí-los por outros símbolos não terminais até derivar em símbolos terminais. A árvore de derivação da gramática utilizada é encerrada quando todos os símbolos não terminais encontrados são derivados em símbolos terminais. O primeiro nível da árvore é formado pelo símbolo inicial da gramática e o último nível abrange os símbolos terminais da gramática.

2.3.2 Análises Top-Down e Bottom-Up

Existem duas abordagens para analisar uma oração a fim de se obter uma árvore gramatical: as análises *Top-Down* e *Bottom-Up*. Nestas análises, todas as palavras da sentença têm sua representação na gramática.

Top-Down - Na análise *Top-Down*, as regras da gramática são aplicadas a partir do símbolo inicial até que os símbolos terminais da gramática correspondam às

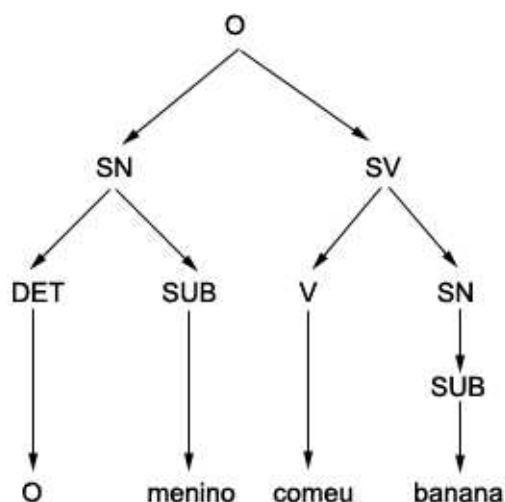


Figura 2.1. Estrutura em árvore de análise gramatical de uma sentença (LEVINE; DRANG; EDELSON, 1988).

palavras da oração analisada na árvore (KNIGHT; RICH, 1993).

Utilizando a gramática definida na tabela 2.4, é possível aplicar a abordagem *Top-Down* à oração “*João foi ao clube.*” e obter a estrutura em forma de árvore mostrada na figura 2.2.

Tabela 2.4. Regras de uma gramática (LEVINE; DRANG; EDELSON, 1988).

O	→	SN, SV
SN	→	DET, SUB
SN	→	DET
SV	→	V, SN

Em Pereira (2006), há um exemplo de derivação da abordagem *Top-Down* que define uma gramática para um fragmento da Língua Portuguesa, bem como um exemplo de aplicação da abordagem. A gramática que contém o fragmento do Português é mostrada na tabela 2.5. A oração “*O gato caçou o rato.*” constitui um exemplo de aplicação para as derivações desta gramática. As derivações da gramática da tabela 2.5 podem ser observadas na tabela 2.6.

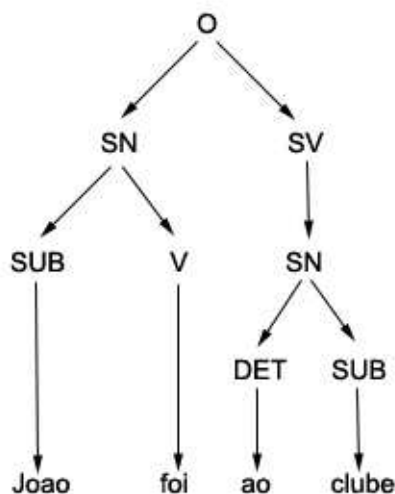


Figura 2.2. Estrutura em forma de árvore a partir da análise *Top-Down*.

Tabela 2.5. Gramática 2.

Oração	→	Sujeito Predicado
Sujeito	→	Artigo Substantivo predicado
Predicado	→	Verbo Artigo Substantivo
Artigo	→	o
Substantivo	→	gato rato
Verbo	→	caçou

Oração: “O gato caçou o rato”

Análise *Top-Down*:

Bottom-Up Na análise *Bottom-Up*, as regras da gramática são aplicadas de trás para a frente, ou seja, as regras são aplicadas dos símbolos terminais até o símbolo raiz. A análise *Bottom-Up* termina quando todas as palavras da oração são os símbolos terminais da gramática cujo nó superior é o símbolo raiz da gramática. As estratégias *Bottom-Up* identificam bem as relações lingüísticas das palavras da entrada e produzem estruturas conceituais destas relações. Além disso, esta abordagem realiza análises e interpretações semânticas mais exatas, assim como detecta inconsistências e lacunas léxicas (RAU; JACOBS, 1988).

Tabela 2.6. Aplicação da análise *Top-Down*.

Oração
→ Sujeito Predicado
→ Artigo Substantivo Predicado
→ o Substantivo Predicado
→ o gato Predicado
→ o gato Verbo Artigo Substantivo
→ o gato caçou Artigo Substantivo
→ o gato caçou o Substantivo
→ o gato caçou o rato

A seguir, na tabela 2.7, um exemplo da análise utilizando a abordagem *Bottom-Up* é mostrado, utilizando a gramática definida na tabela 2.5.

Frase: “O gato caçou o rato”

Análise *Bottom-Up*:

Tabela 2.7. Aplicação da análise *Bottom-Up*.

o gato caçou o rato
→ Artigo gato caçou o rato
→ Artigo Substantivo caçou o rato
→ Sujeito caçou o rato
→ Sujeito Verbo o rato
→ Sujeito Verbo Artigo rato
→ Sujeito Verbo Artigo Substantivo
→ Sujeito Predicado
→ Frase

2.4 ANÁLISE SEMÂNTICA

A análise semântica é o processo pelo qual o computador tenta entender o significado de uma sentença que foi dividida na análise sintática (LEVINE; DRANG; EDELSON, 1988).

A principal finalidade do processamento semântico é a criação da representação do

significado de uma frase. Porém, um outro papel importante é a imposição de restrições sobre a representação que pode ser construída, devido ao fato de ser necessário haver conexões estruturais entre a estrutura sintática e a semântica (KNIGHT; RICH, 1993).

A área da semântica é um segmento de estudo mais nebuloso do que a sintaxe por apresentar questões que são difíceis de tratar de maneira exata e completa. Esta diferença ocorre porque na análise semântica o significado das palavras está ligado ao conhecimento de mundo e às questões mais obscuras, como estados mentais e consciência (VIEIRA; LIMA, 2001).

A análise semântica lida com ambigüidades formal-ortográficas encontradas em sentenças, podendo ser ambigüidades lexicais e sintáticas (VIEIRA; LIMA, 2001). A ambigüidade lexical ocorre quando uma palavra tem mais de um significado. Um exemplo de ambigüidade lexical é visto nas seguintes frases:

- Maria está sentada no *banco* da praça.
- João pagou a conta de telefone no *banco*.

A palavra “*banco*” tem significados distintos em cada frase. Na primeira frase, a palavra *banco* significa um assento estreito e comprido, que pode apresentar inúmeras formas, qualidades e materiais, enquanto na segunda frase a mesma palavra significa uma instituição financeira que realiza diversas transações monetárias. Ao analisar a sentença como um todo e não apenas palavra a palavra, os casos em que ocorrem ambigüidade lexical são resolvidos mais facilmente, pois as palavras ao seu redor ajudam a identificar o significado correto.

A ambigüidade sintática ocorre quando uma frase apresenta mais de uma interpretação possível, como é o caso da seguinte oração:

- Todas as meninas gostam de uma boneca.

Ao analisar esta oração é possível obter duas interpretações. Na primeira inter-

pretação, cada menina gosta de uma boneca diferente, enquanto na segunda interpretação existe uma única boneca da qual todas as meninas gostam.

Para que o computador possa encontrar o significado de uma oração, é necessário estabelecer um conjunto de regras com a finalidade de que o computador possa aplicá-las a uma oração e obter seu significado. Por exemplo, para obter o significado da oração

- “O menino comeu banana.”,

o analisador semântico definido em Levine, Drang e Edelson (1988) utilizou o seguinte conjunto de regras:

Regra 1 Se um determinante constituir a primeira parte de uma sentença e for seguido por um substantivo, então o substantivo é conhecido como sujeito.

Regra 2 Se um verbo segue o sujeito, então o verbo expressa a ação que o sujeito executa.

Regra 3 Se um substantivo está localizado após um sujeito e um verbo, então o substantivo é um objeto.

Regra 4 Se uma sentença tem a forma de sujeito, verbo e objeto, então sabe-se qual foi a ação do sujeito em relação ao objeto.

Com estas regras, o computador pode obter informações a respeito da ação do menino com relação à banana e armazenar estas informações sob forma de conhecimento. Existem diversas abordagens para a criação da representação semântica de uma frase, dentre as quais está incluída a gramática semântica, demonstrada a seguir:

2.4.1 Gramática Semântica

Uma gramática semântica é uma gramática livre de contexto, onde funções semânticas e sintáticas escolhem os símbolos não-terminais e regras de produção. O resultado da análise, utilizando a gramática semântica, é o significado da oração (KNIGHT; RICH, 1993).

Em Knight e Rich (1993), demonstra-se uma gramática semântica específica para um contexto, utilizada em uma parte do sistema operacional para reconhecer comandos em linguagem natural. Esta gramática é mostrada na figura 2.3.

```
S → qual é PROPRIEDADE-DO-ARQUIVO de ARQUIVO?
    {consultar ARQUIVO.PROPRIEDADE-DO-ARQUIVO}
S → Eu quero AÇÃO
    {comando AÇÃO}
PROPRIEDADE-DO-ARQUIVO → o PROP-ARQ
    {PROP-ARQ}
PROP-ARQ → extensão | proteção | data de criação | proprietário
    {valor}
ARQUIVO → NOME-DO-ARQUIVO | ARQUIVO1
    {valor}
ARQUIVO1 → ARQUIVO2 do USUÁRIO
    {ARQUIVO2.proprietário: USUÁRIO}
ARQUIVO1 → ARQUIVO2
    {ARQUIVO2}
ARQUIVO2 → arquivo EXT
    {instância: estrut-arq
     extensão: EXT}
EXT → .init | .txt | .lsp | .for | .ps | .mss
    valor
AÇÃO → imprimir ARQUIVO
    {instância: imprimir
     objeto: ARQUIVO}
AÇÃO → imprimir ARQUIVO na IMPRESSORA
    {instância: imprimir
     objeto: ARQUIVO
     impressora: IMPRESSORA}
USUÁRIO → Mário | Susana
    {valor}
```

Figura 2.3. Exemplo de gramática semântica (KNIGHT; RICH, 1993).

Ao utilizar a gramática semântica da figura 2.3 na frase

- “Eu quero imprimir o arquivo .init do Mário.”,

obtém-se a estrutura de árvore demonstrada na figura 2.4, onde é possível identificar qual ação o sistema operacional deve tomar em relação à frase de entrada.

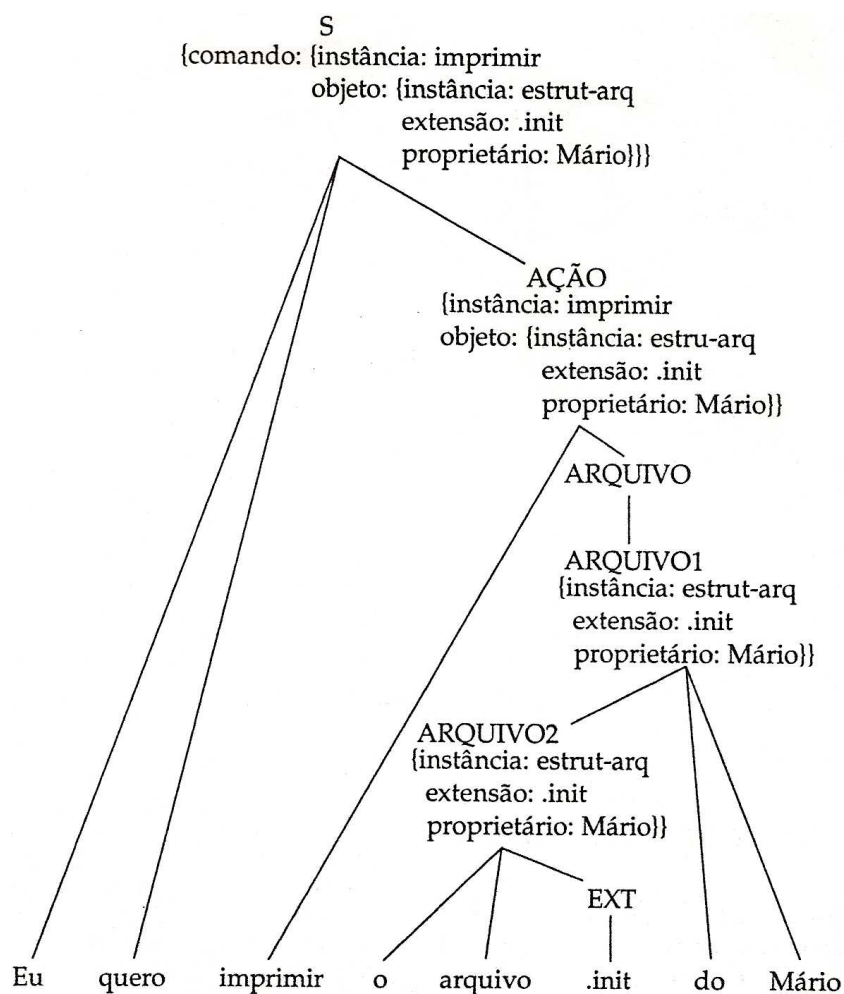


Figura 2.4. Resultado da análise com uma gramática semântica (KNIGHT; RICH, 1993).

2.5 ANÁLISE PRAGMÁTICA

A análise pragmática tem por objetivo encontrar o real significado de uma oração. Uma dificuldade típica encontrada na análise pragmática é a ocorrência de referência anafórica, que consiste em citar um elemento anteriormente expresso por meio de um pronome (KNIGHT; RICH, 1993). As orações a seguir são exemplos de referência anafórica:

- João comprou um livro de ficção científica.
- *Ele* está bastante empolgado.

Ao analisar as orações, é possível observar que na segunda oração o pronome pessoal “*Ele*” se refere a *João*, citado na primeira frase. Sendo assim, a leitura que pode ser feita com o conhecimento desta referência é que João está bastante empolgado com o livro de ficção científica que comprou.

Uma outra forma de referência a palavras citadas no texto é a ocorrência de cataforas. A referência catafórica ocorre quando um termo se refere a outro que vem adiante e fornece o seu significado na sentença. Por exemplo, considere a seguinte oração:

- *Esta* foi sempre a minha doutrina: *tudo que há de bom e útil no mundo, se consegue procedendo por amor ao próximo.*

Analisando a oração acima, é possível observar que o pronome demonstrativo “*Esta*” faz referência à frase “*tudo que há de bom e útil no mundo, se consegue procedendo por amor ao próximo*”, que ainda está por vir na sentença.

Segundo Vieira e Lima (2001), no segmento da pragmática, questões ligadas ao uso da linguagem são estudadas abordando aquilo que é relativo a quem usa as expressões e ao contexto de uso.

Em Harris (1988), a pragmática é definida como o conjunto dos aspectos do discurso em linguagem natural que não estão explicitamente contidos nas palavras usadas no texto, e constitui uma ferramenta útil quando o significado da sentença está fundido no contexto do discurso ou no conhecimento do mundo. O aspecto mais importante para fazer um sistema em linguagem natural é poder contar com análises inteligentes.

A pragmática verifica se o significado associado a uma estrutura sintática é realmente o significado mais apropriado no contexto considerado. Por exemplo, na frase “um animal perseguiu/capturou outro animal”, no contexto predador-presa, as palavras “perseguiu/capturou” podem ser entendidas como “comeu”, transmitindo o conhecimento que um animal comeu o outro (PEREIRA, 2006).

Conclusão

A utilização do Processamento de Linguagem Natural no desenvolvimento da interface com máquinas requer muito conhecimento sobre diversos aspectos da Língua Portuguesa como, por exemplo, a formação das palavras, os aspectos gramaticais relacionados à formação de orações e o modo de captar o significado de uma oração analisando as palavras que a compõem. As análises necessárias para o PLN devem considerar, inclusive, o contexto no qual a oração está inserida.

A complexidade do processo de dotar o computador de conhecimento para compreender a Língua Portuguesa constitui um fator limitante para o uso desta técnica.

DESCOBERTA DE CONHECIMENTO

Considerando a complexidade e as restrições para a aplicação de técnicas de PLN, este capítulo aborda a Descoberta de Conhecimento em Textos (KDT) como uma possível alternativa para a proposição de interfaces de comunicação. Este capítulo discorre sobre os detalhes teóricos da Descoberta de Conhecimento em Textos e, no final do capítulo, exemplos de aplicação são apresentados.

Atualmente, a maioria das empresas possui um sistema computacional para armazenar e processar dados importantes da sua organização. Bases de dados de empresas contêm uma potencial fonte de informação; porém, tais dados raramente se traduzem em informação devido à falta de ferramentas apropriadas para a sua extração. O grande volume de dados armazenados extrapola, em muito, a capacidade do ser humano de analisar e extrair relações significativas entre eles sem o uso de um *software* para auxiliar na análise dos dados (REZENDE, 2003). Assim, a maioria das organizações geralmente não consegue aproveitar à totalidade o que está armazenado em seus bancos de dados. Diversas informações valiosas também não podem ser obtidas utilizando os Sistemas de Gerenciamento de Banco de Dados (SGBDs) convencionais.

A área de Descoberta de Conhecimento surgiu a partir dos estudos sobre Inteligência Artificial e preocupa-se não só com a descoberta de conhecimento, mas também com a descoberta de formas de aquisição e armazenamento deste conhecimento (WIVES; LOH, 1999). Com o advento dos Sistemas de Gerenciamento de Bancos de Dados, os pesquisadores em Bancos de Dados da área de Sistemas de Informação passaram a pesquisar novas aplicações, na busca de informações implícitas que também pudessem ser úteis nestes bancos de dados. Com isso, surgiram os primeiros sistemas

de análise de dados e de mineração de dados relacionais, dando início às pesquisas em áreas como a de Mineração de Dados (*Data Mining*), resultando em ferramentas para Processamento Analítico de Dados e Armazéns de Dados ou *Data Warehouses*.

Em Gimenes (2000), *Data Warehouse* é definido como um repositório de informações para suportar decisões. Neste repositório, dados são coletados a partir de diversas aplicações de uma organização, integrados em áreas lógicas de assuntos dos negócios e informações são armazenadas de forma acessível e compreensível a pessoas não técnicas, perfil comum entre as pessoas responsáveis pelas tomadas de decisão.

3.1 DADO, INFORMAÇÃO E CONHECIMENTO

Segundo Rezende (2003), os conceitos de dado, informação e conhecimento estão interligados. A figura 3.1 mostra uma representação gráfica do relacionamento existente entre estes conceitos, em função da capacidade de entendimento e da independência de contexto.

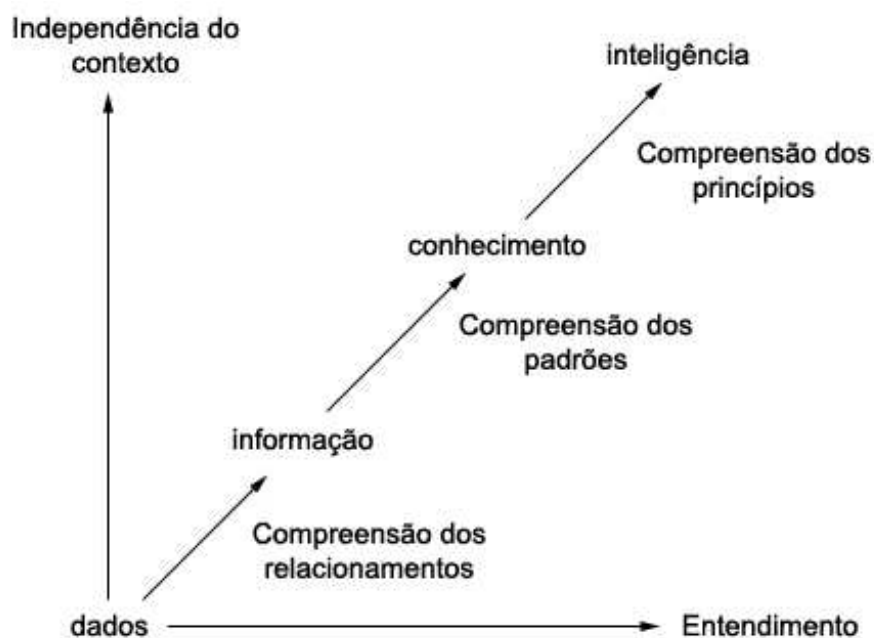


Figura 3.1. Dado, Informação, Conhecimento e Inteligência (REZENDE, 2003).

O dado é formado por um conjunto de símbolos, letras e números que podem ser processados pelo computador e, por si só, não oferece embasamento para o entendimento da situação (REZENDE, 2003). Um exemplo de dado é *Temperatura = 40° C*, pois este não oferece nenhum tipo informação. Em Fedeli, Pollini e Peres (2003), dado é o elemento identificado em sua forma bruta que por si só não conduz a uma compreensão de um fato ou uma situação.

A informação, segundo Rezende (2003), é o dado analisado e contextualizado, envolvendo a interpretação de um conjunto de dados. A informação é o significado atribuído aos dados a partir de uma análise. Como por exemplo, a partir dos dados *Temperatura = 40° C* e *Paciente = Maria*, pode-se extrair a informação que Maria está com febre. Segundo Fedeli, Pollini e Peres (2003), informação é o dado trabalhado, que permite sua utilização numa tomada de decisão.

Conhecimento é a habilidade de criar um modelo que descreva o objeto e indique as ações a serem implementadas e as decisões a serem tomadas. O conhecimento gerado a partir de dados de uma organização pode ser utilizado para o planejamento e para a tomada de decisões (REZENDE, 2003).

Supermercados, por exemplo, possuem dados referentes a compras de produtos efetuadas por clientes. A utilização de um processo de descoberta de conhecimento sobre os dados armazenados pode extrair perfis de clientes e de compras. Após análises, pode-se obter informações sobre, por exemplo, quais produtos em geral são comprados juntamente com outros, em quais dias da semana o estabelecimento recebe mais clientes, quais produtos são mais vendidos e outros tipos de informações. A análise das informações obtidas pode levar o estabelecimento a fazer promoções de produtos em alguns dias da semana, colocar produtos perto de outros com bons índices de venda, aumentar e reduzir o preço de alguns itens com base na compra dos clientes.

3.2 DESCOBERTA REATIVA X PROATIVA

De acordo com Wives, Loh e Oliveira (2000) e Loh (2001a), existem dois modos para aquisição de informação: o modo reativo e o modo proativo. No primeiro caso, a informação é adquirida para resolver um problema específico do usuário. Neste caso, o usuário sabe o que quer e poderá identificar a solução para o problema quando a encontrar. É o caso típico de quando se deseja encontrar uma informação específica, como, por exemplo, um valor para um atributo ou um processo para resolver um problema. Uma aplicação de descoberta reativa pode ser encontrar atributos comuns nos produtos mais vendidos, encontrar motivos que levam à evasão ou a reclamações de clientes, encontrar clientes potenciais para propaganda seletiva.

No modo proativo, o propósito de adquirir informação é exploratório e com o objetivo de detectar problemas potenciais ou oportunidades Loh (2001a). Neste segundo caso, o usuário não tem um objetivo específico. Isto quer dizer que há uma necessidade ou objetivo, mas o usuário não consegue definir o que precisa para resolver o problema. A necessidade típica do modo proativo poderia ser representada pela expressão *“diga-me o que há de interessante nesta coleção”*. Neste caso, o usuário não sabe de forma definida do que lhe é de interesse, podendo tal interesse mudar durante o processo. Pode-se dizer que se trata um processo exploratório, sendo, em geral, iterativo tendo retroalimentação e interativo, com ativa participação e intervenção do usuário.

Na abordagem proativa, não há hipóteses iniciais ou elas são muito vagas. O usuário deverá descobrir hipóteses para a solução do seu problema e explorá-las, investigá-las e testá-las durante o processo. Em geral, isto acontece porque o usuário não sabe exatamente o que está procurando. É o caso típico de quando se quer monitorar alguma situação ou encontrar algo de interesse e que possa levar a investigações posteriores. Depois que hipóteses são levantadas, o processo pode seguir como no paradigma reativo, usualmente sendo necessário avaliar as hipóteses para verificar se são verdadeiras ou não.

A seguir serão discutidas a Descoberta de Conhecimento em Base de Dados e a Descoberta de Conhecimento em Textos.

3.3 DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS

A Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Databases*), também conhecida como KDD, é um processo que envolve a automação da identificação e do reconhecimento de padrões em um banco de dados (GIMENES, 2000). Segundo Sferra e Corrêa (2003), KDD é a extração, a partir de dados, de conhecimento implícito, previamente desconhecido e potencialmente útil.

Conforme visto em Rezende (2003) e Sferra e Corrêa (2003), alguns autores consideram *Data Mining* e KDD como termos sinônimos, enquanto outros consideram *Data Mining* como uma etapa de KDD e a mais importante de todo o processo. De fato, *Data Mining* é a etapa principal do processo de KDD, onde são aplicadas técnicas de mineração nos dados pré-processados com o propósito de transformar dados armazenados em conhecimento. Em Gimenes (2000) *Data Mining* é definido como uma tecnologia usada para revelar informação estratégica escondida em grandes massas de dados.

O objetivo da KDD é transformar dados armazenados em conhecimento, tratando-se de um processo de análise expresso por regras e relações entre dados (REZENDE, 2003).

3.3.1 Etapas do processo de KDD

O processo de KDD é definido, em Gimenes (2000), como um conjunto de atividades contínuas que compartilham o conhecimento descoberto a partir de bases de dados. Este conjunto é formado por: Seleção de dados, Pré-processamento e limpeza, Transformação, *Data Mining* e Interpretação.

A figura 3.2 demonstra as etapas da KDD (GIMENES, 2000), onde pode-se observar que o retorno a etapas anteriores é possível caso alguma anormalidade seja identificada.



Figura 3.2. Etapas do processo de KDD.

Seleção de dados

Nesta etapa, o domínio sobre o qual se pretende executar o processo de descoberta é definido. Neste domínio, o conjunto de dados que será utilizado ao longo do processo é coletado.

Pré-processamento e limpeza

Os dados inconsistentes são tratados durante esta etapa e também são estabelecidas as estratégias para resolver os problemas de ausência de dados. As causas que levam à situação de ausência de dados são a não disponibilidade do dado ou a inexistência do mesmo.

Transformação

Os algoritmos de mineração normalmente não podem acessar os dados em seu formato nativo, seja em razão da forma como são armazenados, seja pela normalização adotada na modelagem do banco. É necessária a conversão dos dados para um formato apropriado com o objetivo de facilitar o uso das técnicas de *Data Mining* (SFERRA;

CORRÊA, 2003). Estes dados são armazenados em *Data Warehouses*, pois neles os dados são não-voláteis, e geralmente já permanecem organizados por assunto.

Data Mining

Segundo Sferra e Corrêa (2003), o objetivo principal dessa etapa é a aplicação de técnicas de mineração aos dados pré-processados, o que envolve ajuste de modelos e/ou determinação de características dos dados. Gimenes (2000) afirma que a maioria dos métodos de *Data Mining* são baseados em conceitos de aprendizagem de máquina, reconhecimento de padrões, estatística, classificação, clusterização e modelos gráficos.

Interpretação

Segundo Sferra e Corrêa (2003), a informação extraída na etapa de *Data Mining* é analisada com relação ao objetivo proposto, e as melhores informações são identificadas e apresentadas. Se após a etapa de *Data Mining* for verificada a necessidade de um dado que não havia sido previsto anteriormente, o processo de KDD pode retornar para alguma etapa anterior (GIMENES, 2000).

3.3.2 Métodos tradicionais de Data Mining

A etapa de *Data Mining*, conforme visto anteriormente, é considerada o estágio mais importante do processo de descoberta de conhecimento, pois ela define o processo automatizado de captura e análise de grandes conjuntos de dados para extrair um significado. Esta etapa é utilizada tanto para descrever características do passado como para prever tendências para o futuro (SFERRA; CORRÊA, 2003).

A seguir são apresentados os métodos tradicionais de *Data Mining*.

Classificação

Este método classifica ou associa um item a uma ou a várias classes categóricas pré-definidas (SFERRA; CORRÊA, 2003). Gimenes (2000) define este método como uma técnica que consiste na aplicação de um conjunto de exemplos pré-classificados para desenvolver um modelo capaz de classificar uma população maior de registros.

Um algoritmo classificador desenvolvido de forma eficiente pode ser usado de forma preditiva para classificar novos registros nas classes pré-definidas (GIMENES, 2000). Gimenes (2000) exemplifica um classificador que pode ser treinado para identificar empréstimos arriscados, onde informações cadastrais de milhares de interessados são analisadas, recurso que pode ser usado como suporte para a decisão no momento de conceder um empréstimo a alguém.

Análise de Agrupamento

Neste método, grupos de registros correlatos são identificados, os quais serão usados no decorrer do processo (GIMENES, 2000). A Análise de Agrupamento, também conhecida por “*clusterização*” - do inglês *cluster* -, detecta a existência de diferentes grupos dentro de um determinado conjunto de dados e os separa em classes, com base em medidas de similaridade ou modelos probabilísticos (SFERRA; CORRÊA, 2003).

As classes são criadas, nesta técnica, a partir dos dados analisados, onde dados semelhantes são reunidos na mesma classe de forma dinâmica. Diferentemente da Análise de Agrupamento, na técnica de Classificação, as classes são pré-definidas e os dados analisados são classificados em uma destas classes.

Sumarização

A Sumarização é a abstração ou generalização dos dados, onde um conjunto de dados é analisado, resultando em um pequeno grupo de dados que apresenta a visão geral do conjunto analisado (FU, 1997). A sumarização é freqüentemente usada na análise exploratória de dados com geração automatizada de relatórios, sendo responsável pela descrição compacta de um conjunto de dados (SFERRA; CORRÊA, 2003).

Segundo Fu (1997), a sumarização pode ter diferentes níveis de abstração, assim como pode ser vista de diferentes ângulos, o que pode revelar padrões nos dados. Em Fu (1997), um exemplo apresentado é uma empresa de telefonia, que tem a possibilidade de enviar as despesas de ligações contabilizadas por semana, mês, ano, como também por ligações locais, ligações para diferentes estados ou países.

Associação

Esta técnica consiste na descoberta de uniões ou conexões entre objetos, que são expressas por regras de associação. A presença de um conjunto de objetos que compõe uma regra é relacionada à presença de um outro conjunto de objetos, onde os objetos são os dados (FU, 1997). Uma regra de associação é definida como “se X , então Y ”, sendo representada por $X \Rightarrow Y$, onde X e Y são conjuntos de objetos que não possuem interseções, com X sendo o antecedente da regra e Y , o seu conseqüente (SFERRA; CORRÊA, 2003). Segundo Loh et al. (2001b) e Sferra e Corrêa (2003), se o antecedente da regra for encontrado na coleção de dados, então seu conseqüente também está na coleção com certo grau de certeza, definido por valores de confiança e suporte. O grau de confiança é a proporção de textos onde os dados de X e de Y estão presentes em relação ao número de textos que possuem apenas os dados de X . O valor de suporte da mesma regra é o número de documentos onde os dados de X e de Y estão presentes. O grau de confiança permite predizer, com certo grau de certeza,

a presença de um conjunto de dados em função de um outro.

As associações entre conjuntos de dados auxiliam na tomada de decisão, pois permitem análises e conclusões sobre as relações entre os dados. Gimenes (2000) exemplifica o uso de regras de associação em projetos de catálogos, projetos de leiaute de lojas e prateleiras, onde itens podem ser agrupados de modo a induzir a venda de artigos relacionados. Outro exemplo é fornecido por Fu (1997), onde afirma que regras de associação podem ser usadas para descobrir que quando os clientes compram um determinado produto de lojas, geralmente levam um outro produto. Com esta informação, as lojas podem colocar estes produtos mais próximos entre si, com o intuito de induzir sua compra.

3.3.3 Áreas de Aplicações

O processo de Descoberta de Conhecimento em Base de Dados pode ser aplicado em diversas áreas para auxiliar na tomada de decisão e na extração de conhecimento. Hoje em dia, a maioria das organizações possui um grande volume de dados que são gerados a partir das suas atividades cotidianas. A análise destes dados pode trazer muitas informações que antes não eram conhecidas, auxiliando na tomada de decisão e aumentando o conhecimento a respeito da origem dos dados.

Estão listados a seguir alguns exemplos de possíveis aplicações para o processo de KDD:

- **Bancária:** aprovação de crédito;
- **Ciências e Medicina:** descoberta de hipóteses, diagnóstico, classificação e predição;
- **Comerciais:** segmentação, localização de consumidores e identificação de hábitos de consumo;

- **Engenharia:** simulação e análise, reconhecimento de padrões, processamento de sinais e planejamento;
- **Financeira:** apoio para investimentos e controle de carteira de ações;
- **Gerencial:** tomadas de decisão e gerenciamento de documentos;
- **Manufatura:** modelagem e controle de processos, controle de qualidade e alocação de recursos;
- **Segurança:** detecção de bombas, icebergs e fraudes;

3.4 DESCOBERTA DE CONHECIMENTO EM TEXTO

A área de Descoberta de Conhecimento em Texto, também conhecida como KDT (*Knowledge Discovery from Text*), surgiu da necessidade de manipular informações informais e textuais, pois os sistemas de descoberta de conhecimento existentes foram concebidos para manipular dados estruturados (PERRIN; PETRY, 2003). Segundo Feldman e Hirsh (1997) e Tan (1999), esta área pode ser entendida como a aplicação de técnicas de KDD sobre dados extraídos de textos.

Sferra e Corrêa (2003) define KDD como um processo de extração não-trivial de informação implícita, previamente desconhecida e potencialmente útil para o usuário. Sendo assim, KDT pode ser definida como o processo de extração não-trivial de informação implícita, previamente desconhecida e potencialmente útil, contida em textos Loh (2001a).

A Descoberta de Conhecimento em Textos, também conhecida como *Text Mining* (mineração em texto), é o processo de extrair padrões interessantes e não-triviais ou conhecimento de documentos textuais sem estrutura (TAN, 1999). Este processo inclui técnicas e ferramentas inteligentes e automáticas para auxiliar pessoas a analisar grandes volumes de dados, com a finalidade de garimpar conhecimento útil para uma determinada situação ou domínio.

Tan (1999) apresenta um método para a Descoberta de Conhecimento em Texto, que consiste em duas etapas: a primeira etapa, (*Text refining*), tem a finalidade de transformar os documentos que possuem textos livres em formas intermediárias; a segunda etapa, (*knowledge distillation*), tem como objetivo a busca de padrões e conhecimento sobre as formas intermediárias resultantes da primeira etapa.

Segundo Tan (1999), as formas intermediárias podem ser semi-estruturadas, como a representação de grafos de conceitos, ou estruturadas, como a representação de relações de dados. As formas intermediárias também podem ter como base o documento - onde cada entidade representa um documento -, ou o conceito - onde cada entidade representa um objeto ou conceito de interesse em um domínio específico. A mineração de informação com base em documentos procura padrões e relações entre os documentos, com a possibilidade de utilizar técnicas de agrupamento e categorização, enquanto a mineração com base em conceitos procura padrões e relações entre os objetos e conceitos, utilizando, por exemplo, modelagem preditiva e descoberta associativa.

A figura 3.3 mostra o método proposto em Tan (1999), no qual é possível observar que uma forma intermediária baseada em documento pode ser transformada na forma intermediária baseada em conceito. Para tanto, é necessário extrair informações relevantes de um determinado domínio. Tan (1999) comenta que a forma intermediária com base em documento geralmente é independente do domínio, enquanto a forma com base em conceito é dependente do domínio.

3.4.1 Etapas do processo de KDT

Wives (2002) comenta que, pelo fato da KDT ser uma área de pesquisa recente, ainda não existem livros que tratem especificamente sobre Descoberta de Conhecimento em Texto. Para identificar um guia para o processo de KDT, Wives fez uma análise técnica de artigos que abordavam o tema KDT e identificou que este guia é análogo ao processo de KDD; porém, ele também indica que nem todas as etapas

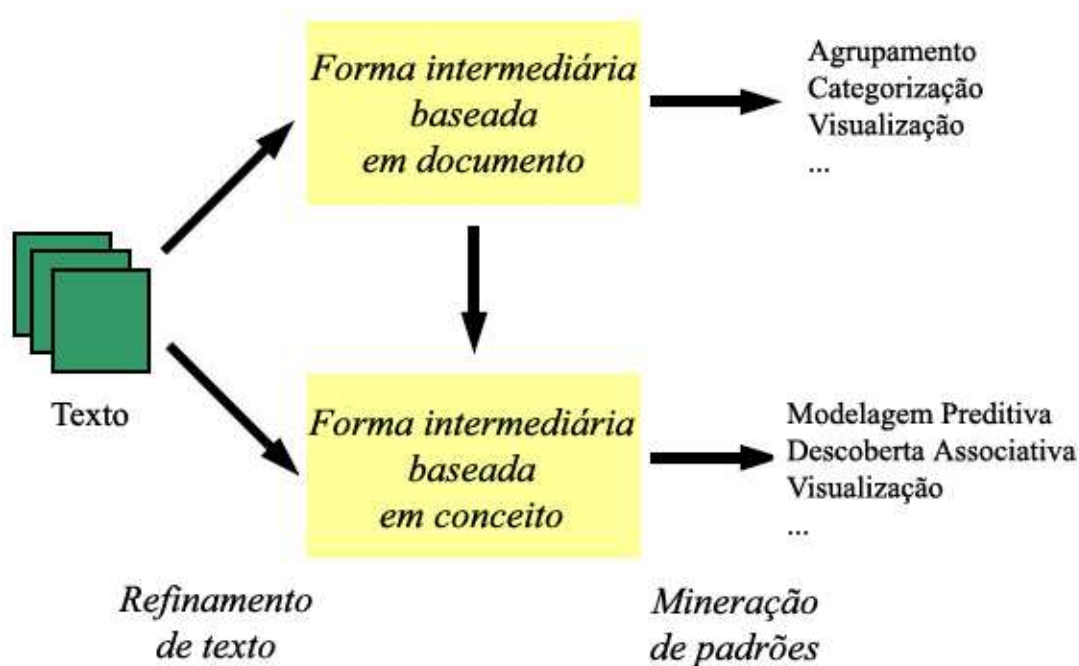


Figura 3.3. Método para KDT (TAN, 1999).

identificadas no seu guia precisam ser seguidas e que outras etapas podem ser criadas. Estas etapas são: definição de objetivos, seleção de um subconjunto de dados, pré-processamento ou limpeza dos dados, redução ou projeção dos dados, escolha da técnica, mineração, interpretação dos resultados e consolidação do conhecimento descoberto, e são descritos a seguir.

Definição de objetivos

Compreensão do domínio, identificação do que deve ou pode ser descoberto com base no conhecimento do domínio.

Seleção de um subconjunto de dados

Nem todas as informações disponíveis devem ser utilizadas, pois a utilização de

muitas informações pode influenciar negativamente no resultado de uma descoberta. Além disso, um processo totalmente inclusivo pode tornar a descoberta de conhecimento mais demorada.

Pré-processamento ou limpeza dos dados

Nesta etapa, os dados são preparados e os indesejáveis são removidos. Diversos métodos podem e devem ser aplicados, incluindo métodos de limpeza de caracteres indesejados, revisão ortográfica e morfológica e até análise semântica e normalização de vocabulário. Nesta etapa da KDT, alguns estágios da compreensão de Linguagem Natural, apresentados no capítulo 2, podem ser aplicados para a limpeza dos dados como, por exemplo, as análises sintática e semântica, segregando orações que não estejam corretamente estruturadas.

Redução ou projeção dos dados

Nem todas as palavras do texto são importantes e que, dependendo do objetivo da análise e do conhecimento do domínio no qual os textos estão inseridos, algumas partes de um documento ou conjunto de documentos podem ser mais importantes do que outras.

Escolha da técnica, método ou tarefa de mineração

Nesta etapa, a técnica de mineração para a descoberta de conhecimento é selecionada. Wives (2002) discute em seu trabalho algumas técnicas de descoberta de conhecimento.

Mineração

Nesta etapa, uma ou mais técnicas de mineração são aplicadas para a descoberta de conhecimento nos textos selecionados.

Interpretação dos resultados

Após a etapa de mineração, a análise dos resultados obtidos é realizada. Nesta fase, é possível retornar a etapas anteriores caso o conhecimento obtido não seja satisfatório.

Consolidação do conhecimento descoberto

Nesta etapa, a documentação ou a incorporação dos dados a um sistema é efetuada, podendo definir ou redefinir a estratégia de uma empresa (WIVES, 2002).

3.4.2 Estratégias Baseadas em Palavras e Baseadas em Conceitos

Os sistemas de extração de conhecimento geralmente utilizam estratégias de descoberta de conhecimento baseadas em termos ou palavras existentes nos textos e não em conceitos da realidade Loh et al. (2001b). Estratégias baseadas em palavras dificultam o entendimento do conhecimento descoberto. Por exemplo, em um processo de descoberta em textos médicos, resultados concluíram que o termo “*visual*” é muito comum em prontuários de pacientes com determinada doença mental. Entretanto, não é possível concluir se o termo se refere a “*deficiência visual*” ou a “*ilusão visual*”. Esta incerteza ocorre devido ao fato de que os termos não estão relacionados a conceitos da realidade ou a um conceito específico do domínio.

Outra limitação das técnicas baseadas em termos é o chamado “*problema do vocabulário*” Loh (2001a). Esse problema pode ocasionar erros semânticos devido aos sinônimos, às variações léxicas e aos chamados quase-sinônimos. Se o processo de análise dos textos for realizado sem considerar estes problemas, o processo de KDT pode levar a resultados incorretos. Alguns autores sugerem a análise dos termos sinônimos com o processo de KDT na tentativa de minimizar estes problemas Loh (2001a). Em Yang e Lee (2000), o processo de categorização de textos obteve melhor desempenho usando termos sinônimos, pois termos diferentes podem ser utilizados para descrever conceitos similares.

Loh (2001a) utiliza a abordagem baseada em conceitos para a extração de conhecimento em textos. Esta abordagem representa melhor que palavras, por exemplo, objetos, sentimentos e ações do mundo real. Cada conceito é definido através de uma ou mais regras para sua identificação. Esta abordagem é detalhada na seção 3.6

A utilização de um dicionário controlado é outra alternativa para melhorar o entendimento do conhecimento descoberto e minimizar o problema do vocabulário. Assim, os textos de um determinado domínio são analisados levando em consideração os termos encontrados neste domínio. Este método, porém, não ajuda necessariamente a resolver erros semânticos. Por exemplo, no caso de negação (“*paciente negador de cabeça*”), o vocabulário analisa em nível de documento a expressão (“*dor de cabeça*”), sendo que o correto é analisar no nível de conceitos do domínio, descobrindo informações mais próximas da realidade. Segundo Loh (2001a), é necessário elaborar metodologias que trabalhem no nível de conceitos, pois estas caracterizam melhor eventos, características, ações, objetos, etc., do que palavras.

3.5 MÉTODOS PARA KDT

A maioria das técnicas utilizadas no processo de KDT são oriundas do processo de KDD, especialmente quando dados são armazenados sob a forma de textos. Neste caso,

é possível aplicar técnicas de KDD em textos, desde que os textos sejam estruturados e tratados pelo método de KDT.

3.5.1 Recuperação de Informação

Segundo Loh (2001a), a Recuperação de Informação (RI) é a técnica mais utilizada nesta área. O objetivo é encontrar um documento ou um conjunto de documentos que contenham uma determinada informação textual (CLIFTON; STEINHEISER, 1998). Loh (2001a) apresenta o método de recuperação de passagens, no qual a técnica de RI é aplicada. Este método, porém, resulta em fragmentos de documentos que contêm a informação desejada.

3.5.2 Extração de Informação

O objetivo da extração de informação é obter tipos específicos de informação de um documento, geralmente usando palavras-chave nas buscas e nas técnicas estatísticas para recuperação de documentos relevantes (RILOFF; LEHNERT, 1994). Em Riloff e Lehnert (1994), a utilização da extração de informação é exemplificada em um domínio terrorista, do qual podem ser extraídos os nomes de todos envolvidos, vítimas, alvos e armas usadas no ataque terrorista. A principal vantagem deste método é que partes do texto que não são relevantes ao domínio podem ser ignoradas, fazendo com que o processamento seja menos custoso.

3.5.3 Agrupamento

Agrupamento é um método de aprendizado não supervisionado que separa um conjunto de padrões em grupos. Os métodos para agrupamento de KDD usam padrões numéricos para gerar os grupos de padrões. Em Runkler e Bezdek (2003), a técnica de agrupamento é utilizada em páginas da Web, onde as palavras do texto que constituem

dados não-numéricos são convertidos em dados numéricos. Para esta finalidade, tal técnica compara palavras à procura de similaridades entre letras que as compõem, gerando dados numéricos para a análise de agrupamento.

3.5.4 Classificação

O método de classificação, descrito na seção 3.3.2, associa um item a uma ou diversas classes categóricas pré-definidas.

Em Ladeira et al. (1996) é proposto um agente inteligente dotado de uma Rede Neural Auto-Organizada - conhecida como Rede SOM (*Self-Organizing Maps*), para classificação e extração de informação em documentos textuais, com a finalidade de atuar como moderador de listas de discussão de cursos a distância, de forma que a maior parte das perguntas seja respondida sem o intermédio do professor. O agente verifica uma base de conhecimento com respostas e perguntas já produzidas para responder à pergunta formulada pelo aluno.

Yang e Lee (2000) descreve a utilização de uma rede SOM para gerar automaticamente categorias de termos e estruturas. Em Honkela et al. (1996) é apresentado um modo de exploração e organização de páginas da Internet utilizando uma variação da Rede SOM chamada WEBSOM.

3.5.5 Associação

O método de Associação, descrito na seção 3.3.1, é a técnica que descobre uniões ou conexões entre objetos. De acordo com esta técnica, a presença de um conjunto de objetos é relacionada à presença de um outro conjunto de objetos.

Loh (2001a) utiliza regras associativas para descobrir relações entre conceitos encontrados em uma coleção de textos de psiquiatria, analisando características dos pacientes, tais como sintomas, sinais e seu comportamento social e familiar. A tabela

3.1 exibe a distribuição dos conceitos encontrados na coleção de textos de psiquiatria.

Tabela 3.1. Distribuições de Conceitos e regras associativas encontradas em texto de psiquiatria Loh (2001a).

Distribuições de Conceitos	Regras Associativas
familiares - 84,5%	casos de alcoolismo → inapetência (84%)
agressividade - 77%	casos de autismo → alteração de pensamento (95,3%)
inapetência - 76%	casos de agressividade → familiares (92,8%)
remédios - 74,5%	casos de depressão → insônia (85,1%)
insônia - 71%	
alteração de pensamento - 70,5%	
nervosismo - 68,5%	
alteração de atenção - 54,5%	

Após a análise da tabela 3.1, Loh (2001a) identificou o perfil do paciente típico que é atendido na clínica. É possível observar a presença de conceitos relacionados a familiares, agressividade, falta de apetite (inapetência), insônia, dentre outros. Utilizando as regras associativas formadas a partir destes conceitos, pode-se prever ou inferir características com base na presença de outras. É possível notar que os pacientes apresentaram falta de apetite em 84% dos casos com características de alcoolismo e, em 92,8% dos casos de agressividade, a presença do conceito “familiares” foi observada.

3.5.6 Sumarização

Este método tem por objetivo extrair resumos de um texto ou de uma coleção; estes resumos podem constituir uma visão geral ou as partes mais importantes ou mais interessantes (WIVES; LOH; OLIVEIRA, 2000). Em Mallett, Elding e Nascimento (2004), a sumarização é definida como um processo de condensação de um texto preservando seu conteúdo de informação e sua legibilidade. Segundo Mallett, Elding e Nascimento (2004), a extração de resumos de texto pode ser empregada em níveis de palavras-chave, sentenças e parágrafos. Em geral, a maioria das pesquisas usa o nível de sentença, pois a legibilidade do nível de palavras-chave é baixo e o nível de parágrafos apresenta pouca probabilidade de cobrir as informações de um documento,

dados o espaço reservado ao resumo em um texto.

3.5.7 Outros métodos

Um outro método para a KDT é visto em Neto e Moraes (2002), que utiliza formalismos adaptativos - em particular os autônomos adaptativos - para a resolução de alguns dos problemas tipicamente encontrados na representação e no processamento de linguagens naturais. Com os autônomos adaptativos, as ambigüidades ¹ e os não-determinismos ² são tratados de maneira a possibilitar construções sintáticas válidas.

Em Hahn e Homacker (1999), um sistema chamado SynDiKaTe foi desenvolvido para a aquisição automática de conhecimento em textos, no qual a extração e a representação do conhecimento tem como base grafos compostos por hierarquia de classes de palavras, como verbo, pronome, preposição, substantivo, etc. O SynDiKaTe é composto por duas camadas. A primeira contém os elementos que compõem o núcleo essencial para a compreensão da linguagem natural, como gramática, conhecimento do domínio, interpretação semântica e rotinas de inferência para a análise no nível de sentença. A segunda camada contém os mecanismos para operar na análise e compreensão dos textos, utilizando os dados da análise da sentença.

3.6 DESCOBERTA DE CONHECIMENTO EM TEXTO BASEADO EM CONCEITOS

A maioria das técnicas de extração de conhecimento tem como base termos ou palavras-chave. Estas técnicas apresentam uma limitação grave no vocabulário, pois podem ocorrer sinônimos, mesma palavra com mais de um significado, variações léxicas, como uso de radicais, conjugações verbais, variações de gênero e número. A descoberta

¹fenômenos lingüísticos em que uma sentença pode apresentar duas ou mais interpretações válidas.

²duas ou mais construções sintáticas com prefixo comum que podem ocorrer em um determinado ponto das sentenças da linguagem.

de conhecimento baseada em palavras-chave não fornece conhecimentos claros, pois é preciso analisar o resultado para tentar identificar informações presentes no texto. Esta abordagem geralmente resulta em um novo texto que, porém, contém apenas a essência das informações contidas no documento.

Segundo Loh (2001a), os conceitos³ representam melhor que as palavras diversos aspectos do mundo real, como, por exemplo, objetos, eventos, sentimentos e ações. A abordagem baseada em conceitos reduz o problema do vocabulário, pois esta não se preocupa em localizar palavras-chave nos documentos, verificando todas as variações possíveis que estas palavras podem ter em um texto; esta abordagem busca localizar conceitos presentes no texto.

Na abordagem baseada em conceitos, cada conceito é representado através de regras. As regras são compostas por termos, que podem ser abreviações, siglas, nomes próprios, ou qualquer conjunto de caracteres que apresentem significado no domínio da análise. Esta abordagem conta com dois modelos, o Modelo Espaço de Vetores e o Modelo Contextual.

3.6.1 Modelo Espaço de Vetores

De acordo com Loh (2001a), neste modelo os conceitos são representados por um vetor de termos, onde cada termo tem um peso, que é definido considerando a relevância dele para a definição do conceito.

O Modelo Espaço de Vetores (MEV) possui três tipos de termos, sendo eles:

Termo indicador - Aponta para um objeto ou elemento específico. Geralmente indica a presença de um conceito.

Termo caracterizador - Termo que é associado ao conceito, mas não indica sua presença. Utilizado para restringir elementos em um conjunto.

³Ação de formular uma idéia por meio de palavras. Noção, idéia, concepção (FERREIRA, 1975).

Termo descritor - Termo mais freqüente dentro de um texto, porém infreqüente na coleção como um todo. O termo descritor é formado por termos indicadores e caracterizadores.

No Modelo Espaço de Vetores (MEV), os termos indicadores - que são os termos mais relevantes - devem receber um peso mais alto em relação aos demais, pois a presença de um termo indicador no texto aumenta a chance do conceito estar presente no texto. Por outro lado, os termos caracterizadores - menos relevantes - devem receber pesos mais baixos em relação aos termos indicadores, pois apenas ajudam a identificar um conceito presente no texto. Os pesos devem ser normalizados para uma escala entre zero e um, para indicar a relevância do termo descritor Loh (2001a).

Por exemplo, para representar o conceito “*desnutrição*”, a palavra “*desnutrido*” deve receber um peso mais alto que a palavra “*nutrientes*”, pois “*nutrientes*” pode aparecer em outros textos com conceitos diferentes, como “*alimentação*” e “*saúde*”.

No modelo espaço de vetores, o texto é analisado como um todo à procura dos termos que compõem os conceitos. Este modelo não leva em consideração o contexto no qual os termos estão inseridos, o que pode prejudicar a confiança de um resultado onde um conceito foi encontrado. Um exemplo disto é a frase “*As crianças analisadas não apresentam sintomas de desnutrição.*”. Nesta frase, a palavra “*não*” foi utilizada para negar a existência de crianças desnutridas; porém, como a análise no MEV não considera o contexto no qual os termos estão inseridos, a presença da palavra “*não*” é descartada da análise, o que pode levar a uma identificação errada do conceito *desnutrição*.

3.6.2 Modelo Contextual

No Modelo Contextual, a idéia é permitir a análise do contexto em que os termos aparecem no texto. Assim, é possível entender melhor o significado dos termos e decidir se um conceito está presente ou não Loh (2001a).

O conceito, no Modelo Contextual, é formado por uma ou mais regras, onde cada regra possui um conjunto de termos positivos e pode conter um conjunto de regras de negação formado por termos negativos. Para uma regra ser verdadeira, todos os termos positivos devem estar presentes na frase e nenhuma regra de negação deve ser encontrada, i.e., nenhum dos termos de uma determinada regra de negação pode estar presente. Para um conceito estar presente no texto, basta que uma regra seja verdadeira para a frase sob análise Loh (2001a). No Modelo Contextual, a busca pelos conceitos no texto é realizada frase a frase - isto faz com que o contexto das palavras encontradas também seja analisado e levado em consideração. Este modelo é diferente do Modelo Espaço de Vetores, no qual todas as frases do texto são analisadas ao mesmo tempo à procura do conceito.

O conceito *Internet*, por exemplo, pode ser facilmente identificado em diversos contextos como em uma conversa entre pessoas ou em uma reportagem de uma revista. Nos dias atuais, o uso da Internet faz parte do dia-a-dia de muitas pessoas. A Internet é utilizada nas empresas para comunicação interna com os funcionários e comunicação externa com clientes. Muitas pessoas passam horas em salas de bate papo, baixando arquivos interessantes, lendo jornais *on-line*, checando *e-mails*, ou realizando qualquer outra atividade pela Internet. A seguir, na tabela 3.2, são definidas regras para o conceito *Internet* seguindo o Modelo Contextual:

Com base nessas regras, pode-se identificar a presença do conceito *Internet* na frase “*João está navegando na Internet à procura de um site para fazer o download de um aplicativo que execute arquivos MP3.*”. Analisando esta frase, é possível identificar que os termos *navegando* e *Internet* da *Regra 2* estão presentes na frase, tornando a regra verdadeira e conseqüentemente indicando a presença do conceito Internet no texto. Além da *Regra 2*, a *Regra 7* também é verdadeira, pois o termo *download* está presente na frase e nenhum termo negativo desta regra foi encontrado.

Mais um exemplo para a identificação do conceito *Internet* é encontrado na frase

Tabela 3.2. Regras do conceito Internet.

Conceito	Regras
Internet	<i>Regra 1:</i> navegar Web <i>Regra 2:</i> navegando Internet <i>Regra 3:</i> acessando ftp <i>Regra 4:</i> checar e-mail <i>Regra 5:</i> bate-papo <i>Regra 6:</i> baixar arquivo <i>Regra 7:</i> download ⇒ <i>Regra de Negação 1:</i> equipamento ⇒ <i>Regra de Negação 2:</i> máquina ⇒ <i>Regra de Negação 3:</i> dispositivo ⇒ <i>Regra de Negação 4:</i> aparelho <i>Regra 8:</i> upload ⇒ <i>Regra de Negação 1:</i> equipamento ⇒ <i>Regra de Negação 2:</i> máquina ⇒ <i>Regra de Negação 3:</i> dispositivo ⇒ <i>Regra de Negação 4:</i> aparelho

“João está conversando com amigos em um bate-papo para aprender a fazer upload de imagens de seu aparelho celular.”. Nesta frase, a *Regra 5* é verdadeira pela presença do termo “bate-papo” e a *Regra 8* é falsa, pois o termo “upload” foi encontrado, mas a *regra de negação 4* desta regra também foi validada pela presença do termo de negação “aparelho”. Embora a *Regra 8* seja falsa, o fato da *Regra 5* ser verdadeira é suficiente para afirmar a existência do conceito *Internet* no texto, pois pelo menos uma regra foi validada.

Todas as regras para a busca de um conceito devem ser escritas com base no domínio no qual o conceito está inserido. Regras com muitos termos podem ser ruins para a busca, visto que todos os termos positivos devem ser encontrados na regra para que a presença do conceito no texto possa ser validada. Por outro lado, regras com poucos termos podem ser genéricas demais, levando a inconsistência na extração do conhecimento. Logo, é necessário haver uma visão especialista do domínio para uma definição ideal das regras para os conceitos Loh (2001a).

3.6.3 Exemplo de aplicação do KDT baseado em Conceitos

Com base no estudo sobre Descoberta de Conhecimento em Texto, mais precisamente Descoberta de Conhecimento em Texto Baseado em Conceitos, dois experimentos foram elaborados. O primeiro experimento consistiu no desenvolvimento de um módulo em um sistema comercial para extração de conhecimento de uma base de dados em formato textual. Após o desenvolvimento do primeiro trabalho, o potencial da utilização da técnica de KDT baseada em Conceitos mostrou-se eficiente na extração de conhecimento do texto. Sendo assim, o potencial desta técnica motivou o desenvolvimento de um segundo experimento, consistindo na criação de uma interface para o controle de robôs com base na linguagem natural.

O primeiro experimento está detalhado a seguir, enquanto o segundo experimento está detalhado no capítulo 4.

3.6.4 Sistema SGPA

O SGPA (Sistema de Gerenciamento de Poços Automatizados) é um projeto em desenvolvimento numa parceira UFBA-Petrobras. É um sistema que tem por objetivo fornecer aos especialistas em engenharia de petróleo informações e conhecimentos que suportem as atividades de gerenciamento da produção de petróleo proveniente de poços *onshore*, isto é, aqueles localizados em terra. Neste sentido, são utilizadas diferentes bases de dados, modelagem de conhecimento de especialistas e técnicas de IA.

As funcionalidades do SGPA, em sua segunda versão, encontram-se divididas em blocos desenvolvidos de forma modular e integrados ao sistema. Um dos principais módulos desenvolvidos é conhecido como Jargão, sendo responsável pela extração de conhecimento das bases textuais da Petrobras. O módulo Jargão é ilustrado na figura 3.4 e detalhado a seguir.

Módulo Jargão

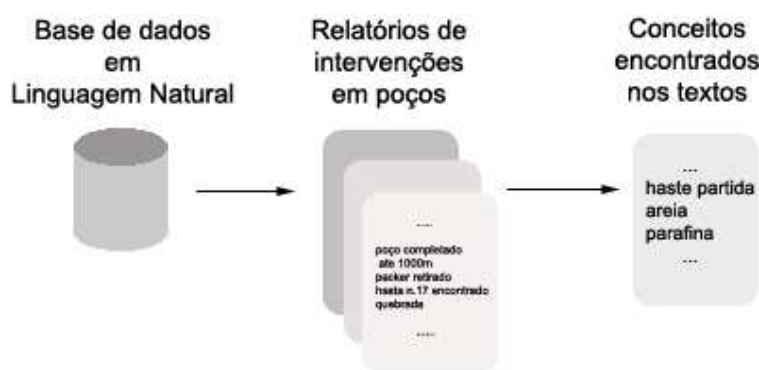


Figura 3.4. Estrutura do Módulo JARGÃO.

O módulo Jargão é voltado para o apoio à tomada de decisão gerencial de poços de petróleo, através da extração de conhecimento da Base de Dados - em linguagem natural - dos relatórios realizados durante as intervenções nos poços da PETROBRAS/UN-BA. Os dados textuais utilizados no Módulo Jargão são os relatórios de intervenção escritos por técnicos de campo. Estes relatórios utilizam termos específicos de engenharia de petróleo e compõem o que é chamado de “base de dados de intervenção”.

As informações textuais presentes nas bases de dados da Petrobras podem complementar a análise realizada pelo especialista. Porém, atualmente, as informações das bases de dados não são utilizadas de forma rápida e precisa, pois os textos não são armazenados de forma estruturada. Devido às dificuldades encontradas pelos especialistas, a leitura de relatórios impressos demanda um alto investimento de tempo para extrair informações.

Para aperfeiçoar o processo de apoio à tomada de decisão, a implementação do módulo Jargão utiliza a Descoberta de Conhecimento Baseada em Conceitos, proposta por Loh (2001a), nos dados das intervenções dos poços. Com base no conhecimento

dos engenheiros de petróleo da Petrobras, alguns conceitos da engenharia de petróleo foram mapeados e inseridos no módulo Jargão.

O módulo Jargão tem como objetivo encontrar, em textos, os conceitos inseridos em sua base de dados. Além disso, este módulo provê os resultados da análise dos textos a outras funcionalidades do SGPA; tais resultados podem servir como dados de entrada para uma análise posterior.

Após a descoberta dos conceitos do domínio da engenharia de petróleo nos registros das intervenções ocorridas nos poços, o usuário pode fazer uma nova análise, com o objetivo de agrupar poços semelhantes com base nos conceitos encontrados nas coleções de intervenções de cada um. A seguir, os conceitos serão descritos, bem como a análise inteligente de agrupamento.

Conceitos utilizados no Módulo Jargão

Os conceitos especialistas utilizados no módulo Jargão foram inseridos pelos engenheiros especialistas em petróleo da Petrobras, com base nas análises dos relatórios de intervenções dos poços. A identificação das características de um poço antes da utilização das técnicas de Descoberta de Conhecimento em Texto era demorada e cansativa, exigindo dos engenheiros de petróleo a leitura de todos os relatórios impressos sobre o que foi realizado na intervenção do poço em questão. A leitura destes relatórios demandava um tempo considerável, pois em alguns casos podia ser necessário ler as três últimas ou até mesmo todas as intervenções que o poço sofreu desde a sua perfuração, com o propósito de identificar as características de um poço.

A seguir, as regras para identificar os conceitos “areia” e “haste partida” - presentes no contexto de problemas na prospecção de petróleo - são apresentados na tabela 3.3:

Conforme visto na seção 3.6 e detalhado na subseção 3.6.2, para um conceito estar presente na frase, pelo menos uma regra positiva deve ser encontrada. Isto é, todos

Tabela 3.3. Alguns conceitos da engenharia de petróleo.

Conceitos	Regras
Areia	<i>Regra 1:</i> areia ⇒ <i>Regra de Negação 1:</i> fraturamento ⇒ <i>Regra de Negação 2:</i> colchão areia <i>Regra 2:</i> circulação areia <i>Regra 3:</i> obstruído filtro areia <i>Regra 4:</i> removido detrito
Haste partida	<i>Regra 1:</i> danificada rosca <i>Regra 2:</i> desenroscada haste <i>Regra 3:</i> partida haste <i>Regra 4:</i> suada rosca

os termos positivos da regra devem ser encontrados na frase e nenhuma das regras de negação - também compostas por termos - deve ser encontrada na frase.

Considere a seguinte frase:

- Efetuando colchão de areia a 1200m.

O conceito areia não será encontrado nesta na frase, pois nenhuma das regras que compõem o conceito Areia foram validadas. Conforme pode ser observado, apenas na *Regra 1* todos os termos positivos foram encontrados, ou seja, o termo “areia”. A *Regra 1* seria validada, mas a *Regra de Negação 2* que a compõe também foi identificada, negando a presença do conceito *Areia*.

Com o propósito de ampliar a busca de conceitos para vários poços e melhorar a leitura sobre os conceitos identificados nas coleções textuais das intervenções, foi implementada a técnica de agrupamento da Descoberta de Conhecimento em Texto, comentada na subseção 3.5.3. A seguir, a utilização da técnica de agrupamento é comentada.

Agrupamento de Poços

Visando possibilitar ao usuário a identificação visual de grupos de poços forma-

dos por características semelhantes, foi implementado o algoritmo de agrupamento de poços utilizando a Rede Neural SOM (*Self-Organizing Map*). O algoritmo de agrupamento de poços possui como entrada o conhecimento adquirido na execução do algoritmo de *Text Mining* implementado no módulo Jargão, que objetiva identificar a ocorrência dos conceitos especialistas da engenharia de petróleo nas coleções textuais das intervenções ocorridas nos poços.

Para a execução do algoritmo de agrupamento de poços é necessário formar vetores de características que serão submetidos a entrada da Rede Neural SOM. Cada vetor de característica é formado com base na análise de um poço, onde cada característica é formada pela quantidade de vezes que um conceito analisado foi identificado na coleção textual de intervenções do poço analisado. O detalhamento do funcionamento da Rede SOM pode ser visto em Kohonen (1990).

Após o processamento da Rede SOM, ocorre o agrupamento dos poços, com características semelhantes, inseridos na entrada da Rede Neural SOM. Os agrupamentos de poços semelhantes são possíveis de serem identificados visualmente, observando as áreas de mesmo tom de cor.

Como exemplo, a tabela 3.4 apresenta características de 14 poços, onde estas características foram retiradas de boletins de intervenções ocorridas nos poços dessa amostra. Um fragmento das características encontradas nos poços analisando a coleção textual dos boletins de intervenção são apresentadas na tabela 3.5. As características analisadas foram: Parafina, Areia, Corrosivo, Haste Partida, *Scale* e Fraturamento. Observando a tabela 3.4 pode-se visualizar a ocorrência das características citadas nos poços de petróleo.

Tabela 3.4. Dados de entrada da Rede SOM.

Poço	Parafina	Areia	Corrosivo	Haste Partida	Scale	Fraturamento
P0004	0	0	0	0	4	0
P0013	3	5	0	5	3	1
P0036	0	6	0	4	0	0
P0078	4	5	0	0	2	1
P0079	2	1	0	0	0	1
P0093	1	4	0	0	1	1
P0098	0	0	1	0	0	1
P0134	2	1	1	0	5	1
P1021	0	1	0	1	0	1
P1823	8	0	0	0	0	0
P2833	0	1	1	2	3	0
P2946	0	4	1	0	8	0
P2991	2	1	0	1	2	1
P3674	0	2	1	1	0	1

Tabela 3.5. Conceitos identificados em alguns poços.

Poço	Conceito	Texto em linguagem natural
P0013	Areia e Parafina	Retirando coluna 2 7/8" EU com equipamentos BM, 05 tubos parafinados com petróleo viscoso escorrendo para um tonel, pegando no balde jogando no tanque de lama, poço golfando bastante, poço não circulou, e coluna de haste com auxílio do back-off. Retirado tubo acima da BF preso na BF com areia. * 04 hastes empenou durante a operação.
P0093	Parafina	Retirando coluna 2 7/8" EU com equipamentos BM, (hastes presas na bomba). * Tubos parafinados sem condições de pistonear e óleo bastante viscoso, escorrendo lentamente p/ um tonel, evitando poluir a área.
P2991	Parafina	Completando poço com 8 bbl de fluido e tentando circular reverso, sem êxito, pressão atingindo 1000 psi, drenando pressão e reiniciando bombeio, pressão atingindo 1200 psi, retornando os bolos de parafina.
P1021	Haste partida	Desequipando poço BCP para retirada da coluna de hastes encontrando partida na haste polida.

Os dados da tabela 3.4 foram submetidos à entrada da Rede SOM. Resultando, após o processamento, na imagem da figura 3.5.

É possível observar na figura 3.5, o agrupamento de poços formado pelas características semelhantes identificadas na análise da Rede SOM e exibida na imagem por cores com o mesmo tom.

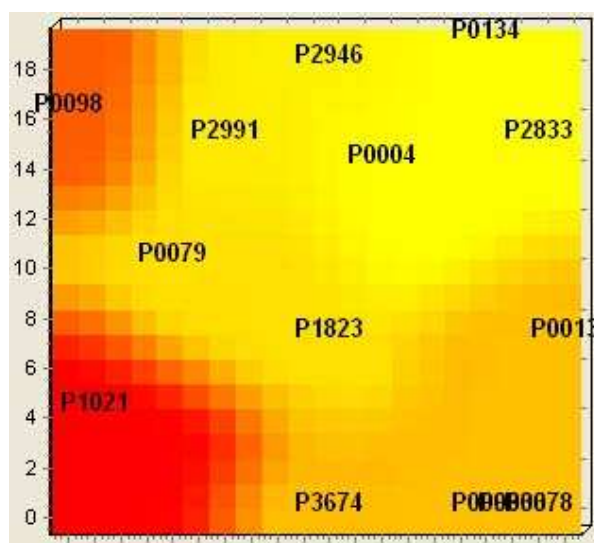


Figura 3.5. Visualização de agrupamento de poços no SGPA.

As áreas visualmente separáveis na imagem são formadas por padrões cujas características são bastante diferentes dos seus vizinhos.

UMA PROPOSTA PARA A INTERFACE ROBÓTICA

Neste capítulo, uma proposta para a interface robótica é apresentada. Esta interface tem como base o uso da linguagem natural para a comunicação entre seres humanos e robôs. Utilizando a interface proposta, o usuário apenas informa os passos que devem ser executados para o cumprimento da tarefa. Após inserir a descrição da tarefa, um processamento é realizado com o objetivo de extrair os comandos e parâmetros necessários para o adequado cumprimento da tarefa proposta.

Para validar e definir a interface proposta, um protótipo robótico foi projetado e construído, servindo como plataforma para os experimentos da interface.

4.1 ESTRUTURA DE HARDWARE

A estrutura de *hardware* do protótipo robótico é apresentada na figura 4.1, onde os seguintes componentes merecem destaque:

- 1 computador;
- 1 módulo eletrônico;
- 2 motores DC;
- 1 câmera de vídeo CMUCam1;
- 2 servos motores.

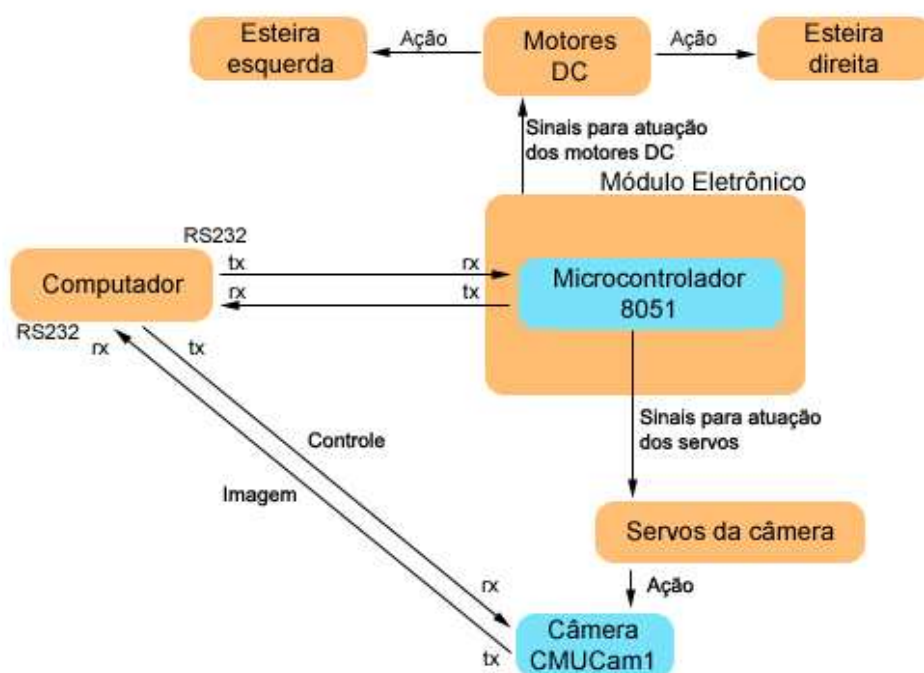


Figura 4.1. Diagrama eletrônico do protótipo.

O computador é responsável por todo processamento realizado para a interpretação da linguagem natural utilizada na interface. A partir do texto inserido na interface, algoritmos presentes no computador extraem os parâmetros e os comandos que o robô deve executar para o cumprimento da tarefa. Os algoritmos que controlam a execução das tarefas são processados pelo computador, porém os algoritmos para o controle dos motores DC e servos motores são processados no módulo eletrônico. A comunicação entre o computador e o módulo eletrônico é realizada por um cabo umbilical conectado entre a porta serial do computador e o conector serial do módulo eletrônico. Os sinais transmitidos pelo computador utilizam o protocolo RS232.

O módulo eletrônico possui diversos componentes eletrônicos responsáveis pela alimentação e pelo controle dos motores utilizados no protótipo. O principal componente presente no módulo eletrônico é um microcontrolador da família 8051, que é responsável pelos pulsos elétricos necessários ao acionamento dos servos motores e motores DC. Os servos motores são acionados por sinais PWM (*Pulse-Width Modulation*

- Modulação por Largura de Pulso) gerados pelo microcontrolador. Os componentes da placa do robô e o protocolo de comunicação com os componentes do robô estão detalhados no apêndice A.2.

Os dois motores DC são responsáveis pela movimentação do robô no ambiente e cada motor está fixado a uma esteira com o propósito de facilitar manobras do robô em pequenos espaços. A partir do movimento dos motores DC, o robô pode se deslocar para a frente, para trás, para a direita, para a esquerda, além de girar sobre o seu próprio eixo no sentido horário ou anti-horário.

A câmera de vídeo CMUCam1 possui um *firmware* capaz de enviar as imagens capturadas no ambiente para o computador através de comunicação serial, utilizando o protocolo RS232. O computador utiliza a mesma comunicação serial para enviar comandos referentes à captura de imagens no ambiente.

Os servos motores são responsáveis pela movimentação da câmera independentemente do robô, com o objetivo de obter uma melhor leitura das informações do ambiente. Os servos motores são utilizados para formar uma cabeça de visão, onde o servo da base propicia o movimento angular horizontal da câmara enquanto que o servo do topo, associa um movimento angular vertical.

O protótipo construído é apresentado na figura 4.2, onde é possível observar a acomodação dos componentes citados.

4.2 DEFINIÇÃO DO CONTEXTO

Conforme descrito ao longo da seção 4.1, o robô é capaz de se movimentar para a frente, para trás, para a direita e para a esquerda utilizando os motores DC. Além disso, é também possível identificar padrões de cores na imagem, o que possibilita a utilização do conjunto de instruções do *firmware* da câmera.

Observando o conjunto de possibilidades que a estrutura de *hardware* da proposta

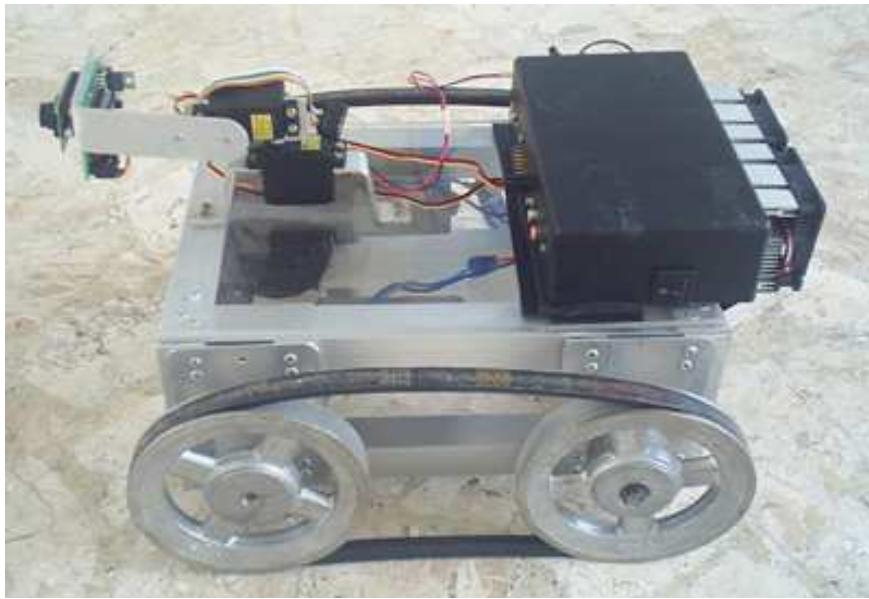


Figura 4.2. O protótipo.

fornece, é possível definir um universo de ações que o robô é capaz de executar. O universo definido para a validação da interface está apresentado a seguir:

- Navegar pelo ambiente;
- Localizar um objeto com uma cor especificada;
- Seguir um objeto com uma cor especificada;
- Movimentar a câmera;
- Capturar uma imagem do ambiente.

A direção e o tempo constituem parâmetros para a execução do comando movimentar. Portanto, para o robô se movimentar pelo ambiente, é necessário que sejam definidos o tempo que os motores devem permanecer acionados e a direção do movimento.

Para a localização de uma cor no ambiente, é necessário que esta constitua um tom de cor reconhecido pela interface. Caso o tom de cor seja válido, a interface utiliza o

comando *movimentar*, buscando identificar a cor informada no ambiente.

O comando *seguir cor* utiliza comandos do *firmware* da câmera de vídeo para identificar o tom de cor do objeto que está na frente da câmera. A partir da identificação da cor a ser seguida, a interface utiliza o comando *movimentar*, com o objetivo de manter aquele tom de cor diante do robô. Caso a cor deixe de ser identificada, a interface utiliza o comando *procurar cor*, com a finalidade de deixar o robô de frente para o objeto. Logo em seguida, a interface reinicia o movimento para seguir a cor.

Para movimentar a câmera, o valor do movimento angular horizontal e vertical dos servos da base e topo da câmera devem ser informados como parâmetro.

A captura da imagem do ambiente pode ser solicitada a qualquer momento pela interface, visando documentar o cumprimento da tarefa ou disponibilizar a imagem para outros *softwares*.

A identificação dos comandos que o robô deve executar a partir de um texto utiliza a técnica de Descoberta de Conhecimento em Texto baseada em Conceitos, descrita no capítulo 3. Analisando o universo de ações que o robô é capaz de executar, é possível identificar os conceitos existentes no universo da proposta, como, por exemplo, os conceitos movimentar, procurar e seguir. A próxima seção detalha o universo de conceitos existentes na proposta e a forma de identificá-los e associá-los a comandos definidos na interface para a manipulação do robô.

4.2.1 Conceitos implementados na interface

Os conceitos implementados na interface, que são responsáveis pelo mapeamento dos comandos que o robô é capaz de executar, são classificados em duas categorias:

- Comandos
- Parâmetros

Os conceitos identificados como comandos são aqueles que mapeiam os comandos que o robô é capaz de executar. Exemplos desta categoria incluem os conceitos *movimentar* - responsável por mapear a ação de locomoção do robô -, e *procurar* - responsável por mapear a ação de procurar uma cor no ambiente. Estes conceitos estão expostos com detalhes na tabela 4.1.

Como é possível observar na tabela 4.1, os conceitos *movimentar*, *procurar*, *movimentar câmera* e *capturar* são dotados apenas de regras positivas. Nenhuma palavra foi encontrada que pudesse invalidar a presença de uma das regras positivas que compõem os conceitos. Geralmente, um conceito composto por regras positivas contém regras de negação quando alguma palavra que compõe a regra positiva pode apresentar mais de um significado, a depender do contexto no qual está inserida, ou pode ter outro significado, quando aparece próxima a uma determinada palavra. A regra de negação encontrada na definição do conceito *seguir* é um bom exemplo de palavras com mais de um significado, pois as palavras *seguir* e *siga* associadas à palavra *frente* apresentam conotação de direção de movimentação e, por este motivo, foram inseridas como regras positivas para o conceito *movimentar*.

Os conceitos identificados como parâmetros são aqueles que se comportam como os argumentos de um comando, sendo utilizados com o intuito de indicar a forma como o comando deve ser executado. Por exemplo, a tabela 4.2 mostra o mapeamento existente entre os conceitos *movimentar*, *procurar*, *seguir*, *movimentar câmera* e *capturar*, com seus respectivos parâmetros.

Como pode ser observado na tabela 4.2, o conceito *movimentar* apresenta os parâmetros *frente*, *ré*, *esquerda* e *direita*, que podem ser utilizados para orientar a direção do movimento que o robô deve executar. O parâmetro *segundos* é utilizado como temporizador para a execução do comando *movimentar*.

O conceito *procurar* possui o parâmetro *cor* para indicar que uma cor deve ser localizada no ambiente, e esta cor é definida por um outro parâmetro do comando. Os parâmetros *azul*, *amarelo*, *verde* e *vermelho* são utilizados para indicar a cor que deve

Tabela 4.1. Conceitos implementados na interface para o mapeamento dos comandos.

Conceitos	Regras positivas
movimentar	caminhar caminhe correr ande andar movimentar ⇒ <i>Regra de Negação:</i> câmera movimente ⇒ <i>Regra de Negação:</i> câmera vá siga frente siga adiante seguir frente
procurar	encontrar localizar localize procurar procure ache achar
seguir	seguir ⇒ <i>Regra de Negação:</i> frente siga ⇒ <i>Regra de Negação:</i> frente
movimentar câmera	movimente câmera vá câmera movimentar câmera vire câmera mexa câmera
capturar	capture imagem pegue imagem capturar imagem capturar ambiente tirar foto tire foto capte imagem armazenar imagem armazene imagem

Tabela 4.2. Associação dos conceitos comandos com os conceitos parâmetros.

Conceitos comandos	Conceitos parâmetros	Parâmetro temporal
movimentar	frente ré esquerda direita	segundos
procurar	cor → azul → amarelo → verde → vermelho	segundos
seguir	cor	não se aplica
movimentar câmera	esquerda direita cima baixo ângulo	não se aplica
capturar	imagem	não se aplica

ser localizada.

A tabela 4.3 descreve a formação de alguns *conceitos parâmetros* utilizados pelos *conceitos comandos* com a finalidade de permitir a execução de alguma ação definida na tarefa.

Tabela 4.3. Conceitos parâmetros.

Conceitos	Regras
frente	frente reto
ré	ré trás
direita	direita
esquerda	esquerda
segundos	segundos segundo
cor	azul amarelo verde vermelho tempo cor

4.3 ALGORITMOS IMPLEMENTADOS

A seguir, os algoritmos utilizados para reconhecimento e execução das ações serão apresentados.

4.3.1 Reconhecimento dos conceitos

O reconhecimento de conceitos, implementado na interface, é descrito no capítulo 3 na seção 3.6.2. O primeiro passo a ser executado na identificação dos conceito é a separação do texto em orações, visando o reconhecimento de conceitos por oração e não ao longo do texto. As orações do texto são separadas por vírgulas ou pontos.

Para um conceito ser reconhecido em um texto é necessário que pelo menos uma regra de afirmação seja identificada e nenhuma regra de negação associada à regra de afirmação deve ser identificada. A tabela 4.4 apresenta o algoritmo de reconhecimento de conceitos.

Tabela 4.4. Algoritmo para o reconhecimento de conceitos.

Passo	Descrição
1	Substituir caracteres com acentos por caracteres sem acentos, visando evitar problemas com acentuações.
2	Separar a frase em várias orações. Os separadores de orações são a vírgula e o ponto.
3	Para cada oração da frase, faça:
4	Para cada conceito cadastrado, faça:
5	Enquanto uma regra de afirmação não é validada, faça:
6	Recuperar uma regra de afirmação.
7	Buscar termos da regra de afirmação e procurar na oração.
8	Se todos os termos da regra forem encontrados, faça:
9	Enquanto uma regra de negação associada a regra de afirmação não é validada, faça:
10	Recuperar uma regra de negação.
11	Buscar termos da regra de negação e procurar na oração.
12	Se algum termo da regra for encontrado, então:
13	Desprezar regra de afirmação. Sair do enquanto da linha 11
14	Fim do Se
15	Fim do Enquanto
16	Se não validar regras de negação, faça:
17	Regra de afirmação válida.
18	Fim do Se
19	Fim do Se
20	Fim do Enquanto
21	Se uma regra de afirmação for validada, então:
22	Armazenar conceito identificado.
23	Fim do Se
24	Fim do Para
25	Fim do Para

4.3.2 Reconhecimento das ações

O texto digital informado pelo usuário é separado nas diversas orações que o compõem. As orações são submetidas ao algoritmo de reconhecimento de conceitos, a partir do qual os conceitos encontrados na oração são retornados.

O algoritmo de reconhecimento das ações que o robô deve executar recebe os conceitos encontrados na oração e procura por um *conceito comando*. Ao encontrar um *conceito comando*, o algoritmo procura, entre os demais conceitos encontrados na oração, seus parâmetros previamente cadastrados como *conceito parâmetros*. Quando um parâmetro é encontrado entre os conceitos, este é então associado ao *conceito*

4.3 ALGORITMOS IMPLEMENTADOS

comando em questão. O algoritmo desenvolvido para o reconhecimento das ações pode ser observado na tabela 4.5.

Tabela 4.5. Algoritmo para o reconhecimento da ação.

Passo	Descrição
1	Submeter o texto ao reconhecimento de conceitos.
2	Armazenar os conceitos encontrados em cada oração em uma coleção, formando coleções de conceitos por oração.
3	Para cada coleção de conceitos encontrada em cada oração do texto, faça:
4	Localizar um <i>conceito comando</i> na coleção de conceitos.
5	Se encontrar um <i>conceito comando</i> , então:
6	Armazenar o conceito comando sob forma de ação.
7	Se a ação identificada requer algum parâmetro, então:
8	Para cada conceito na coleção de conceitos, faça:
9	Se o conceito for <i>conceito parâmetro</i> e estiver associado à ação como parâmetro, então:
10	Associar o <i>conceito parâmetro</i> à ação.
11	Fim do se
12	Fim do para
13	Se nenhum parâmetro for identificado, assumir os valores <i>defaults</i> associados ao comando.
14	Associar os valores <i>default</i> à ação.
15	Fim do se
16	Fim do se
17	Senão:
18	Rejeitar a oração do texto, pois não possui comando a ser executado.
19	Fim do se
20	Fim do para

A figura 4.3 mostra um exemplo de utilização do algoritmo descrito da tabela 4.5.

O reconhecimento de ações é executado oração a oração do texto que compõe a tarefa, impossibilitando a análise do contexto das ações presentes em cada oração. Desta forma, a interface possui limitação semântica em relacionar o conhecimento extraído sobre os comandos identificados nas orações.

4.3.3 Execução das ações

O protótipo do estudo de caso apresenta um conjunto limitado de ações que podem ser executadas. As ações implementadas estão associadas aos conceitos descritos na seção 4.2.1. As ações que o protótipo pode executar são:

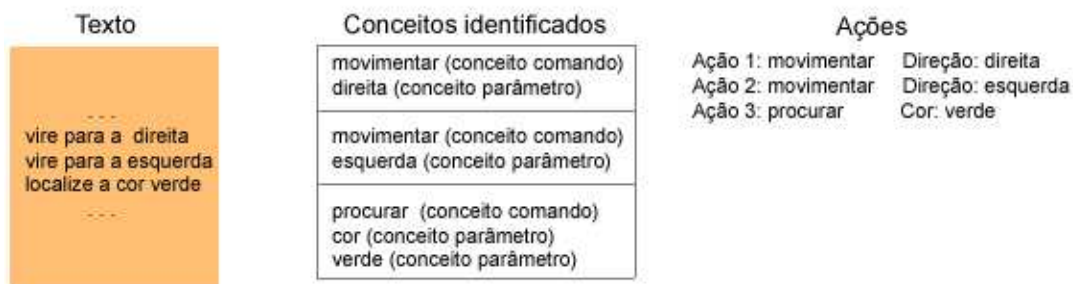


Figura 4.3. Exemplo de reconhecimento de ações a partir de um texto.

- Movimentar
- Procurar
- Seguir
- Movimentar câmera
- Capturar

Comando Movimentar

O comando movimentar é responsável pela movimentação do robô no ambiente através do acionamento dos motores-redutores acoplados ao robô.

Os parâmetros para a execução da ação movimentar estão listados na tabela 4.6.

Tabela 4.6. Comando Movimentar.

Ação	Parâmetros
movimentar	frente ré esquerda direita segundos

A execução da ação *movimentar* segue o algoritmo descrito na tabela 4.7.

Tabela 4.7. Algoritmo da ação movimentar.



1	Verificar os parâmetros para a ação movimentar.
2	Identificar qual a orientação do movimento através do parâmetro correspondente, que pode ser <i>frente</i> , <i>ré</i> , <i>esquerda</i> e <i>direita</i> .
3	Solicitar ao Módulo responsável pelo controle do Microcontrolador (Módulo Microcontrolador) o acionamento dos motores-redutores responsáveis pela movimentação do robô, informando a orientação da movimentação.
4	Se existir o parâmetro <i>segundos</i> , então:
5	Manter os motores-redutores acionados durante o tempo informado no parâmetro <i>segundos</i> .
6	Senão:
7	Assumir o tempo de 3 segundos como <i>default</i> para o movimento.
8	Fim do se.
9	Solicitar ao <i>Módulo Microcontrolador</i> a parada dos motores-redutores.
10	Se cumprir todos os passos sem erros, então:
11	Retornar verdadeiro.
12	Senão:
13	Retornar falso.
14	Fim do se.

Comando Procurar

O comando *procurar* tenta localizar a presença de uma cor em algum objeto inserido no ambiente. As cores que a ação deve localizar no ambiente são previamente cadastradas com base na cor mínima e máxima que caracteriza o tom da cor.

Uma cor é formada pela combinação de vermelho(red), verde(green) e azul(blue) - cores representadas, respectivamente, por R, G e B. Por exemplo, a tabela 4.8 informa qual a cor mínima e máxima para se encontrar o tom amarelo no ambiente, bem como os valores de R, G e B que definem a cor

Tabela 4.8. Tom amarelo.

	cor	R	G	B
Amarelo - Cor mínima		178	178	0
Amarelo - Cor máxima		255	255	76

Uma consideração importante a ser feita acerca da definição do tom da cor é que a iluminação do ambiente deve ser observada, pois a cor capturada pela câmera sofre alterações de acordo com a variação de luminosidade.

Os parâmetros para a execução do comando *procurar* estão enunciados na tabela 4.9.

Tabela 4.9. Comando Procurar.

Ação	Parâmetros
procurar	cor azul amarelo verde vermelho segundos

A execução da ação *procurar* segue o algoritmo descrito na tabela 4.10.

Tabela 4.10. Algoritmo da ação procurar.

1	Verificar os parâmetros para a ação procurar.
2	Identificar, entre os parâmetros, qual cor deve ser localizada no ambiente.
3	Se existir o parâmetro cor, então:
4	Consultar a base de dados para recuperar os valores R, G e B da cor mínima e máxima do parâmetro.
5	Montar o comando que será enviado ao <i>firmware</i> da câmera para que o tom de cor possa ser buscado.
6	Fim do se
7	Começar a contar o tempo em segundos.
8	Enquanto não alcançar o valor do parâmetro <i>segundos</i> , faça:
9	Solicitar ao <i>firmware</i> da câmera a localização da faixa de cor passada.
10	Se não encontrar a cor na imagem, então:
11	Solicitar ao <i>Módulo Microcontrolador</i> a movimentação da cabeça de visão para a direita.
12	Senão:
13	Retornar verdadeiro.
14	Fim do se.
15	Solicitar ao <i>firmware</i> da câmera a localização da faixa de cor passada.
16	Se não encontrar a cor na imagem, então:
17	Solicitar ao <i>Módulo Microcontrolador</i> a movimentação da cabeça de visão para esquerda.
18	Senão:
19	Retorne verdadeiro.
20	Fim do se.
21	Se não encontrar a cor na imagem, então:
22	Solicitar ao <i>Módulo Microcontrolador</i> a movimentação da cabeça de visão para o centro.
23	Escolher aleatoriamente entre ir para frente, trás, direita e esquerda.
24	Executar a ação movimentar com o parâmetro escolhido aleatoriamente.
25	Fim do se.
26	Fim do enquanto.
27	Retorne falso.

Comando Seguir

O comando *seguir* só implementa a funcionalidade seguir cor, que capta a cor média de um objeto que está no centro da imagem da câmera. Com a cor capturada, um cálculo com base na cor é realizado e um comando de localizar cor é emitido, o qual será executado pelo *firmware* da câmera.

A todo instante, a leitura da câmera é realizada em busca da localização da cor na imagem. Caso o ponto de concentração da cor saia da região do centro da imagem, a cabeça de visão é movimentada para direita e esquerda, assim como para baixo e para cima, com o intuito de manter o objeto no centro da imagem da câmera. Caso o objeto saia do centro da imagem, o robô se movimentará para o último lado em que o objeto estava. A ação será executada com duração informada pelo parâmetro tempo; caso este não seja informado, um tempo de 30 segundos será adotado. A tabela 4.11 descreve o algoritmo utilizado para a execução do comando seguir.

Tabela 4.11. Algoritmo da ação seguir.

1	Capturar a cor média do objeto que está no centro da imagem.
2	Assumir que a cor capturada é a cor mínima.
3	Somar 30 a cada valor do R, G e B da cor mínima e gerar a cor máxima.
4	Montar o comando que será enviado ao <i>firmware</i> da câmera para que o tom de cor possa ser rastreado.
5	Se for informado o tempo em que a ação seguir deve ser executada, então:
6	O tempo de execução é atribuído ao valor do parâmetro <i>segundos</i> .
7	Senão:
8	O tempo de execução é atribuído ao valor 30 segundos.
9	Fim do se.
10	Começar a contar o tempo em segundos.
11	Enquanto não alcançar o valor do tempo de execução em segundos, faça:
12	Solicitar ao <i>firmware</i> da câmera a localização da faixa de cor passada.
13	Dividir a imagem em três regiões: esquerda, centro e direita.
14	Se o centro da cor encontrada na imagem estiver na região da esquerda, então:
15	Solicitar ao <i>Módulo Microcontrolador</i> a movimentação da cabeça de visão para esquerda.
16	Senão, se estiver na região da direita, então:
17	Solicitar ao <i>Módulo Microcontrolador</i> a movimentação da cabeça de visão para direita.
18	Senão, se estiver na região do centro, então:
19	Manter a câmera na posição atual.
20	Senão, se a cor não foi identificada, então:
21	Solicitar ao <i>Módulo Microcontrolador</i> a movimentação da cabeça de visão para o centro.
22	Solicitar ao <i>Módulo Microcontrolador</i> a movimentação do robô para o último lado em que a cor foi encontrada.
23	Se tentar por três vezes e não conseguir localizar a cor na imagem, então:
24	Escolher aleatoriamente entre ir para frente, trás, direita e esquerda.
25	Executar a ação movimentar com o parâmetro escolhido aleatoriamente.
26	Fim do se.
27	Fim do enquanto.
28	Fim do enquanto.
29	Solicitar ao <i>firmware</i> da câmera a localização da faixa de cor passada.
30	Se encontrar a cor na imagem, então:
31	Retornar verdadeiro.
32	Senão:
33	Retornar falso.
34	Fim do se.

Comando Movimentar Câmera

O comando movimentar câmera é responsável pela movimentação dos servos motores da cabeça de visão do robô.

Os parâmetros para a execução da ação movimentar câmera estão listados na tabela 4.12.

Tabela 4.12. Comando Movimentar câmera.

Ação	Parâmetros
movimentar câmera	esquerda direita cima baixo ângulo

A execução da ação *movimentar câmera* segue o algoritmo descrito na tabela 4.13.

Tabela 4.13. Algoritmo da ação movimentar câmera.

1	Verificar os parâmetros para a ação movimentar câmera.
2	Identificar qual a orientação do movimento através do parâmetro correspondente, que pode ser <i>esquerda</i> , <i>direita</i> , <i>cima</i> e <i>baixo</i> e qual o ângulo para o movimento.
3	Solicitar ao Módulo Microcontrolador o acionamento dos servos motores responsáveis pela movimentação da cabeça de visão do robô, informando a orientação da movimentação.
4	Se existir o parâmetro <i>ângulo</i> , então:
5	Movimentar o servo motor responsável pelo movimento desejado para o ângulo identificado.
6	Senão:
7	Assumir o extremo da direção como o ângulo <i>default</i> .
8	Fim do se.
9	Se cumprir todos os passos sem erros, então:
10	Retornar verdadeiro.
11	Senão:
12	Retornar falso.
13	Fim do se.

Comando Capturar

O comando capturar é responsável por capturar uma imagem do ambiente utilizando a câmera CMUCam1.

O parâmetro para a execução da ação capturar é listado na tabela 4.14.

Tabela 4.14. Comando Capturar.

Ação	Parâmetro
capturar	imagem

A execução da ação *capturar* segue o algoritmo descrito na tabela 4.15.

Tabela 4.15. Algoritmo da ação capturar.

1	Enviar ao <i>firmware</i> da câmera o comando $DF\backslash r$
2	Armazenar os dados da imagem capturada.
3	Se cumprir todos os passos sem erros, então:
4	Retornar verdadeiro.
5	Senão:
6	Retornar falso.
7	Fim do se.

RESULTADOS EXPERIMENTAIS

O objetivo deste capítulo é demonstrar as etapas para o reconhecimento, a partir de textos livres, de ações que o robô deve executar. O texto informado pelo usuário deve conter os passos necessários para o cumprimento de uma tarefa.

O reconhecimento de ações em um texto começa pela identificação de conceitos presentes nas respectivas orações. Conforme visto no capítulo 4, os conceitos são classificados em conceito comando e conceito parâmetro. A partir do reconhecimento destes conceitos, as ações que o robô deve executar com a finalidade de cumprir a tarefa são identificadas.

O detalhamento das etapas para o reconhecimento das ações é apresentado a seguir.

5.1 EXEMPLO DETALHADO

A demonstração detalhada do procedimento para o reconhecimento das ações a partir de um texto em linguagem natural utiliza o exemplo de texto apresentado na tabela 5.1.

Tabela 5.1. Tarefa em linguagem natural.

Robô, movimente-se para a frente por 5 segundos. Logo em seguida, vire para a direita. Depois, vá para a esquerda. Procure a cor amarela no ambiente por 120 segundos, no máximo. Movimente a câmera para baixo. Capture uma imagem do ambiente e armazene.
--

Dado um texto informado pelo usuário, a interface separa o texto em orações. As orações identificadas na tarefa descrita na tabela 5.1 estão apresentadas na tabela 5.2.

Tabela 5.2. Orações da tarefa descrita na tabela 5.1.

Oração 1	Robô, movimente-se para a frente por 5 segundos.
Oração 2	Logo em seguida, vire para a direita.
Oração 3	Depois, vá para a esquerda.
Oração 4	Procure a cor amarela no ambiente por 120 segundos, no máximo.
Oração 5	Movimente a câmera para baixo.
Oração 6	Capture uma imagem do ambiente e armazene.

Para cada oração do texto, o algoritmo implementado na interface executa a técnica de Descoberta de Conhecimento em Texto baseado em Conceito e extrai os conceitos presentes na oração. A partir da identificação dos conceitos na oração, a interface monta uma ação associada ao conceito comando identificado. A tabela 4.5, apresentada no capítulo 4, descreve o algoritmo implementado na interface para o reconhecimento de ações em uma oração.

Para detalhar e exemplificar o reconhecimento de uma ação, cada uma das orações da tarefa descrita na tabela 5.2 é usada como exemplo.

As tabelas 5.3, 5.4, 5.5, 5.6, 5.7 e 5.8 apresentam a execução do algoritmo de reconhecimento de ação nas orações da tabela 5.2.

5.1 EXEMPLO DETALHADO

Tabela 5.3. Passos para o reconhecimento de uma ação na oração “Robô, movimente-se para a frente por 5 segundos.”.

Passo	Descrição
1	Extrair os conceitos presentes na oração “Robô, movimente-se para a frente por 5 segundos.”.
2	⇒ Conceitos identificados: <i>movimentar</i> , <i>frente</i> , <i>tempo</i>
3	Localizar um <i>conceito comando</i> na coleção de conceitos.
4	⇒ Conceito comando identificado: <i>movimentar</i>
5	Se encontrar um <i>conceito comando</i> , então:
6	Armazenar o conceito comando sob forma de ação.
7	⇒ Criar um item na lista de ações que o robô deve executar
8	⇒ Comando <i>movimentar</i> inserido na lista de ações, aguardando parâmetros.
9	Se a ação identificada requer algum parâmetro, então:
10	Para cada conceito na coleção de conceitos, faça:
11	Se o conceito for <i>conceito parâmetro</i> e estiver associado à ação como parâmetro, então:
12	Associar o <i>conceito parâmetro</i> à ação.
13	Fim do se
14	Fim do para
15	⇒ Os parâmetros <i>direção: frente</i> e <i>tempo: 5 segundos</i> são inseridos como parâmetros para o comando <i>movimentar</i> .
16	Fim do se
17	Fim do se

Tabela 5.4. Passos para o reconhecimento de uma ação na oração “Logo em seguida, vire para a direita.”.

Passo	Descrição
1	Extrair os conceitos presentes na oração “Logo em seguida, vire para a direita.”.
2	⇒ Conceitos identificados: <i>movimentar</i> e <i>direita</i> .
3	Localizar um <i>conceito comando</i> na coleção de conceitos.
4	⇒ Conceito comando identificado: <i>movimentar</i>
5	Se encontrar um <i>conceito comando</i> , então:
6	Armazenar o conceito comando sob forma de ação.
7	⇒ Criar um item na lista de ações que o robô deve executar
8	⇒ Comando <i>movimentar</i> inserido na lista de ações, aguardando parâmetros.
9	Se a ação identificada requer algum parâmetro, então:
10	Para cada conceito na coleção de conceitos, faça:
11	Se o conceito for <i>conceito parâmetro</i> e estiver associado à ação como parâmetro, então:
12	Associar o <i>conceito parâmetro</i> à ação.
13	Fim do se
14	Fim do para
15	⇒ Parâmetro <i>direção: frente</i> identificado, mas o parâmetro <i>tempo</i> não foi identificado. Assumir o valor <i>default</i> para o parâmetro <i>tempo</i> . Parâmetro <i>tempo: 3 segundos</i> .
16	Fim do se
17	Fim do se

5.1 EXEMPLO DETALHADO

Tabela 5.5. Passos para o reconhecimento de uma ação na oração “Depois, vá para a esquerda.”.

Passo	Descrição
1	Extrair os conceitos presentes na oração “Depois, vá para a esquerda.”.
2	⇒ Conceitos identificados: <i>movimentar</i> e <i>esquerda</i> .
3	Localizar um <i>conceito comando</i> na coleção de conceitos.
4	⇒ Conceito comando identificado: <i>movimentar</i>
5	Se encontrar um <i>conceito comando</i> , então:
6	Armazenar o conceito comando sob forma de ação.
7	⇒ Criar um item na lista de ações que o robô deve executar
8	⇒ Comando <i>movimentar</i> inserido na lista de ações, aguardando parâmetros.
9	Se a ação identificada requer algum parâmetro, então:
10	Para cada conceito na coleção de conceitos, faça:
11	Se o conceito for <i>conceito parâmetro</i> e estiver associado à ação como parâmetro, então:
12	Associar o <i>conceito parâmetro</i> à ação.
13	Fim do se
14	Fim do para
15	⇒ Parâmetro <i>direção</i> : <i>esquerda</i> identificado, mas o parâmetro <i>tempo</i> não foi identificado. Assumir o valor <i>default</i> para o parâmetro <i>tempo</i> . Parâmetro <i>tempo</i> : <i>3 segundos</i> .
16	Fim do se
17	Fim do se

Tabela 5.6. Passos para o reconhecimento de uma ação na oração “Procure a cor amarela no ambiente por 120 segundos, no máximo.”.

Passo	Descrição
1	Extrair os conceitos presentes na oração “Procure a cor amarela no ambiente por 120 segundos, no máximo.”.
2	⇒ Conceitos identificados: <i>procurar</i> , <i>cor</i> , <i>amarelo</i> e <i>tempo</i> .
3	Localizar um <i>conceito comando</i> na coleção de conceitos.
4	⇒ Conceito comando identificado: <i>procurar</i>
5	Se encontrar um <i>conceito comando</i> , então:
6	Armazenar o conceito comando sob forma de ação.
7	⇒ Criar um item na lista de ações que o robô deve executar
8	⇒ Comando <i>procurar</i> inserido na lista de ações, aguardando parâmetros.
9	Se a ação identificada requer algum parâmetro, então:
10	Para cada conceito na coleção de conceitos, faça:
11	Se o conceito for <i>conceito parâmetro</i> e estiver associado à ação como parâmetro, então:
12	Associar o <i>conceito parâmetro</i> à ação.
13	Fim do se
14	Fim do para
15	⇒ Os parâmetros <i>cor</i> : <i>amarelo</i> e <i>tempo</i> : <i>120 segundos</i> são inseridos como parâmetros para o comando <i>procurar</i> .
16	Fim do se
17	Fim do se

5.1 EXEMPLO DETALHADO

Tabela 5.7. Passos para o reconhecimento de uma ação na oração “Movimente a câmera para baixo.”.

Passo	Descrição
1	Extraír os conceitos presentes na oração “Movimente a câmera para baixo.”
2	⇒ Conceitos identificados: <i>movimentar câmera</i> e <i>baixo</i> .
3	Localizar um <i>conceito comando</i> na coleção de conceitos.
4	⇒ Conceito comando identificado: <i>movimentar câmera</i>
5	Se encontrar um <i>conceito comando</i> , então:
6	Armazenar o conceito comando sob forma de ação.
7	⇒ Criar um item na lista de ações que o robô deve executar
8	⇒ Comando <i>movimentar câmera</i> inserido na lista de ações, aguardando parâmetros.
9	Se a ação identificada requer algum parâmetro, então:
10	Para cada conceito na coleção de conceitos, faça:
11	Se o conceito for <i>conceito parâmetro</i> e estiver associado à ação como parâmetro, então:
12	Associar o <i>conceito parâmetro</i> à ação.
13	Fim do se
14	Fim do para
15	⇒ Parâmetro <i>direção</i> : <i>baixo</i> identificado, mas o parâmetro <i>ângulo</i> não foi identificado. Assumir o valor <i>default</i> para o parâmetro <i>ângulo</i> . Parâmetro <i>ângulo</i> : <i>extremo da direção</i> .
16	Fim do se
17	Fim do se

Tabela 5.8. Passos para o reconhecimento de uma ação na oração “Capture uma imagem do ambiente e armazene.”.

Passo	Descrição
1	Extraír os conceitos presentes na oração “Capture uma imagem do ambiente e armazene.”.
2	⇒ Conceitos identificados: <i>capturar</i> e <i>imagem</i> .
3	Localizar um <i>conceito comando</i> na coleção de conceitos.
4	⇒ Conceito comando identificado: <i>capturar</i>
5	Se encontrar um <i>conceito comando</i> , então:
6	Armazenar o conceito comando sob forma de ação.
7	⇒ Criar um item na lista de ações que o robô deve executar
8	⇒ Comando <i>capturar</i> inserido na lista de ações, aguardando parâmetros.
9	Se a ação identificada requer algum parâmetro, então:
10	Para cada conceito na coleção de conceitos, faça:
11	Se o conceito for <i>conceito parâmetro</i> e estiver associado à ação como parâmetro, então:
12	Associar o <i>conceito parâmetro</i> à ação.
13	Fim do se
14	Fim do para
15	Fim do se
16	Fim do se

A tabela 5.9 abrange todas as ações identificadas no texto da tabela 5.1.

A execução das ações baseia-se na ordem de inserção na árvore de ações identifi-

Tabela 5.9. Ações reconhecidas.

Ação 1: Movimentar; Direção: frente Tempo: 5 segundos
Ação 2: Movimentar Direção: direita Tempo: 3 segundos(default)
Ação 3: Movimentar Direção: esquerda Tempo: 3 segundos(default)
Ação 4: Procurar Cor: amarelo Tempo: 120 segundos
Ação 5: Movimentar Câmera; Direção: baixo Ângulo: extremo da direção.
Ação 6: Capturar imagem.

casas. As ações são executadas uma por vez, onde a cada tentativa de execução um registro da execução é realizado. Se a ação for executada com êxito, a palavra *OK* é inserida abaixo da ação no registro. Em caso de falha, a palavra *Falhou* é inserida abaixo da ação no registro.

A interface para entrada da tarefa em linguagem natural utilizada pelo usuário pode ser observada na figura 5.1. Nesta figura também estão expostas as ações identificadas no texto e o registro da execução das ações.

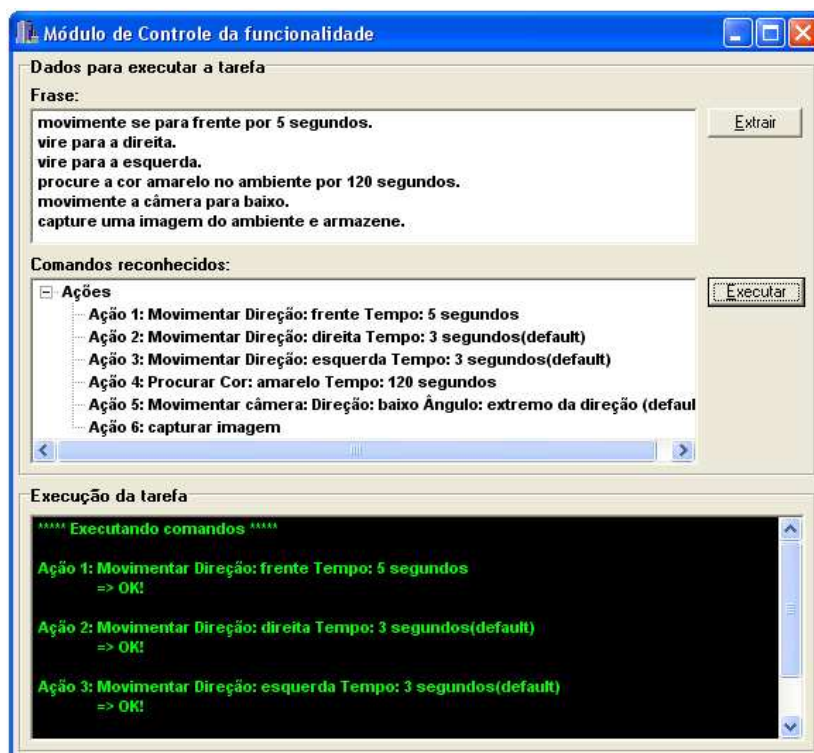


Figura 5.1. Interface para execução da tarefa em linguagem natural.

5.2 OUTROS EXEMPLOS

Nesta seção são apresentados exemplos de tarefas que o robô deve executar, seguidas das ações identificadas nas tarefas.

A tabela 5.10 apresenta um exemplo de descrição de tarefa.

Tabela 5.10. Exemplo de tarefa submetida à interface.

Robô, caminhe para frente por 30 segundos. Vire para a direita. Procure um objeto verde. Siga o objeto verde por 300 segundos.

A tabela 5.11 apresenta as ações identificadas no texto da tabela 5.10.

Tabela 5.11. Ações reconhecidas na tarefa descrita na tabela 5.10.

Ação 1: Movimentar; Direção: frente Tempo: 30 segundos Ação 2: Movimentar Direção: direita Tempo: 3 segundos(default) Ação 3: Procurar Cor: verde Tempo: 120 segundos(default) Ação 4: Seguir Cor: verde Tempo: 300 segundos

A tabela 5.12 apresenta outro exemplo de descrição de tarefa.

Tabela 5.12. Descrição de tarefa submetida à interface.

Localize a cor vermelha no ambiente. Capture uma imagem do objeto achado. Movimente-se para a esquerda. Movimente a câmera para cima para o ângulo 30°. Procure um objeto amarelo por 30 segundos. Siga em frente por 10 segundos e pare.
--

A tabela 5.13 apresenta as ações identificadas no texto da tabela 5.12.

Tabela 5.13. Ações reconhecidas na tarefa descrita na tabela 5.12.

Ação 1: Procurar Cor: vermelho Tempo: 120 segundos(default)
Ação 2: Capturar Imagem
Ação 3: Movimentar Direção: esquerda Tempo: 3 segundos(default)
Ação 4: Movimentar câmera Direção: cima Ângulo: 30°
Ação 4: Procurar Cor: amarelo Tempo: 30 segundos
Ação 5: Movimentar Direção: frente Tempo: 10 segundos

A tabela 5.14 apresenta mais um exemplo de descrição de tarefa.

Tabela 5.14. Descrição de tarefa submetida à interface.

Capture uma imagem do ambiente.
Siga o objeto que esta na frente da câmera por 40 segundos.
Armazene uma imagem do ambiente.
Depois, ande para a trás por 15 segundos.
Vire para a direita, vire para a esquerda.
Siga o objeto que esta na frente da câmera por 20 segundos e pare.

A tabela 5.15 apresenta as ações identificadas no texto da tabela 5.14.

Tabela 5.15. Ações reconhecidas na tarefa descrita na tabela 5.14.

Ação 1: Capturar imagem.
Ação 2: Seguir objeto apresentado Tempo: 40 segundos
Ação 3: Capturar imagem.
Ação 4: Movimentar Direção: trás Tempo: 15 segundos
Ação 5: Movimentar Direção: direita Tempo: 3 segundos(default)
Ação 6: Movimentar Direção: esquerda Tempo: 3 segundos(default)
Ação 7: Seguir objeto apresentado Tempo: 20 segundos

5.3 LIMITAÇÕES DA INTERFACE PROPOSTA

As orações dos diversos exemplos apresentados ao longo deste capítulo possuem seus objetivos bem definidos. Isto é, as orações das tarefas não possuem contradições, nem negações de comandos.

Tarefas que não possuem objetivos bem definidos possuem difícil interpretação quando utilizada a abordagem proposta para a interface, onde conceitos são utilizados

para identificar comandos. A identificação de conceitos é realizada usando regras pré-estabelecidas, as quais possuem termos positivos e negativos, conforme descrito no capítulo 3. A depender das palavras da oração e das regras definidas para um conceito, pode ocorrer que um conceito seja identificado de forma incorreta na oração, levando a um erro na identificação do comando que o robô deve executar. Por exemplo, considere a seguinte oração:

- “Robô, não vá para frente, vá para trás.”

Nesta oração é possível observar que utilizando a tabela 4.1 apresentada no capítulo 4, dois comandos *movimentar* serão identificados. O primeiro comando *movimentar* é reconhecido após o conceito *movimentar*, que é um *conceito comando*, é identificado através da regra positiva que contém o termo *vá*. Este comando possui como parâmetro a direção *frente*. O segundo comando reconhecido na oração é novamente o comando *movimentar*, porém com a direção *trás*. A identificação dos dois comandos *movimentar* não está correto, pois no primeiro caso, o palavra *não* é utilizada para negar a execução do comando *movimentar*.

Problemas com a característica de negar a execução de um comando podem ser resolvidos com a inserção de regras de negação. Para resolver o problema na identificação do conceito *movimentar* na oração “Robô, não vá para frente, vá para trás.”, é necessário inserir uma regra negativa com o termo *não*, associada à regra de afirmação que possui o termo *vá*. A nova lista de regras para o conceito *movimentar* é apresentado na tabela 5.16. Desta forma, após identificar na oração a presença da regra de afirmação que possui o termo *vá*, será realizado a procura da regra de negação que possui o termo *não*. Ao identificar a regra negativa associada à regra positiva, o conceito *movimentar* identificado de forma errada não é mais reconhecido, pois nenhuma regra positiva é encontrada.

Outro problema encontrado na interpretação das tarefas são orações onde não são encontrados *conceitos comandos*. Neste caso, a interface ignora a oração pois não

Tabela 5.16. Novas regras para o conceito movimentar.

Conceito <i>movimentar</i>	
Regras positivas	Regras negativas
caminhar	
correr	
ande	
andar	
movimentar	
movimente	
vá	<i>não</i>
siga frente	
siga adiante	
seguir frente	

é possível reconhecer uma ação. Um exemplo deste caso é apresentado na seguinte oração:

- Robô, ir em frente parece um bom caminho.

Nesta oração nenhuma ação é reconhecida, pois não existe um *conceito comando* identificado, embora seja encontrado o *conceito parâmetro frente* associado ao *conceito comando movimentar*. A solução para problemas com essa característica foi a implementação de uma regra no algoritmo de identificação de ações, no qual após o algoritmo de identificação de conceitos extrair os conceitos, é realizado uma verificação a procura de um *conceito comando*. Caso nenhum seja encontrado, a partir dos *conceitos parâmetros* identificados na oração é realizada uma busca nos *conceitos parâmetros* associados aos *conceitos comandos*, visando encontrar um único *conceito comando* que possua os *conceitos parâmetros* da oração. Se encontrar um único *conceito comando*, este é definido como a ação que deve ser executada, tendo como seus parâmetros os *conceitos parâmetros* encontrados na oração. Esta regra é acionada após não ser encontrado um *conceito comando* uma oração.

Caso não seja possível identificar uma ação na oração, a interface emite um aviso informando que a oração não possui dados para a identificação de uma ação. Por exemplo, as orações da tabela 5.17 não são interpretadas corretamente.

Tabela 5.17. Orações que são ignoradas pela interface.

Oração 1: Robô, vá para frente procurando um objeto da cor vermelha.
Oração 2: Vá para frente e para trás.
Oração 3: Procure todos os objetos da cor azul.
Oração 4: Movimente-se no sentido da cor amarela.
Oração 5: Robô, nunca ande para trás, ande para trás.

A primeira oração da tabela 5.17 possui mais de um *conceito comando* identificado. Neste caso os conceitos reconhecidos foram *movimentar* e *procurar*, bem como os *conceitos parâmetros frente, objeto e vermelho*. Ao analisar os conceito identificados, a interface é capaz de reconhecer a ação *movimentar* com a direção para frente com o tempo *default* de 3 segundos e a ação *procurar* com o parâmetro vermelho e tempo *default* 120 segundos. Orações que possuam mais de um *conceito comando* podem ser recusadas visando não executar uma ação de forma errada ou faltando parâmetros devido ao uso comum de *conceitos parâmetros* nos *conceitos comandos* envolvidos. Nesta oração os parâmetros dos *conceitos comandos* são distintos, possibilitando o reconhecimento das ações.

A segunda oração é ignorada pois possui mais de um *conceito parâmetro* associado à direção do movimento, sendo eles *frente* e *trás*.

A tarefa da terceira oração é procurar por todos os objetos da cor azul no ambiente, porém a interface procura por apenas um objeto da cor azul e pára a busca, não completando corretamente o objetivo da tarefa.

A interpretação da quarta tarefa é realizada de forma incorreta, pois é identificada apenas a ação *movimentar* possuindo a direção para frente por 3 segundos como *default*, pois nenhum parâmetro foi identificado. Mas o correto a ser executado era identificar uma cor amarela no ambiente e depois o robô movimentar-se na direção da cor. Neste caso, é possível criar um *conceito comando* e uma ação específicos para cobrir as variações deste tipo de oração.

A quinta oração ocorre um erro de semântica, que não é tratado pela técnica utilizada na interface. Neste caso, se for associada uma regra negativa com o termo *nunca*

em todas as regras do conceito *movimentar*, o conceito não é reconhecido no fragmento da oração que possui o termo *nunca*. Porém o outro fragmento que possui termos que validam a presença do conceito *movimentar* é reconhecido como a ação movimentar para trás, contradizendo a afirmação presente na oração de nunca movimentar para trás.

5.4 CONSIDERAÇÕES FINAIS

Com a interface proposta neste trabalho, o usuário apenas descreve os passos para a execução de uma tarefa que o protótipo robótico deve executar. A descrição da tarefa é realizada na língua portuguesa. Desta forma, torna possível manipular um robô sem a necessidade de ter que aprender uma nova linguagem, repleta de formalismo e complexa, para a interação com o robô. O usuário utiliza a linguagem escrita ao qual está familiarizado, a própria linguagem natural usada para sua comunicação.

A interface em linguagem natural descrita, mostra sua potencialidade na utilização de uma língua sem ter a necessidade de informar como as palavras são formadas, como as palavras se relacionam em uma oração, além de como as orações se relacionam, tornando desnecessário que o usuário da interface tenha um conhecimento apurado do seu idioma.

Os testes realizados na interface robótica proposta mostraram que o desenvolvimento da interface baseada em linguagem natural escrita é um caminho viável para a construção de interfaces entre seres humanos e máquinas mais amigáveis.

A utilização da técnica de Descoberta de Conhecimento em Texto baseado em Conceitos, torna a configuração da interface simplificada, pois o usuário informa apenas os termos que compõe as regras de um conceito, não há a necessidade de informar a forma com a qual as palavras se relacionam. Os conceitos cadastrados são associados às ações que o robô está capacitado a executar.

Como as regras para a descrição de conceitos utilizam apenas termos, sem qualquer

análise sobre a estrutura do idioma, há a possibilidade de criar novas regras sobre um conceito utilizando uma outra língua e até mesmo diversas línguas, tornando a interface multilingual para a interação com o robô. Como exemplo, a tabela 5.18 apresenta novas regras para a definição do conceito *movimentar*, possuindo regras com termos na língua inglesa, francesa e espanhola. Tornando multilingual a definição do conceito e uma ação associada, neste caso a ação movimentar.

Tabela 5.18. Definição do conceito movimentar em diversas línguas.

Conceito	Regras positivas
movimentar	caminhar correr andar movimentar movimento vá siga frente siga adiante seguir frente <i>move</i> (<i>movimentar em inglês</i>) <i>go ahead</i> (<i>siga adiante em inglês</i>) <i>mouvement</i> (<i>movimento em francês</i>) <i>mueva</i> (<i>movimento em espanhol</i>)

Limitações foram impostas ao desenvolvimento da proposta desta interface robótica, como por exemplo, o reconhecimento de ações e seus parâmetros é realizado oração a oração de forma isolada. Caso uma oração possua algum conhecimento que deveria ser compartilhado com outras orações do texto, não será possível pois o conhecimento é acessível apenas na oração ao qual foi descoberto. Esta limitação é devida a técnica de Descoberta de Conhecimento baseado em Conceito utilizada na interface, que extrai conceitos das orações sem relacionar aos outros encontrados. Desta forma, pode causar erros semânticos e pragmáticos no conhecimento descoberto, impactando na execução das ações. Esta limitação foi imposta visando retirar a complexidade inerente à análise do texto completo.

Outra limitação imposta à proposta é a quantidade de ações disponíveis para o protótipo robótico executar, sendo respectivamente: navegar pelo ambiente, localizar

um objeto com uma cor especificada, seguir um objeto com uma cor especificada, movimentar a câmera, capturar uma imagem do ambiente. Neste caso, a limitação é devido às possibilidades de ações encontradas utilizando a estrutura do protótipo robótico, a qual foi desenvolvida com o intuito de validar a interface robótica.

Algumas ações que são reconhecidas e executadas pela interface proposta dependem do parâmetro *tempo*, que foi limitado ao reconhecimento temporal baseado em segundos para a execução das ações.

CONCLUSÕES

As interfaces entre os seres humanos e as máquinas evoluem a cada dia para formas mais amigáveis e elementares. Interfaces mais amigáveis possibilitam sua utilização por usuários sem treinamentos técnicos específicos.

Um exemplo de interface amigável entre homem e máquina é a utilização da linguagem natural para a descrição de atividades que a máquina deve executar. O Processamento de Linguagem Natural é o caminho mais espontâneo para o desenvolvimento desta interface; porém, a aplicação da linguagem natural exige um vasto conhecimento sobre o idioma. Devido aos tratamentos lingüísticos, a utilização de técnicas de PLN exige um árduo esforço de implementação, além da complexidade computacional para interpretar um texto em linguagem natural.

Uma alternativa para tratar a linguagem natural é a utilização de técnicas de Descoberta de Conhecimento em Texto com base em Conceito. Com o uso desta técnica, o conhecimento aprofundado sobre um idioma não é necessário, pois a abordagem tem o intuito de extrair conceitos encontrados no texto, o que exige conhecimento sobre o domínio no qual a linguagem natural está inserida.

Considerando a oração “Robô, procure um objeto com a cor amarela no ambiente”, o reconhecimento da ação utilizando a técnica baseada em conceitos diminui significativamente a complexidade de implementação da interface. Neste exemplo de oração, a técnica baseada em conceitos pode extrair os conceitos *procurar*, *objeto* e *amarelo*, desde que as regras que formam estes conceitos sejam definidas. Após a identificação dos conceitos, a interface pode associá-los a uma ação que o robô é capaz de executar

para o cumprimento da tarefa transmitida em linguagem natural.

Algumas interfaces utilizam comando de voz para a comunicação entre os seres humanos e as máquinas. A maioria destas interfaces, no entanto, apresenta complexidade computacional na interpretação dos sons de vozes de diferentes pessoas transmitindo comandos. Uma alternativa é utilizar a linguagem natural escrita para uma interface homem-máquina, possibilitando a expansão do universo de discurso em relação às restrições encontradas no reconhecimento de comando de voz.

A implementação da interface utilizando técnica de Descoberta de Conhecimento em Texto baseada em Conceito se mostrou consistente na identificação de ações que o protótipo robótico é capaz de executar, pois orações sintaticamente incorretas foram interpretadas e as ações associadas foram reconhecidas corretamente. Os exemplos de tarefas em linguagem natural e as ações corretamente reconhecidas demonstram a potencialidade desta interface empregada em máquinas.

6.1 TRABALHOS FUTUROS

Os conhecimentos produzidos a partir deste trabalho, possibilitam o desenvolvimento de diversos trabalhos baseados em interfaces mais amigáveis entre seres humanos e máquinas.

Uma evolução para este trabalho é a ampliação do universo de ações que o protótipo robótico pode executar. Como exemplo, incorporar técnicas inteligentes para o reconhecimento de objetos. Com isso, os comandos *procurar* e *seguir* podem ter como parâmetro o objeto que o protótipo robótico deve encontrar no ambiente e conseqüentemente identificá-los e executar suas funcionalidades. Outro exemplo é o desenvolvimento e incorporação de uma garra ao protótipo, provendo a capacidade de pegar objetos no ambiente.

Visando uma melhor movimentação do protótipo robótico, sugere-se a incorporação da Lógica Fuzzy no controle da locomoção do protótipo robótico. Além de componentes

eletrônicos necessários para captar e monitorar a movimentação do protótipo robótico.

A técnica de Descoberta de Conhecimento em Texto baseada em Conceitos utilizada na interface robótica deste trabalho foi limitada a extrair conhecimento oração a oração, visando extrair as ações da tarefa definida passo a passo pelo usuário. Sendo assim, o conhecimento extraído é restrito às orações analisadas, não estando disponíveis às demais orações do texto. Esta limitação pode gerar erros semânticos e pragmáticos na interpretação de uma ação reconhecida em uma oração. Tornando necessário a implementação de métodos que analisem os conhecimentos descobertos e os disponibilizem as demais conhecimentos que serão extraídos das orações seguintes. Com isso, diminuir ou até mesmo excluir a possibilidade de ocorrer erros semânticos e pragmáticos.

A interface robótica proposta e desenvolvida neste trabalho, pode ser inserida em qualquer máquina que opere baseada em ações ou comandos de forma direta ou indireta. Como exemplo, a interface proposta pode ser inserida em manipuladores robóticos, visando simplificar a definição dos comandos a serem utilizados. Os manipuladores robóticos geralmente têm suas atuações definidas na linguagem específica do seu fabricante, tornando necessário que seus operadores aprendam uma nova linguagem, repleta de formalismos e complexa, para os utilizar. Geralmente a atuação de um manipulador robótico é baseada na movimentação sobre seus eixos e na utilização da sua garra. O universo do manipulador robótico pode ser mapeado na interface proposta neste trabalho. Com isso, os operadores dos manipuladores robóticos podem definir as suas ações baseadas em conhecimentos mais genéricos dos manipuladores robóticos, não se importando com os comandos específicos dos fabricantes.

REFERÊNCIA

AMARAL, M. A.; BARRIVIERA, R.; TEIXEIRA, E. C. Reconhecimento de voz para automação residencial baseado em agentes inteligentes. *Revista Eletrônica de Sistemas de Informação - Faculdade Cenecista Presidente Kennedy*, v. 3, n. 1, 2004.

BORGES, L. A. *Sistemas de adaptação ao locutor utilizando autovozes*. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 2001.

CEGALLA, D. P. *Novíssima Gramática da Língua Portuguesa*. [S.l.]: Editora Nacional, 1997. ISBN 8504007898.

CHOWDHURY, G. Natural language processing. *Annual Review of Information Science and Technology*, v. 37, p. 51–89, 2003.

CLIFTON, C.; STEINHEISER, R. Data mining on text. *Proceedings - IEEE Computer Society's International Computer Software and Applications Conference*, p. 630–635, 1998.

FEDELI, R. D.; POLLINI, E.; PERES, F. E. *Introdução à Ciência da Computação*. [S.l.]: Pioneira Thomson Learning, 2003.

FELDMAN, R.; HIRSH, H. Exploiting background information in knowledge discovery from text. *Journal of Intelligent Information Systems*, v. 9, n. 1, p. 83–97, 1997.

FERREIRA, A. B. de H. *Novo Dicionário Aurélio*. [S.l.]: Editora Nova Fronteira S.A., 1975.

FROMM, P.; DREWS, P. Natural language processing for dynamic environments. *IECON Proceedings (Industrial Electronics Conference)*, v. 4, p. 2018–2021, 1998.

FU, Y. Data mining - tasks, techniques and applications. *IEEE Potentials*, v. 16, n. 4, p. 18–20, 1997.

GIMENES, E. “Data Mining - Data Warehouse” *A Importância da mineração de Dados em Tomadas de Decisões*. Dissertação (Mestrado) — Faculdade de Tecnologia de Taquaritinga, 2000.

GONZALEZ, M.; LIMA, V. L. S. de. Sintagma nominal em estrutura hierárquica temática na recuperação de informação. *XXI Congresso da Sociedade Brasileira da Computação - Encontro Nacional de Inteligência Artificial (ENIA)*, 2001.

HAHN, U.; ROMACKER, M. Syndikate - generating text knowledge bases from natural language texts. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, v. 5, p. 918–923, 1999.

HARRIS, L. R. *Expert Systems and Artificial Intelligence*. [S.l.]: Howard W. Sams and Company, 1988.

HONKELA, T. et al. Exploration of full-text databases with self-organizing maps. *Proceedings of the ICNN96, International Conference on Neural Networks*, IEEE Service Center, Piscataway, NJ, I, p. 56–61, 1996.

KITANO, H. et al. Robocup: The robot world cup initiative. In: *AGENTS '97: Proceedings of the first international conference on Autonomous agents*. New York, NY, USA: ACM Press, 1997. p. 340–347. ISBN 0-89791-877-0.

KNIGHT, K.; RICH, E. *Inteligência Artificial*. 2a. ed. [S.l.]: McGraw-Hill, 1993. ISBN 155860221.

KOHONEN, T. The self-organizing map. *IEEE Transactions on Neural Networks*, v. 1, n. 1, p. 93–99, 1990.

KROVETZ, R.; CROFT, W. B. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, v. 10, n. 2, p. 115–141, 1992.

LADEIRA, A. P. et al. Uma agente inteligente moderador de listas de discussão. *II Seminário Nacional de Tecnologia para EAD - Universidade Federal de Uberlândia*, 2002.

LEVINE, R. I.; DRANG, D. G.; EDELSON, B. *Inteligência artificial e sistemas especialistas*. 1a. ed. [S.l.: s.n.], 1988.

LOH, S. *Abordagem Baseada em Conceitos para Descoberta de Conhecimento em Textos*. Dissertação (Mestrado) — Universidade Federal do Rio grande do Sul, Porto Alegre, 2001.

LOH, S. et al. Formalizando e explorando conhecimento tácito com a tecnologia de text mining para inteligência. *ISKM/DM '2001*, v. 1, p. 379–397, Agosto 2001.

MALLETT, D.; ELDING, J.; NASCIMENTO, M. Information-content based sentence extraction for text summarization. *International Conference on Information Technology: Coding Computing*, v. 32, n. 2-3, p. 214–218, 2004.

NETO, J. J.; MORAES, M. de. Formalismo adaptativo aplicado ao reconhecimento de linguagem natural. *CISCI - Conferencia Iberoamericana en Sistemas, Cibernética e Informática*, Orlando, Florida, Julho 2002.

PEREIRA, S. do L. *Processamento de Linguagem Natural*. 2006. Disponível em: <<http://www.ime.usp.br/slago/pln.pdf>>. Acesso em: 07 de março de 2006.

PERRIN, P.; PETRY, F. E. Extraction and representation of contextual information for knowledge discovery in texts. *Information Sciences*, v. 151, p. 125–152, 2003.

PINILLA, A.; RIGONI, C.; INDIANI, M. T. *Português Ensino a Distância*. Universidade Federal do Rio de Janeiro, 2006. Disponível em: <<http://acd.ufrj.br/pead/>>. Acesso em: 01 de julho de 2006.

RAU, L. F.; JACOBS, P. S. Integrating top-down and bottom-up strategies in a text processing system. *Proc. of the Second Conference on Applied Natural Language Processing*, p. 129–135, 1988.

REZENDE, S. O. Mineração de dados. *XXV Congresso da Sociedade Brasileira de Computação*, São Leopoldo - RS, p. 397–433, 2003.

RICH, E. *Artificial Intelligence*. 1a. ed. [S.l.]: McGraw-Hill, 1986. ISBN 0070522634.

RILOFF, E.; LEHNERT, W. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, v. 12, n. 2, p. 296–333, 1994.

RUNKLER, T.; BEZDEK, J. Web mining with relational clustering. *International Journal of Approximate Reasoning*, v. 32, n. 2-3, p. 296–333, 2003.

SANTOS, A. dos; PELIZZONI, J. M. *Introdução ao Processamento de Linguagem Natural*. 2005. Disponível em: <LABIC-ICMC-USP - <http://labic.icmc.usp.br/portugues/SIAE/default.htm>>. Acesso em: 02 de setembro de 2005.

SCHILDT, H. *Inteligência Artificial Utilizando Linguagem C*. 1a. ed. [S.l.]: McGraw Hill, 1989.

SFERRA, H.; CORRÊA Ângela. Conceitos e aplicações de data mining. *Revista de Ciência e Tecnologia*, v. 11, n. 22, p. 19–34, 2003.

TAN, A.-H. Text mining: The state of the art and the challenges. *Pacific-Asia Workshop on Knowledge Discovery from Advanced Databases-PAKDD 99*, p. 65–70, 1999.

VIEIRA, R.; LIMA, V. L. S. de. Linguística computacional: princípios e aplicações. *XXI Congresso da Sociedade Brasileira de Computação*, v. 2, p. 47–88, 2001.

WINSTON, P. H. *Inteligência Artificial*. [S.l.]: Livros Técnicos e Científicos Editora LTDA, 1987. ISBN 0201518767.

WIVES, L. K. *Tecnologias de Descoberta de Conhecimento em Textos Aplicados a Inteligência Artificial*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, 2002.

WIVES, L. K.; LOH, S. Tecnologias de descoberta de conhecimento em informações textuais. *Oficina de Inteligência Artificial - OIA*, n. III, p. 28–48, 1999.

WIVES, L. K.; LOH, S.; OLIVEIRA, J. P. Descoberta proativa de conhecimento em coleções textuais: Iniciando sem hipóteses. *IV Oficina de inteligência artificial*, p. 143–154, 2000.

YANG, H.-C.; LEE, C.-H. Automatic category generation for text documents by self-organizing maps. *Proceedings of the International Joint Conference on Neural Networks*, v. 3, p. 581–586, 2000.

ESTRUTURA DO ROBÔ

A.1 DETALHAMENTO DA ESTRUTURA DO ROBÔ

Para a validação deste experimento, um protótipo de um robô móvel foi desenvolvido. Os pré-requisitos do projeto do robô incluíram características como baixo peso, dimensões compactas, facilidade de manobra e capacidade de comportar os componentes eletrônicos necessários a fim de torná-lo autônomo.

A meta de baixo peso foi atingida com a utilização de alumínio em toda a estrutura. O primeiro protótipo foi construído utilizando-se perfis em L, também conhecidos como cantoneiras. Os perfis foram unidos com rebites, assim como toda a estrutura do robô. A figura A.1 exibe o quadro em formato retangular utilizado na estrutura do robô.

Na estrutura do robô, os dois quadros mostrados na figura A.1 foram unidos com chapas que serviram de suporte para o sistema de locomoção. A figura A.2 mostra a primeira estrutura construída para o robô.

Para o mecanismo de locomoção, um sistema com polia e correia em V foi utilizado, além de dois motores-redutores de acionamento de vidro-elétrico automotivo. A figura A.3 mostra os componentes do sistema de locomoção, os quais estão listados na tabela A.1.

A montagem completa do primeiro protótipo do robô pode ser vista na figura A.4.

A fim de aumentar a rigidez e o alinhamento do quadro, as cantoneiras laterais nos quadros foram substituídas por tubos de perfil quadrado com mesma espessura e dimensões da cantoneira, que podem ser vistos na figura A.5. Uma cantoneira também

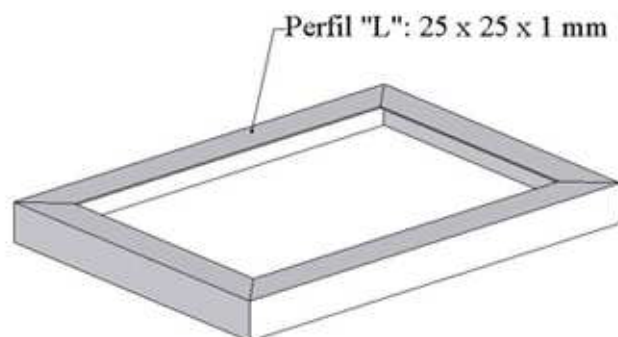


Figura A.1. Quadro com cantoneiras.

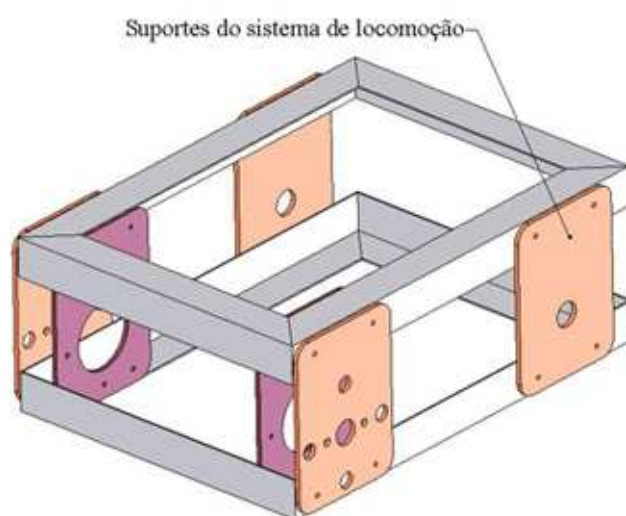


Figura A.2. Primeira estrutura do robô.

foi adicionada para a união dos dois quadros, conforme exibido na figura A.6. Com estas modificações, a estrutura do robô mostrou-se adequada para os propósitos deste projeto. A montagem completa do segundo protótipo pode ser vista na figura A.7.

Há duas possibilidades para manobrar o robô. A primeira é desligar um motor e acionar o outro, utilizando a correia parada como pivô do giro. A segunda é girar os

Tabela A.1. Lista dos componentes do sistema de locomoção.

1 - motor-reductor	6 - correia em V
2 - eixo motriz - aço SAE-1045	7 - Manga de eixo polia movida;
3 - mancal eixo motriz - nylon	8 - Espaçador;
4 - parafuso do mancal	9- Polia movida;
5 - polia em V de 5- alumínio	10 - Anel elástico.

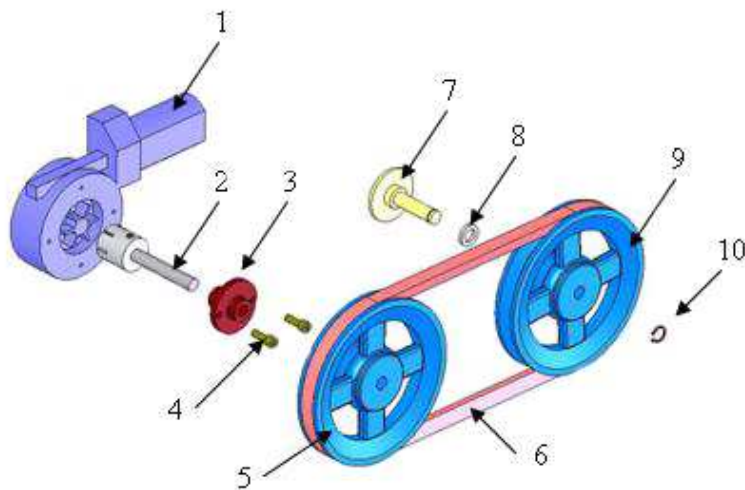


Figura A.3. Componentes do sistema de locomoção.

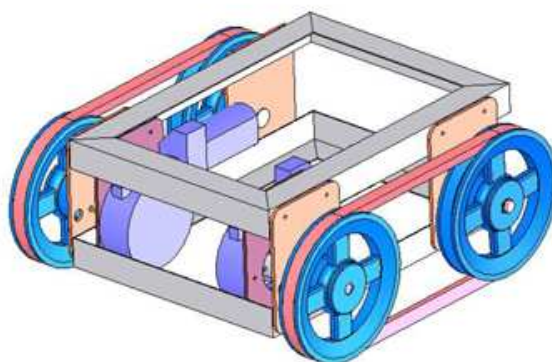


Figura A.4. Primeiro protótipo montado.

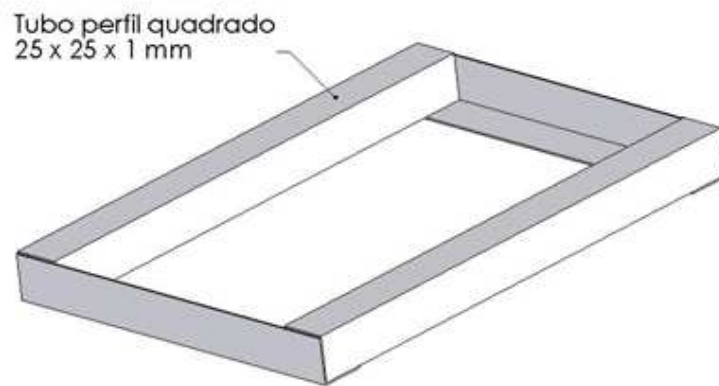


Figura A.5. Segundo quadro do robô.

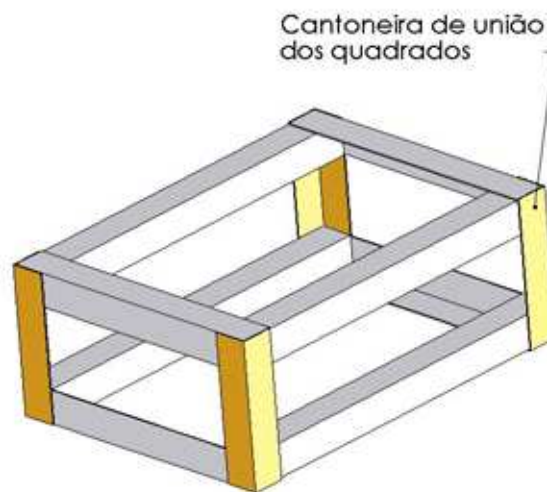


Figura A.6. Estrutura montada do robô.

motores em sentidos contrários, utilizando o centro do robô como pivô do giro.

A figura A.8 mostra as duas possibilidades.

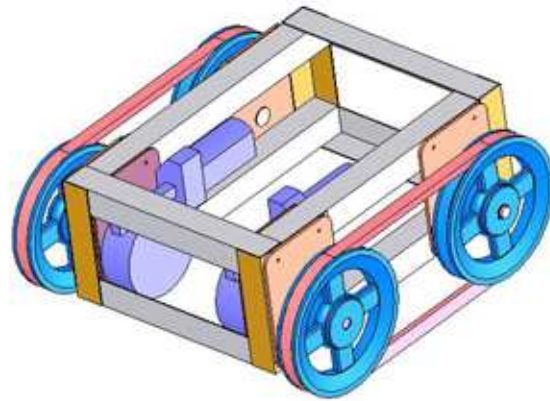


Figura A.7. Montagem do segundo robô.

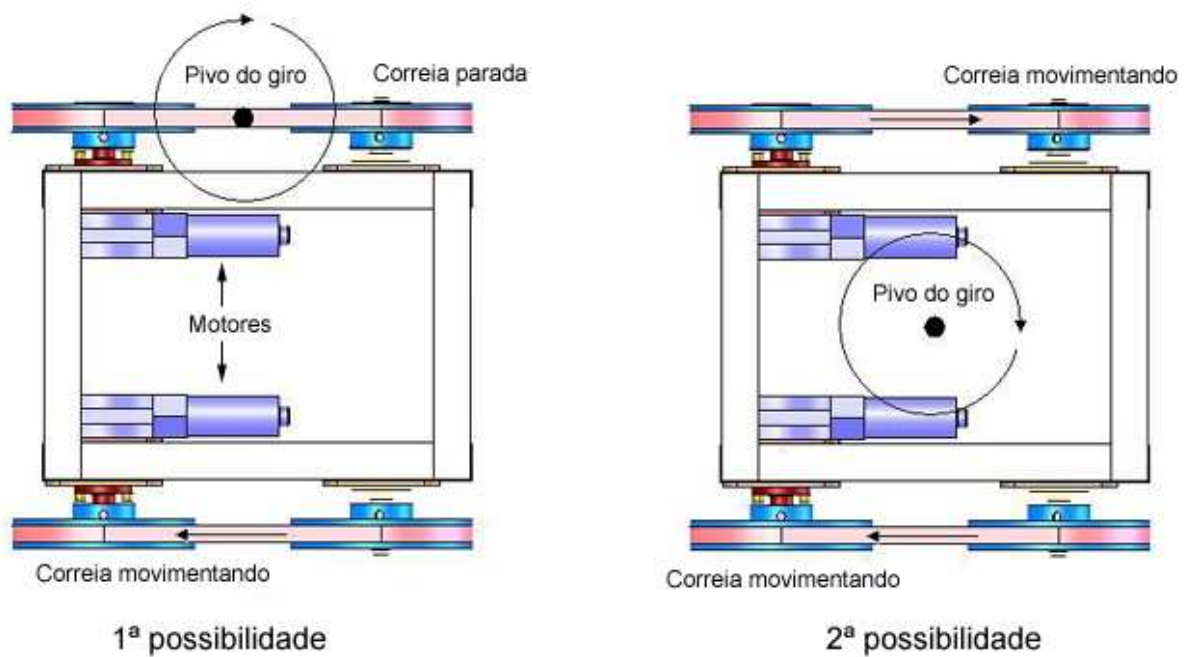


Figura A.8. Possibilidades de giro do Robô.

A.1.1 Mecanismo de movimentação da câmera

O mecanismo de movimentação da câmera recebeu dois servos motores a fim de possibilitar a rotação em torno do eixo Z - giro para a esquerda e direita -, e em torno

do eixo Y - giro para cima e para baixo. Desta maneira é possível obter um amplo campo de visão. A figura A.9 mostra este mecanismo.

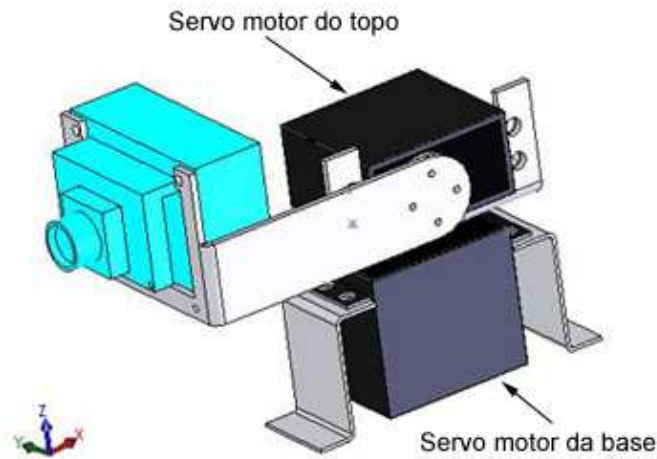


Figura A.9. Mecanismo de Visão.

Após o teste do sistema de locomoção e visão, uma placa de acrílico foi posicionada na parte superior do robô. Sobre a placa de acrílico foram fixados o sistema de visão e o módulo de controle dos servos motores e motores-redutores. A montagem final do robô pode ser observada na figura A.10.

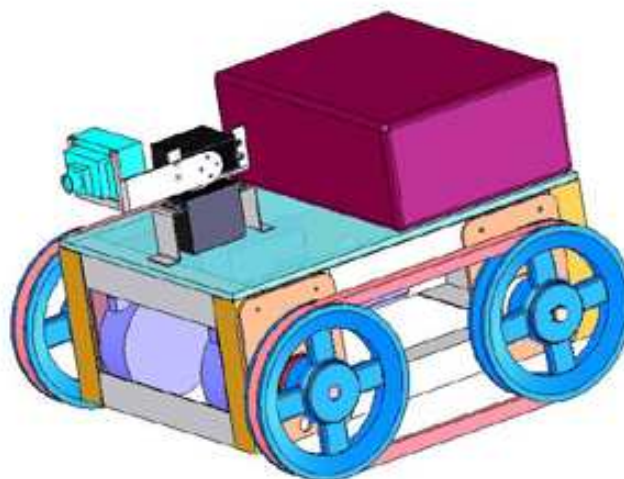


Figura A.10. Montagem final.

A.2 DETALHAMENTO DO MÓDULO ELETRÔNICO

A.2.1 Protocolo de comunicação do módulo eletrônico

A comunicação entre o computador e o módulo eletrônico utiliza um protocolo que tem como base uma palavra de seis *bytes*. Estes bytes são interpretados e processados pelo microcontrolador 8051. A tabela A.2 apresenta a composição dos *bytes* que formam o protocolo de comunicação do módulo eletrônico.

Tabela A.2. Estrutura do protocolo de comunicação do módulo eletrônico.

Servos Motores				Motor DC esquerdo	Motor DC direito
1° byte	2° byte	3° byte	4° byte	5° byte	6° byte

Os quatro primeiros *bytes* da palavra são responsáveis pelo posicionamento dos quatro servos motores que a placa do robô pode manipular, onde cada *byte* é dedicado ao posicionamento de um servo motor. O dado contido em cada *byte* pode ter seu valor variando entre 1 e 60, sendo utilizado para gerar o sinal PWM. A depender da largura do pulso gerado, o servo motor é posicionado em um ângulo diferente. Na figura A.11, é possível observar que o posicionamento do servo motor depende do sinal aplicado a sua entrada. Entre dois pulsos consecutivos, é necessário aguardar aproximadamente 12ms para que o circuito interno do servo motor possa identificar qual posição deve ser atendida.

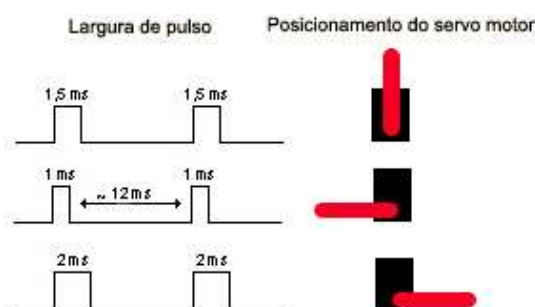


Figura A.11. Posicionamento do servo relativo à largura de pulso.

Estas larguras de pulso e o valor do *byte* respectivo para o posicionamento do servo

estão detalhados na tabela A.3:

Tabela A.3. Posicionamento do servo relativo à largura de pulso.

Largura de pulso	Posicionamento do servo	Valor do byte
1 ms	O servo motor se posiciona totalmente à esquerda no grau 0.	1
$1\text{ms} < x < 1,5\text{ms}$	O servo motor é posicionado entre o grau 1 e o 89, na parte esquerda.	1 a 30
1,5ms	O servo motor se posiciona no centro, no grau 90.	30
$1,5\text{ms} < x < 2\text{ms}$	O servo motor é posicionado entre o grau 91 e o 179, na parte esquerda.	31 a 59
2ms	O servo motor se posiciona totalmente à direita, no grau 180.	60

Os dois *bytes* seguintes aos *bytes* utilizados para o posicionamento dos servos motores são dedicados ao controle dos dois motores DC. Cada *byte* é responsável pelo controle de um dos motores DC. O *byte* dedicado a um motor DC possui a estrutura apresentada na tabela A.4, onde o primeiro *bit* indica a orientação do movimento do motor e os sete *bits* restantes indicam a velocidade do deslocamento do motor.

Tabela A.4. Estrutura do *byte* utilizado para o controle do motor DC.

Motor DC (1 <i>byte</i>)	
1° <i>bit</i> (orientação)	2° ao 8° <i>bit</i> (velocidade)

A orientação do movimento do motor DC pode ser para a frente e para trás. A tabela A.5 apresenta a variação da orientação do movimento relativa ao sinal aplicado aos dois primeiros *bits* do *byte* dedicado a um motor.

Tabela A.5. Estado do motor DC em relação ao sinal aplicado.

Orientação do movimento do motor	Sinal (1° <i>bit</i>)
Para a frente	0
Para trás	1

A velocidade de um motor DC é controlada pela tensão aplicada a sua entrada. O módulo eletrônico controlado pelo microcontrolador gera tensões de 0 a 12 volts, que podem ser definidas, respectivamente, como tensão mínima e tensão máxima. A tabela A.6 apresenta a variação da velocidade do motor relativa ao sinal aplicado nos *bits* 2 a 8 do *byte* dedicado a um motor.

A movimentação do robô é definida pela atuação dos dois motores DC. O robô movimenta-se para a frente quando os motores estão orientados para a frente e com

Tabela A.6. Velocidade do motor DC em relação ao sinal aplicado.

Velocidade do motor	Sinal (2° ao 8° bit)	Tensão aplicada
Parado	0000000 (valor 0)	0 volt
Velocidade máxima	1111111 (valor 127)	12 volts
Velocidade controlada	0000001 a 1111110 (valor 1 a 126)	0 volts < <i>tensão</i> < 12 volts

velocidades iguais. O mesmo ocorre para movimentar para trás, porém a orientação é para trás. Para a direita é necessário que o motor esquerdo tenha maior velocidade que o direito, enquanto que para movimentar para a esquerda o contrário.

Uma outra possibilidade de movimento do robô é o giro sobre o próprio eixo; nesta situação, os motores apresentam a mesma velocidade em direções contrárias de movimento. A tabela A.7 apresenta as possibilidades da direção do movimento do robô com relação aos estados dos motores.

Tabela A.7. Direção do movimento do robô em relação ao acionamento dos motores DC.

Direção de movimento do robô	Motor esquerdo		Motor direito	
	Orientação	Velocidade	Orientação	Velocidade
Parado	qualquer	parado	qualquer	parado
Frente	frente	igual ao motor direito	frente	igual ao motor esquerdo
Trás	trás	igual ao motor direito	trás	igual ao motor esquerdo
Direita	frente	qualquer	qualquer	parado ou menor que o esquerdo
Esquerda	qualquer	parado ou menor que o direito	frente	qualquer
Girar sobre o eixo para a direita	frente	igual ao motor direito	trás	igual ao motor esquerdo
Girar sobre o eixo para a esquerda	trás	igual ao motor direito	frente	igual ao motor esquerdo

A figura A.12 ilustra as duas possibilidades para manobrar o robô. Maiores detalhes sobre a estrutura do robô estão apresentados no apêndice A.

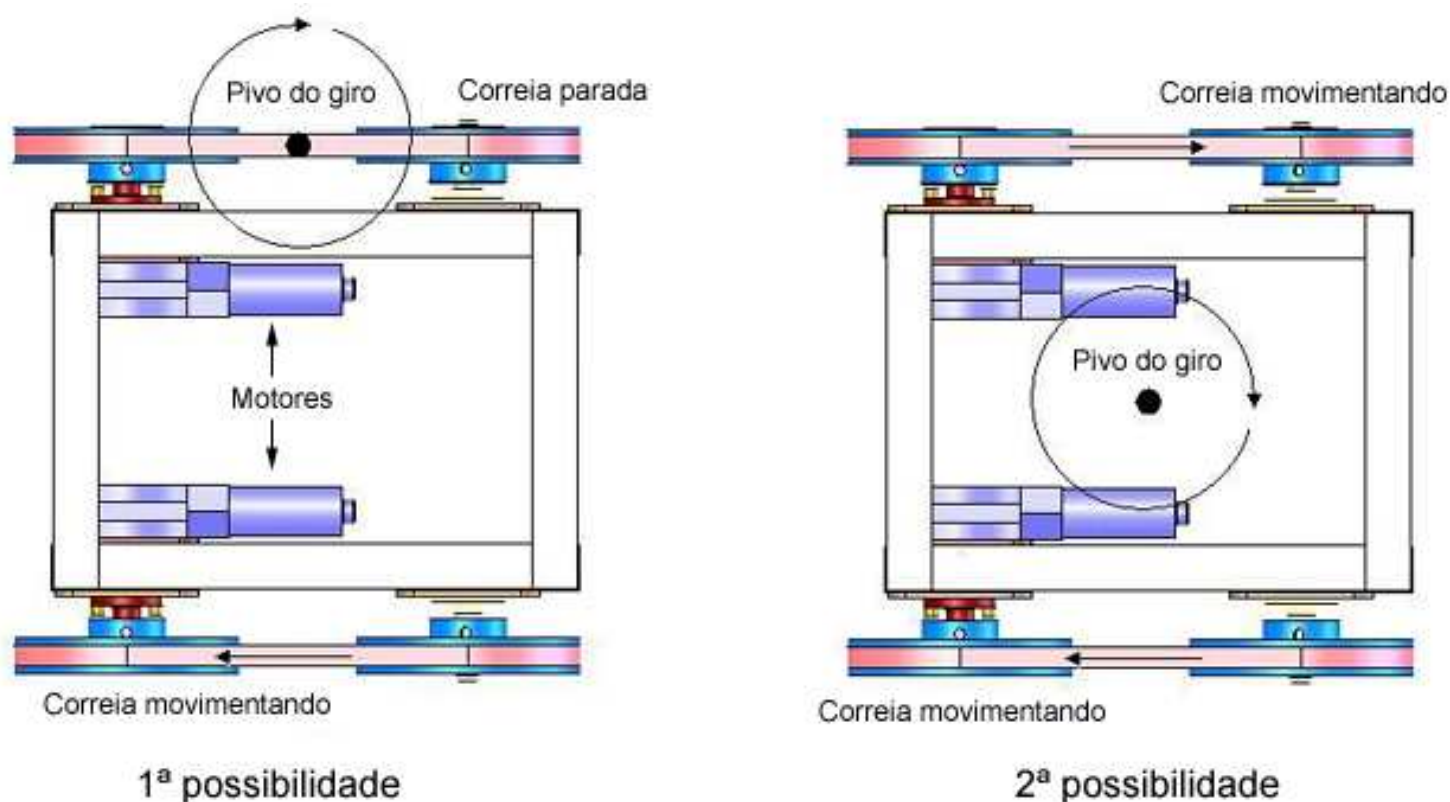


Figura A.12. Possibilidades de giro do Robô.

A.2.2 Protocolo de comunicação do firmware da câmera CMUCam1

O reconhecimento das cores da imagem e a captura de imagens é realizado pelo *firmware* da câmera CMUCam1, onde os comandos para tais ações seguem um protocolo. Para solicitar a captura de uma imagem à câmera é necessário enviar ao firmware da câmera os seguintes caracteres:

- DF \r

Onde, *DF* é o comando *Dump Frame* e o caractere \r indica o fim do envio do comando.

Caso o comando seja identificado corretamente, o firmware da câmera retorna os seguintes caracteres:

- ACK

- 1 2 r g b r g b ... r g b r g b 2 r g b r g b r ... r g b r g b ...

A seqüência de caracteres *ACK* indica que o comando foi reconhecido. O caractere *1* indica que será enviado uma nova imagem. O caractere *2* indica que os dados que compõem os *pixels* de uma coluna da imagem será enviada. Cada seqüência de *RGB* é um *pixel* da coluna que está sendo enviado. A indicação de fim de envio é reconhecida pelo caractere com o valor do *byte*, que o compõe, com o valor 0.

Para solicitar o reconhecimento de uma cor na imagem deve ser enviado ao firmware da câmera a seguinte seqüência de caracteres:

- TC *Rmin Rmax Gmin Gmax Bmin Bmax* \r

Onde, *TC* é o comando de *Track Color*. *Rmin Rmax Gmin Gmax Bmin Bmax* são os valores RGB da cor mínima e máxima que o *firmware* irá utilizar para reconhecer uma cor dentro deste intervalo de cores. O caractere \r indica o fim do envio do comando.

Caso o comando seja identificado corretamente, o firmware da câmera retorna os seguintes caracteres:

- ACK

- M *mx my x1 y1 x2 y2 pixels confidence* \r

Onde, o caractere *M* indica que o pacote de dados que está sendo enviado possui informações sobre o centro da posição na imagem onde a cor solicitada foi encontrada, a região na imagem onde a cor foi encontrada, quantos *pixels* da imagem está no intervalo de cores passado por parâmetro e o valor da confiança da identificação da cor na imagem, sendo respectivamente, *mx my, x1 y1 x2 y2, pixels e confidence*. O caractere \r indica o fim do envio do comando.

As dimensões das imagens capturadas pela câmera são de 160 *pixels* de largura e 143 *pixels* de altura, conforme pode ser visualizado na figura A.13, onde a imagem capturada está localizada na parte esquerda da figura.

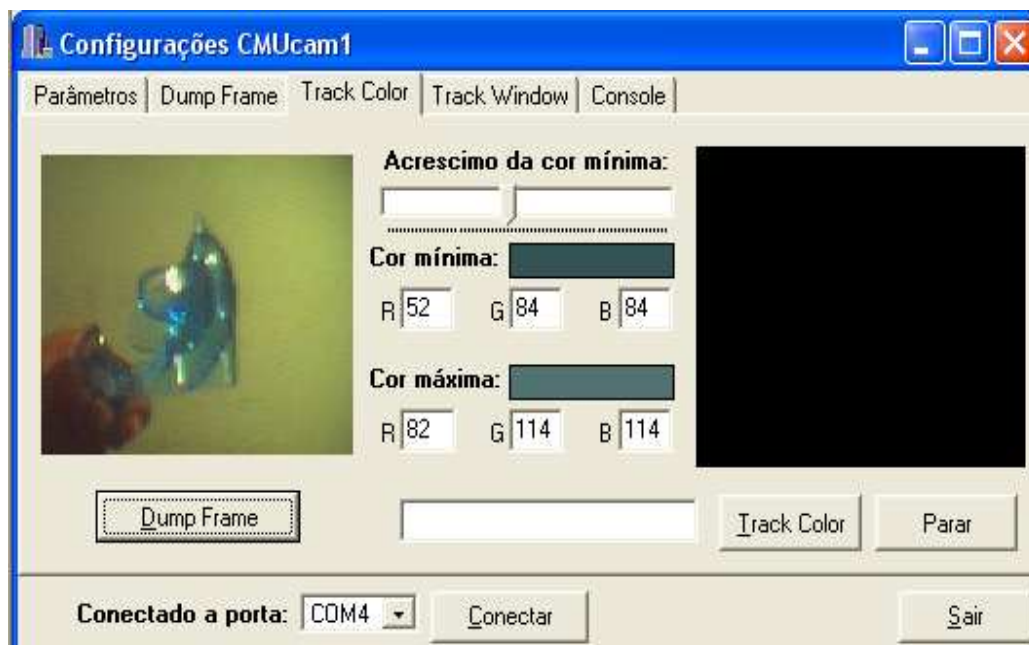


Figura A.13. Imagem capturada da câmera.

Observando a imagem da figura A.13, é possível identificar os componentes RGB das cores mínima e máxima que representam um tom de azul, indicadas nos campos *Cor mínima* e *Cor máxima* na figura. O tom de azul retirado do objeto da imagem será enviado ao *firmware* da câmera CMUCam1 para que seja retornado a localização do tom de azul na imagem. O localização da faixa de cor selecionada na figura A.13, pode ser observada nas imagens da figura A.14.

Na figura A.14, é possível identificar que o objeto da imagem capturada que possui um tom de azul em sua totalidade, foi localizado nas imagens capturadas pela câmera. Pois o retângulo vermelho e o ponto amarelo formados nas imagens (a), (b) e (c) da figura A.14, refere-se a região da imagem que possui o tom de cor passado ao *firmware* da câmera, enquanto que o ponto amarelo refere-se ao ponto central da região da imagem que possui o tom de cor.

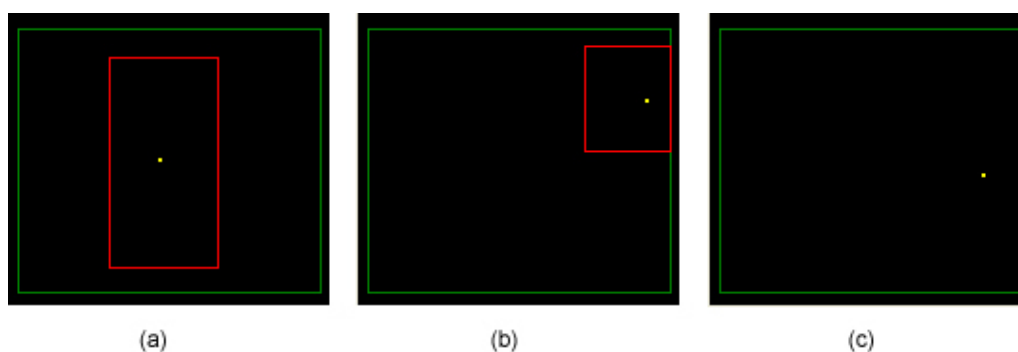


Figura A.14. Dados da localização do tom de cor na imagem.

As imagens da figura A.14, mostram a variação da localização do tom de cor nas imagens capturadas pela câmera. Na imagem (a) da figura A.14, o objeto encontra-se na mesma posição da imagem capturada da figura A.13.

Na imagem (b) da figura A.14, o objeto encontra-se mais a direita e mais para cima da câmera em relação ao instante da imagem (a). É possível afirmar também, que o objeto encontra-se mais afastado, devido a diminuição do retângulo vermelho que refere-se à região onde o tom de cor foi encontrado.

Na imagem (c) na figura A.14 é possível observar que o objeto encontra-se mais a direita em relação a câmera, porém no meio da imagem em relação ao posicionamento para cima e para baixo. Além de ser possível observar que o objeto encontra-se bem afastado da câmera, pois apenas o centro da região que contém a o tom de cor é exibido, não sendo mostrado o retângulo de delimitação da região onde a cor foi encontrada.

A.2.3 Componentes eletrônicos do módulo eletrônico

Todos os componentes eletrônicos utilizados para o controle da movimentação do robô estão inseridos na placa do robô. A figura A.15 exibe o projeto eletrônico da placa principal do robô, na qual é possível observar regiões destacadas e rotuladas.

As regiões em destaque estão detalhadas a seguir:

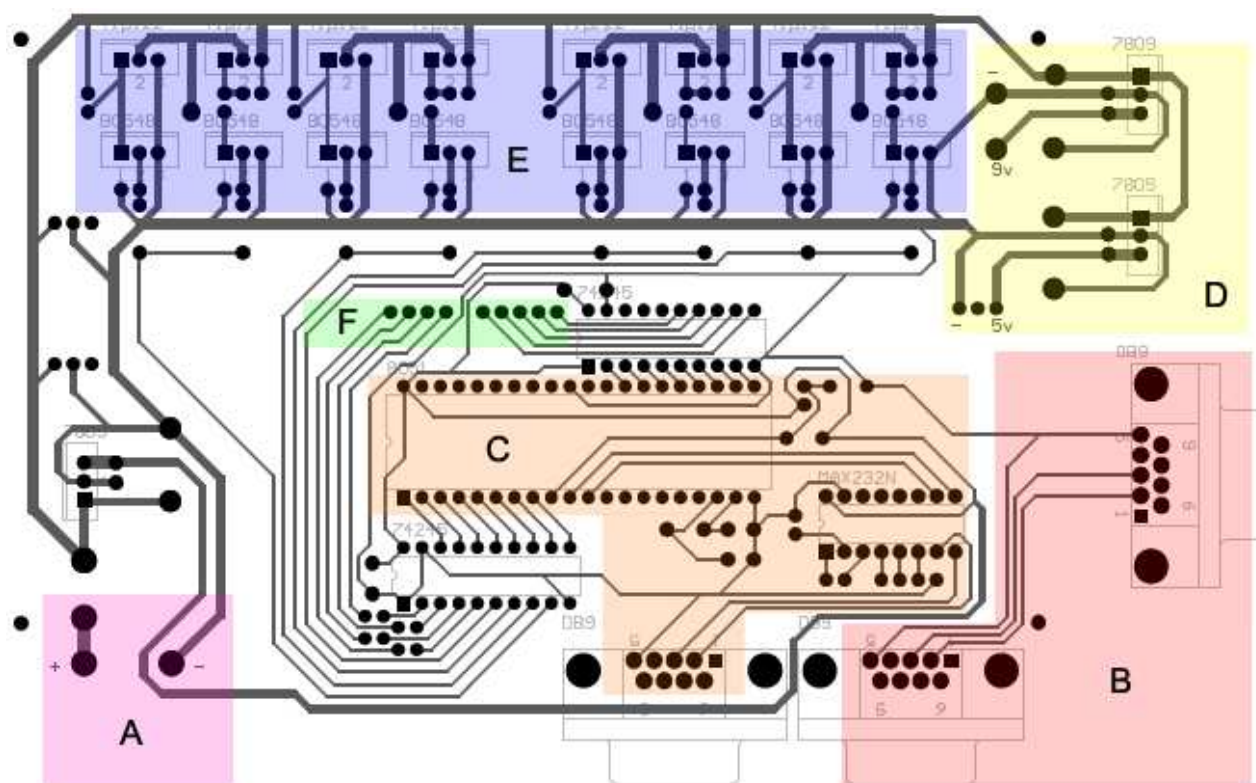


Figura A.15. Placa eletrônica principal do robô com suas regiões mais importantes destacadas.

Região A: Local de conexão da fonte de alimentação do robô, que neste caso apresenta tensão de 12V e corrente de 17A.

Região B: Circuito dedicado à conexão do cabo serial do computador ao cabo serial da câmera de vídeo. O circuito de conexão entre o computador e a câmera de vídeo não manipula os dados que são transmitidos do computador para o *firmware* da câmera e do *firmware* da câmera para o computador, conforme é possível observar na figura A.15. A função deste circuito é apenas manter unidos pela placa todos os componentes do robô que necessitam de energia elétrica.

Região C: Circuito dedicado à recepção dos sinais elétricos e responsável pela execução dos comandos enviados pelo computador. Este circuito é composto por um módulo de conversão dos sinais seriais de padrão RS232 - vindos da porta serial do computador - em sinais TTL utilizados por todos os circuitos integrados da

placa do robô. A conversão do sinal serial para TTL utiliza um CI MAX232N. A saída deste módulo de conversão está conectada à entrada de comunicação do microcontrolador 8051, e a saída de comunicação do microcontrolador também está conectada ao módulo de conversão de sinais.

O microcontrolador recebe os dados vindos do computador, após a conversão, e os processa com o propósito de manipular algum componente eletrônico do robô. Através do microcontrolador, é possível controlar a movimentação do robô, as posições dos servos motores, além de enviar e capturar dados de sensores.

O microcontrolador utilizado na placa do robô pode transmitir dados para o computador. Estes dados, por exemplo, podem ser as informações capturadas pelos sensores ou as posições atuais dos servos motores, além de quaisquer informações úteis para o controle do robô.

Região D: Módulo regulador de tensão para 5V e 9V. A tensão de 5V é utilizada na alimentação dos servos motores e sensores utilizados no robô. A tensão de 9V é utilizada para alimentar a câmera de vídeo.

Região E: Módulo responsável pela amplificação da corrente e geração das tensões de 12V utilizadas na alimentação dos motores DC, usando a configuração de transistores em Ponte H. As tensões geradas na Ponte H são usadas para controlar a direção e a velocidade de movimento do robô. Os sinais vindos do microcontrolador são formados por uma tensão de 5V e baixa corrente, e os mesmos são convertidos para uma tensão de 12V e corrente de 17A ao chegar neste módulo.

Região F: Esta parte do circuito é responsável pela passagem dos sinais de controle dos servos motores e sensores para uma placa auxiliar.

A tabela A.8 lista todos os componentes utilizados na placa do robô relacionados a cada uma das regiões destacadas na figura A.15.

Na figura A.16, é possível observar o projeto da placa auxiliar usada no robô. A placa auxiliar foi desenvolvida para possibilitar uma melhor conexão dos servos

Tabela A.8. Lista de componentes utilizados na placa do robô.

Região	Componentes
A	1 conector para a alimentação. 1 chave
B	1 conector DB-15 macho 1 conector DB-15 fêmea
C	1 microcontrolador 8051 1 Max232N 5 capacitores de $1\mu\text{F}$ 50v 1 cristal 11.0592MHz 2 capacitores de $33\ \rho\text{F}$ 1 resistência de $8\text{k}2\Omega$ 1 botão para <i>reset</i>
D	1 CI 7809 1 CI 7805 2 capacitores de $330\ \mu\text{F}$ 2 capacitores de $1\ \mu\text{F}$
E	4 TIP 122 4 TIP 127 8 BC548 4 resistências $3\text{k}2\Omega$ 4 resistências $470\text{M}\Omega$ 20 resistências $4\text{K}7\Omega$
F	1 CI 74245 4 resistências 56Ω 1 capacitor de $1\ \mu\text{F}$

motores e sensores que são utilizados no robô.

A placa auxiliar do robô é formada por trilhas, para a condução da corrente elétrica, e por conectores. Observando as áreas destacadas na figura A.16, é possível notar três regiões, que estão detalhadas a seguir:

Região A: Local de conexão entre a placa principal e a placa auxiliar do robô. Há um cabo conectado na placa auxiliar para servir como meio de condução, a fim de possibilitar a transmissão dos sinais elétricos após a conexão do cabo à placa principal.

Região B: Quatro conectores disponíveis para a conexão de servos motores que podem ser utilizados no robô. Cada conector possui três pinos. Dois dos três pinos são dedicados à alimentação do motor, através dos quais são transmitidas tensões

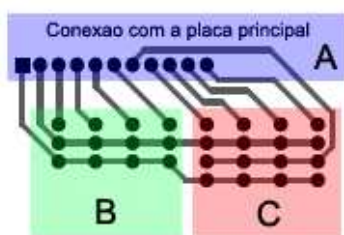


Figura A.16. Placa auxiliar para conexões de servos motores e sensores.

de 5V e 0V. O terceiro pino é responsável pelo sinal de posicionamento do servo motor, que é transmitido pelo microcontrolador da placa principal.

Região C: Quatro conectores disponíveis para a conexão de sensores que podem ser utilizados no robô. Cada conector possui quatro pinos. Dois dos quatro pinos são dedicados à alimentação do sensor, através dos quais são transmitidas tensões de 5V e 0V. O terceiro pino é responsável pela emissão do sinal para ativação do sensor e o quarto pino é responsável pela recepção do sinal originado do sensor, que é transmitido à placa principal para ser corretamente interpretado.