DI-PUCRJ

Série: Monografias do Departamento de Informática
MCC 25/94

Representação do Conhecimento em Banco de Dados Textuais

José Helano Matos Nogueira
Antônio Luz Furtado

Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marques de São Vicente, 225 Gávea
CEP 22453-900 Rio de Janeiro - RJ, Brasil
e-mail: helano@inf.puc-rio.br
e-mail: furtado@inf.puc-rio.br

Julho/94

Resumo

Tentativas de desenvolver uma representação semântica que possa suportar linguagem, raciocínio e recuperação de informações em Banco de Dados Textuais — BDTs torna-se algo não trivial. Este trabalho teve como objetivo um estudo dos principais modelos existentes de representação do conhecimento em Inteligência Artificial e sua possível adequação em Banco de Dados Textual. Apesar da compatibilidade de alguns modelos, vimos que ainda são modelos de pouca expressividade e de difícil manuseio para tratar com os BDTs. Portanto, criamos um novo modelo, TDM (Textual Database Model), voltado especificamente para o uso com textos.

Abstract

Attempts to develop a good semantic network that can support language, reasoning, and information retrieval upon Textual Databases — TDB is not easy. This work has got as goal to make an overview of the main models of knowledge representatation in Artificial Intelligence and at the same time to make the link between this models and Textual Databases. In spite of complexity of the models, we find out that some models had little expression and sometimes they were dificult to handle with TDB. So, we create a new model, TDM (Textual Database Model), where it is specifical to treat texts.

Sumário:

1. Introdução

2. Fundamentos de Representação do Conhecimento
   2.1 Componentes Básicos
   2.2 Nível de Representação
   2.3 Formas de Raciocínio

3. Paradigmas de Representação do Conhecimento
   3.1 Lógica
      3.1.1 Lógica de Predicados
      3.1.2 Lógica Modal de Ação
   3.2 Gramáticas
      3.2.1 Gramática Sintática
      3.2.2 Gramática Semântica
   3.3 Redes Semânticas
   3.4 Frames
   3.5 Sistemas de Produção
   3.6 Dependência Conceitual

4. Textual Database Model - TDM
   4.1 Esquema Gráfico
   4.2 Estrutura Formal
   4.3 Forma de Raciocínio
   4.4 Vantagens e Desvantagens

5. Conclusões

1. Introdução

Este trabalho tem como objetivo um estudo dos principais modelos existentes de representação do conhecimento e sua possível adequação em Banco de Dados Textuais — BDT. Portanto, será feita uma pesquisa sobre as formas de representação, exemplos, vantagens e desvantagens de cada modelo e suas relações com os BDTs. Todavia, as ligações dos modelos de representação do conhecimento existentes com os BDTs não foi encontrado na literatura e nos trabalhos de pesquisa existentes. Com isso, fizemos tais ligações para cada modelo pesquisado. Apesar da compatibilidade de alguns modelos, vimos que ainda são modelos de pouca expressividade e de difícil manuseio para tratar com os BDTs. Portanto, criamos um novo modelo, TDM (Textual Database Model), voltado especificamente para o uso com textos.

Tentativas de desenvolver uma representação semântica que possa suportar linguagem, raciocínio e recuperação de informações em BDTs torna-se algo não trivial . As abordagens em Inteligência Artificial iniciam com definições que tratam os aspectos do mundo real de uma forma independente da implementação e em seguida adiciona estrutura para representar os dados. Já em abadorgens de Banco de Dados, os conceitos são dados por um típico dicionário de dados. Ambos os tipos de abordagens são importantes: a primeira é necessária para dar suporte a linguagem e raciocínio; a segunda é necessária para que o sistema possa encontrar os dados e usá-los. Portanto, um primeiro passo é conceituar o que vem a ser exatamente um banco de dados textual, uma representação do conhecimento e a representação do conhecimento em banco de dados textuais.

Banco de Dados Textual: É o conjunto de dados interrelacionados (palavras, frases e capítulos) em um texto.

Representação do Conhecimento: É a forma de descrever o conhecimento de algo no mundo real em um modelo computacional.

Representação do Conhecimento em BD Textuais: É o modelo do conhecimento formado a partir de texto(s) e o conjunto de relacionamento entre o(s) texto(s) e o modelo representado no computador.

Na seção 2 será mostrado os fundamentos de representação do conhecimento: componentes básicos, níveis de representação e as formas de raciocínio. O estudo dos principais modelos de representação do conhecimento (lógica, gramáticas, redes semânticas, "frames", sistemas de produção e dependência conceitual) junto com as ligações com os BDTs, será feito na seção 3. Também serão apresentadas as principais vantagens e desvantegens de cada modelo no suporte à textos. Na seção 4 será apresentado o TDM (Textual Database Model), com seu esquema gráfico, estrutura formal, forma de raciocínio e principais vantagens e desvantagens.

2. Fundamentos de Representação do Conhecimento

Desde o início dos estudos em Inteligência Artificial, algumas suposições serviram de base para o desenvolvimento das pesquisas, e em representação do conhecimento estas suposições foram explicitadas por Smith, [Smith82], onde ele introduz a chamada *Hipótese de Representação do Conhecimento*:

> **"Qualquer processo inteligente realizado por uma máquina, será formado por ingredientes estruturais que: (a) serão naturalmente percebidos pelos observadores externos como uma descrição proposicional de conhecimento exibido pelo processo, e (b) independentemente de tal atribuição semântica externa, terão papel formal, causal e essencial na geração do comportamento que gera tal conhecimento".**

Esta hipótese explica o que representa um sistema inteligente, ou ao menos que tipo de sistema inteligente a pesquisa em Inteligência Artificial procura desenvolver atualmente. Para evitar as dificuldades filosóficas intrínsecas à definição do termo representação do conhecimento, uma definição operacional foi proposta:

> **"Uma representação do conhecimento é uma combinação de estrutura de dados e de procedimentos de interpretação que, se usados de maneira correta dentro de um programa, levarão a um comportamento que simule o comportamento dos seres humanos" [BaFe81].**

Duas linhas de pesquisa principais foram seguidas para a construção de tais sistemas inteligentes: o *Modelo Conexionista* e o *Modelo Simbolista*. A linha conexionista visa a modelagem da inteligência humana através da simulação dos componentes do cérebro, i.e. de neurônios e suas interligações (redes neurais ou neuronais). O modelo simbolista segue a tradição da lógica, com sua manipulação simbólica dos fatos, predicados, regras de inferências e métodos de provas formais. Neste trabalho será adotado o modelo simbolista de representação do conhecimento.

2.1 Componentes Básicos

A definição do termo representação do conhecimento apresenta as mesmas dificuldades filosóficas encontradas para se definir o termo Inteligência Artificial. Contudo, os componentes essenciais de uma representação são:

- O mundo a ser representado
- O mundo da representação (modelo computacional)
- As relações entre estes dois mundos

No nosso caso em particular, os componentes são formados pelos os itens (I, II e III), veja as figuras 1,2 e abaixo:

(I) Texto

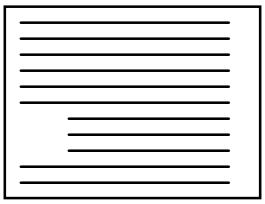

Figura 1 - Mundo a ser representado - Texto

(II) Mundo da Representação

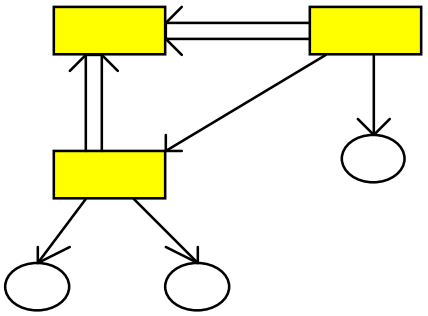

Figura 2 - Modelo Computacional

(III) As relações entre o mundo real e o mundo da representação

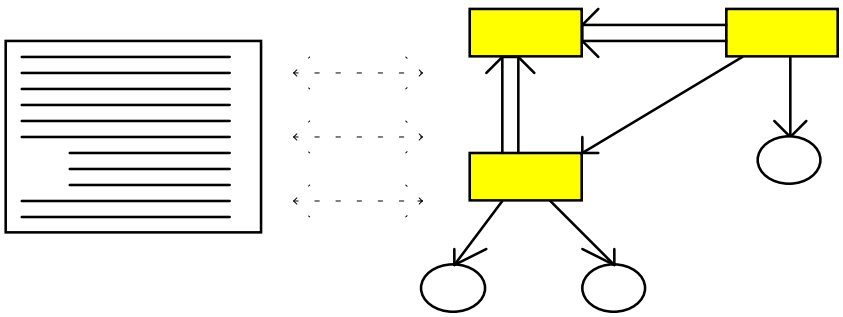

Figura 3 - Relacionamentos entre os dois mundos

Outro fator importante é que tanto o mundo a ser representado quanto o mundo da representação devem contar com operadores que permitam a manipulação dos elementos destes mundos.

A relação entre os dois mundos define a *semântica da representação*. Esta semântica pode ser expressa de diversas maneiras:

- **Informal**: Os significados das expressões utilizadas no mundo da representação são atribuídos intuitivamente a partir dos símbolos utilizados. Este tipo de semântica é

extremamente impreciso e é característico dos primeiros esforços em modelar o conhecimento.

- ▷ **Procedimental**: O significado das expressões do mundo da representação é definido pelo comportamento dos programas que as manipulam. A semântica dos programas é complexa e limita a utilidade desta abordagem.

- ▷ **Descritiva**: O significado das expressões do mundo da representação é estabelecido através de descrições precisas, em lingugem natural, de sua correspondência com o mundo real e a representação.

- ▷ **Equivalência**: Os significados das expressões do mundo da representação são associados a expressões de uma outra linguagem com semântica estabelecida (e.g. lógica).

A tendência atual é adotar uma semântica de equivalência. Neste caso, abstrai-se o mundo a ser representado considerando-o como um conjunto de objetos arbitrários, cuja única propriedade é a identidade. Os operadores que atuam sobre este mundo abstrato são os operadores usuais da teoria dos conjuntos e da lógica. O mundo da representação é definido sob a forma de uma linguagem formal, com operadores definidos sintaticamente. Finalmente, a relação entre os dois mundos é dada por uma teoria semântica que associa um significado à linguagem de representação.

2.2 Nível de Representação

Independente do formalismo de representação do conhecimento que escolhemos é necessário responder a seguinte pergunta:

Em que nível de detalhes o mundo deve ser representado ?

Outra maneira freqüente de se colocar este fato é:

Quais devem ser os nossos termos primitivos ?

Ou seja, deve haver um pequeno número termos de primitivos de baixo nível ou um número maior destes termos abrangendo uma faixa maior ? Como exemplo, suponha que estamos interessados no seguinte fato:

```
O Brasil baixou a inflação !
```

Poderíamos representar isto como

(i) baixou(agente(Brasil), objeto(inflação))

Com esta representação podemos responder perguntas do tipo :

```
Quem baixou a inflação ?
```

Resposta : Brasil

Mas, não podemos responder :

Quem reduziu a inflação ?

A resposta óbvia para esta última pergunta é "Brasil", mas dado o único fato que temos, não podemos descobrir esta resposta. Para responder esta pergunta basta acrescentar uma das soluções abaixo:

(ii) $\forall x,y$ baixou(x,y) $\Rightarrow$ reduziu(x,y)

(iii) reduziu(agente(Brasil), objeto(inflação))

Comentários:

- A principal vantagem de converter declarações ou fatos do mundo real em um pequeno conjunto de termos primitivos é a obtenção de uma forma canônica.
- O argumento mais forte contra a utilização dos termos primitivos de baixo nível é o trabalho de conversão de cada fato de alto nível em baixo nível.
- Um segundo problema com o uso de termos primitivos de baixo nível é o armazenamento. Um fato em alto nível pode gerar muitos termos primitivos.
- Um terceiro problema com o uso de termos primitivos é que , em muitos domínios — principalmente textos, não está devidamente claro quais devem ser os termos primitivos.

Por exemplo:

Como dizer que : João é primo de Maria.

Dado como termos primitivos : mãe, pai, filho e filha

João = filho(irmão(mãe(Maria)))
João = filho(irmã(mãe(Maria)))
João = filho(irmão(pai(Maria)))
João = filho(irmã(pai(Maria)))

- Por último, quanto mais baixo nível os termos primitivos, menos inferência para raciocinar e quanto menos inferência, mais espaço de armazenamento.

Moral da história *: Deve-se ter bastante cuidado ao escolher o nível para representar o conhecimento !!!*

2.3 Formas de Raciocínio

O processo de raciocínio deve ser capaz de gerar novos elementos de conhecimento a partir de conhecimentos armazenados na base de conhecimento. Cinco tipos principais de processos de raciocínio podem ser evidenciados: formal, procedimental, analógico, raciocínio por abstração e generalização e raciocínio meta-nível.

➲ *Raciocínio Formal*

O raciocínio formal consiste na tranformação sintática de elementos do conhecimento através da aplicação de regras de inferência. Três tipos de raciocínio formal são possíveis:

- Inferência Dedutiva
- Inferência Abdutiva
- Inferência Indutiva

Destas três formas de inferência, apenas a primeira é formalmente correta de acordo com a semântica da lógica clássica. As outras dependem do domínio de trabalho e podem produzir resultados errados.

➲ *Raciocínio Procedimental*

O raciocínio procedimental é utlizado sempre que existe um algoritmo para determinar a solução de um problema relevante ao raciocínio corrente. Propriedades geométricas e matemáticas dos elementos do conhecimento são especialmente adequadas para serem inferidas através de raciocínio procedimental. Inferências associadas à representações analógicas de objetos físicos são freqüentemente procedimentais por natureza. As relações espaciais entre dois corpos, por exemplo, podem ser deduzidas através de raciocínio formal a partir de um conjunto de predicados descrevendo suas posições no espaço, no entanto estas mesmas relações podem ser determinadas muito mais eficientemente, caso se disponha de uma representação analógica do domínio.

➲ *Raciocínio Analógico*

O raciocínio analógico consiste em determinar relações entre dois domínios de maneira que os métodos de soluções adequados para problemas de um domínio possam ser adaptados para solucionar problemas de um outro domínio. Embora seja usual para os seres humanos, este método apresenta uma série de dificuldades para ser formalizado. O mecanismo utilizado pela mente humana para gerar analogias não é bem compreendido e sua aplicação para auxiliar o raciocínio em sistemas de Inteligência Artificial é bastante limitada [Wins84]. Um domínio onde o raciocínio analógico foi aplicado com relativo sucesso foi a demonstração automática de teoremas [Nogu92b].

➲ *Raciocínio por Generalização e Especificação*

Este tipo de raciocínio consiste na geração de conceitos a partir de exemplos, contra-exemplos e abstrações . Este método é utilizado principalmente em programas de aprendizado. Tipicamente, apresenta-se ao programa uma série de exemplos e contra-exemplos, então o programa deve gerar um conceito mais específico ou mais genérico em uma determinada linguagem formal [Wins84, Nogu91a, Nogu92c]. Apesar deste raciocínio também ser típico nos seres humanos, sua utilização em Inteligência Artificial apresenta uma complexidade considerável.

➲ *Raciocínio Meta-Nível*

O raciocínio meta-nível consiste na manipulação de informações sobre a utilidade e disponibilidade do conhecimento sobre o domínio de trabalho. Este é um importante tipo de raiocínio, pois ele permite um sistema inteligente ser capaz de determinar suas próprias limitações, e saber reconhecer problemas que estão além de suas capacidades. Outro uso para o raciocínio meta-nível é prover um sistema inteligente com a capacidade de explicar seu próprio raciocínio: para ser capaz de explicar seu raciocínio um sistema deve ser capaz de representar e utilizar o conhecimento sobre seu mecanismo dedutivo, esta capacidade é chamada ***introspecção*** [Maes86, Mins86].

3. Paradigmas de Representação do Conhecimento

3.1 Lógica

Diferentemente das demais técnicas de representação de conhecimento, desenvolvidas nas últimas décadas, a lógica tem uma longa história, que remonta aos antigos filósofos gregos. A lógica trata fundamentalmente de dois conceitos: *verdade* e *prova*. A lógica, enquanto método de representação de conhecimento, consiste na utilização do formalismo lógico como base para a construção de programas e modelos.

A fim de resolver os problemas complexos de representação são necessários uma grande quantidade de conhecimento e certos mecanismos para manipulá-lo, com o intuito de criar soluções para novos problemas. À medida que observamos, em maiores detalhes, os modos de representar o conhecimento, torna-se claro que os modelos específicos de representação de conhecimento permitem mecanismos de inferência mais direcionados e poderosos a um determinado tipo de aplicação. Diversos tipos de lógica foram propostos para capturar o conhecimento, serão vistos aqui uma lógica clássica (lógica de predicados) e uma não clássica (lógica modal de ação). Estas lógicas foram escolhidas, dentre o universo das lógicas existentes, devido suas possíveis adequações ao problema representação do conhecimento em Banco de Dados Textuais [Nogu92b, Nogu92c, Rich93, RoBe92].

3.1.1 Lógica de Predicados

Foi escolhido o uso da lógica de predicados (mais especificamente lógica de primeira-ordem) devido ao seu alto poder de expressividade, o que não ocorre com a lógica proposicional, a lógica de primeira-ordem permite representações de sentenças que não podem ser representadas em lógica proposicional. Na lógica de predicados podemos representar fatos do mundo real como declarações escritas em uma forma de Fórmulas Bem Formadas (FBFs). Uma motivação importante para o uso da lógica na representação do conhecimento deve-se ao fato de que ao empregarmos frases lógicas podemos realizar deduções e raciocinar sobre o conhecimento. Determinar a validade de uma proposição pode ser algo não muito trivial. De fato, existem procedimentos que encontrarão a prova de um teorema proposto se em verdade ouver um teorema: mas eles não dão garantia de parar se a frase proposta não for teorema — *problema da parada*, ou seja, a lógica de primeira-ordem é semi-decidível. Um procedimento simples é usar regras de inferências para gerar teoremas a partir dos axiomas, de uma forma ordenada, testando cada axioma para ver se é aquele para o qual se busca uma prova. Apesar dos resultados negativos, como o de não existir nenhum procedimento de decisão, ela serve como um meio bastante útil de representar e manipular diversas formas de conhecimento.

Para explorar a utilização da lógica de predicados como um meio de representar o conhecimento em Banco de Dados Textuais, considere as seguintes frases retiradas de um trecho de um texto :

- César era soberano
- Marcos era uma homem
- Marcos era pompeiano
- Todos os pompeianos eram romanos
- Todos os romanos eram leais a César ou o odiavam
- Cada habitante era leal a alguém
- As pessoas tentam assassinar soberanos a quem não sejam leais
- Marcos tentou assassinar César

Os fatos descritos por essas frases podem ser representados como um conjunto de FBFs na lógica de primeira-ordem, como segue:

- soberano(César)

> Ignoramos aqui o fato de que nomes próprios freqüentemente não correspondem a referências de individuos em particular, pois muitas pessoas partilham do mesmo

nome. Muitas vezes decidir qual entre as diversas pessoas do mesmo nome está sendo citada em uma frase em particular poderá exigir uma quantidade razoável de raciocínio e conhecimento.

- homem(Marcos)

  Esta representação captura o fato Marcos ser um homem. Ela deixa de capturar algo da informação na frase em português, mas precisamente a noção de tempo passado. Se esta omissão é ou não aceitável dependerá da utilização que pretendemos dar ao conhecimento. Para este exemplo simples esta noção temporal não é relevante. Caso desejemos capturar as noções temporais, basta utilizarmos a lógica temporal, que é uma extensão da lógica clássica [Nogu92c].

- pompeiano(Marcos)

- $\forall x$ pompeiano(x) $\rightarrow$ romano(x)

- $\forall x$ romano(x) $\rightarrow$ (leal(x,César) $\vee$ odiar(x,César)

  Em português, a palavra *ou* às vezes significa o *ou* lógico inclusivo, e em determinado momento significa o *ou* exclusivo. Aqui utilizamos a interpretação do ou inclusivo. Contudo, caso seja desejável o uso do *ou* exclusivo, isto poderia ser descritoda seguinte maneira:$\forall x$ romano(x) $\rightarrow$ (leal(x,César) $\vee$ odiar(x,César)) $\wedge$ $\neg$(leal(x,César) $\wedge$ odiar(x,César))

- $\forall x \exists y$ leal(x,y)

  Um problema importante que surge ao se tentar converter frases do português para declarações lógicas é o escopo dos quantificadores. Esta frase diz, como supomos ao escrever a fórmula lógica acima, que para cada pessoa que exista, existirá alguém que é leal - possivelmente um alguém diferente para cada um ? Ou ela diz que existe alguém a quem todos são leais (que seria escrito $\exists y \forall x$ leal(x,y)) ? Muitas vezes apenas uma das duas interpretações parece provável [Rich93, Nogu92c].

- $\forall x \forall y$ pessoa(x) $\wedge$ soberano(y) $\wedge$ tentar_assassinar(x,y) $\rightarrow$ $\neg$leal(x,y)

  Esta frase também é ambígua. Ela significa que os únicos soberanos que as pessoas tentam assassinar são aqueles a quem elas não sejam leais (a interpretação utilizada aqui), ou ela significa que a única coisa que as pessoas tentam fazer é assassinar soberanos aos quais não sejam leais ?

- tentar_assassinar(Marcos,César)

Vantagens :

i. É um método natural de representação.

ii. Modular e flexível, podendo ampliar e modificar a representação facilmente.

iii. Forma bastante poderosa para derivar novos conhecimentos a partir de conhecimentos anteriores.

iv. Pode resolver problemas de domínios variados.

v. Possui um semântica formal bem definida (Semântica de Tarski)

Desvantagens :

i. Muitas frases em uma determinada linguagem natural são ambíguas, portanto a escolha da interpretação correta poderá ser difícil

ii. Quando o conjunto de declarações é muito grande, por exemplo um texto, torna-se difícil representar todas as relações

iii Não representa o sequenciamento das ações em um texto

iv Não trata questões temporais

3.1.2. Lógica Modal de Ação

A Lógica Modal de Ação foi escolhida como exemplo pois com ela é possível capturar a noção de sequenciamento das ações de um texto baseadas nas informações contidas em cada cenário. Além disso, a noção dos operadores, principalmente a implicação, será na verdade ações efetuadas pelo agente a partir de um estado inicial S_o para um estado seguinte S_i, conforme apresentado na figura 4. Maiores detalhes podem ser vistos em [Gard88, RoBe92].

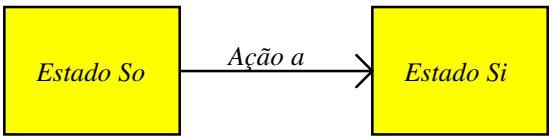

Figura 4 - Mundança de estado ao executar a ação *a*

Uma Lógica Modal de Ação é um aparato lógico que tem por objetivo representar propriedades dos constituintes de um sistema e como tais propriedades são alteradas pela execução de uma ação. Desta forma, uma *ação* é uma entidade do sistema que muda o conjunto de propriedades nele observadas, num determinado instante de tempo.

Numa Lógica Modal de Ação os operadores modais de necessidade □ e possibilidade ◊ são substituídos por uma família de operadores [α] e < α >, respectivamente, onde α é uma ação.

Freqüentemente existe um subconjunto de propriedades que se espera valerem em todos os instantes. Tais propriedades serão referidas aqui por *especificação* e intuitivamente descrevem o mecanismo de funcionamento do sistema.

Um *cenário* é visto como um estado global do sistema. Cada cenário constitui-se numa Teoria da Lógica de Primeira-Ordem representando as características de um estado do sistema, sendo esta teoria uma extensão da teoria representada na especificação. Os cenários se relacionam através de ações. A execução de uma ação faz o sistema mudar de um cenário para outro.

As propriedades dos estados são descritas através de predicados. Os predicados não contém um argumento indicador do estado onde os mesmos se verificam. Assim, este estado é representado pelo operador modal posicionado antes da fórmula. Por exemplo,

Suponha que se *mesa(a)* representa o fato que um bloco *a* está sobre a mesa no estado inicial, [*desemplilha(a,b)*]*mesa(a)* representa o fato de que o bloco *a* está na mesa no estado resultante da execução da ação desempilhar o bloco *a* de cima do bloco *b* no estado inicial.

Um *Modelo* para uma especificação é uma coleção de cenários e uma coleção de funções de transformação de cenários associados às ações.

Vantagens :

i. É um método natural de representação.

ii. Modular e flexível, podendo ampliar e modificar a representação facilmente.

iii. Forma bastante poderosa para derivar novos conhecimentos a partir de conhecimentos anteriores.

iv. Pode resolver problemas de domínios variados.

v. Mostra o sequenciamneto das ações em um texto

vi Pode tratar a questão temporal em um texto

Desvantagens :

i. Muitas frases em uma determinada linguagem natural são ambíguas, portanto a escolha da interpretação correta poderá ser difícil

ii. Quando o conjunto de declarações é muito grande, por exemplo um texto, torna-se difícil representar todas as relações

iii. Ainda é muito recente, as pesquisas nesta lógica

iv. A Lógica Modal de Ação pode gerar uma complexidade exponencial ao representar todas as relações entre as ações e cenários

3.2 Gramáticas

3.2.1 Gramática Sintática

A análise sintática ("*parsing*") de uma frase ou texto é um processo que torna-se necessário utilizar uma *gramática* que descreva a estrutura de cadeias de uma linguagem em particular. Dada tal gramática, o analisador pode designar uma estrutura para cada sentença gramatical. Esta estrutura é denominada *árvore sintática.* A figura 5 apresenta um exemplo de uma gramática simples para uma subconjunto do português.

Frase: **`Peer Gynt seduziu a rainha.`**

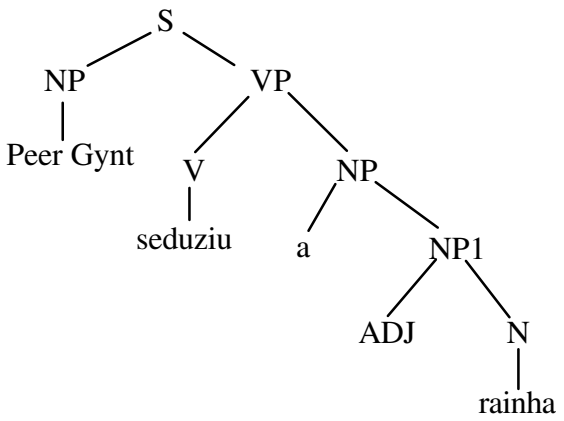

Figura 5 - Árvore Sintática para uma frase

Observe que a estrutura da árvore sintática corresponde ao modo como as regras da gramática poderiam ser aplicadas para transformar o símbolo inicial S na frase final.

O processo de análise sintática faz basicamente duas coisas:

- Determina que frases são aceitas como sintaticamente bem formadas.
- Fornece uma estrutura para as frases sintaticamente bem formadas.

Vantagens:

i. Define a estrutura das frases

ii. Mostras as frases sintaticamente bem formadas

Desvantagem :

i. Não captura a característica semântica de um frase ou texto

3.2.2 Gramática Semântica

A gramática semântica é uma gramática isenta de contexto, onde a escolha de não-terminais e regras de produção é governada por funções semânticas, além das funções sintáticas. Um exemplo de um fragmento de gramática semântica é apresentado abaixo. Esta gramática é uma versão simplificada de uma interface de uma base de dados que contém informações sobre navios e suas características.

S → qual é PROPRIEDADE_NAVIO de NAVIO ?

PROPRIEDADE_NAVIO → a PROP_NAVIO | PROP_NAVIO

PROP_NAVIO → velocidade | comprimento | contingente | tipo | largura

NAVIO → NOME_NAVIO | o mais rápido NAVIO2 | o maior NAVIO2 | NAVIO2

NOME_NAVIO → amazonas | niva | ceará

NAVIO2 → PAÍSES | NAVIO3

NAVIO3 → TIPO_NAVIO

TIPO_NAVIO → porta_aviões | corveta | cruzador | petroleiro

PAÍSES → brasil | nanibia | india | tailândia | filipinas | França | inglaterra

Vantagens :

i. Em vez de conter categorias sintáticas não-terminais, como NP e VP, utiliza-se de categorias semânticas, como NAVIO e PROPRIEDADE_NAVIO

ii. Consegue evitar alguns tipos de ambigüidades

iii. É capaz de analisar as frases sintaticamente bem formadas

Desvantagens :

i. O número de regras exigidas para representar um texto pode ser muito grande

ii. Generalizações sintáticas não são consideradas

iii. Utilizada apenas para um domínio restrito

iv. A complexidade pode se tornar exponencial dependendo do número de relacionamentos existentes no texto

3.3 Redes Semânticas

As redes semânticas foram originalmente projetadas como meio de representar o significado das palavras inglesas. Numa rede semântica, a informação é representada por um conjunto de nós ligados um ao outro por um conjunto de arcos rotulados. Assim, os nós podem representar os objetos e os elos representar as relações entre estes objetos. Entretanto, os nós podem também ser utilizados para representar classes, entidades, palavras de uma linguagem, entre outras possíveis interpretações, dependendo do sistema de rede semântica em questão. As redes semânticas são representadas graficamente por estes arcos e nós, mas não podem ser representadas em um programa deste modo. Os elos *é_um* e *parte_de* são bastante comuns em sistemas de redes semânticas. Este tipo de elo é utilizado para determinar a herança de propriedades. Os demais elos são específicos do domínio e representam propriedades de conceitos. Normalmente as redes semânticas são representadas utilizando-se algum tipo de linguagem declarativa, como PROLOG e LISP. Formalmente ela pode ser representada por uma estrutura de predicados codificada em lógica de primeira-ordem. Uma rede semântica típica para interpretar uma frase é apresentada na figura 6.

Frase: **`Merlim deu o livro Sociedade da Mente para Artur`**

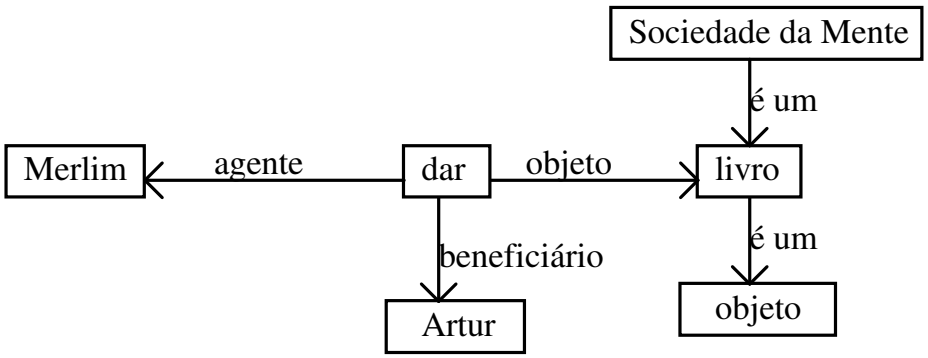

Figura 6 - Rede semântica para uma frase

Vantagens :

i. Modelo de representação genérico (trata diversos domínios)

ii. Bom modelo para dar significado de palavras

iii. Trata herança de propriedades

iv. Boa notação gráfica, porém não pode ser representada em um programa deste modo

Desvantagens :

i. Não trata quantificadores : $\forall$, $\exists$

ii. Grande número de relacionamentos quando houver um texto

iii. Não representa a estrutura interna dos objetos

3.4 Frames

É uma variação das redes semânticas que permite a representação de estruturas internas dos objetos. Tipicamente, o "frame" descreve uma classe de objetos, tais como cadeira ou sala. Tais objetos consistem de uma coleção de escaninhos ("slots", facetas) que descrevem aspectos dos objetos. Eles são preenchidos por valores específicos ou até mesmo por outros quadros. Associado a cada escaninho poderá haver um conjunto de condições que deverá ser satisfeito por qualquer escaninho a ele destinado. Cada escaninho também poderá ser preenchido com uma valor por omissão ("default"), para que na ausência de informação específica em sentido contrário, os escaninhos possam ser instanciados. Informações procedimentais também podem estar associadas a um escaninho em particular. Por exemplo, muitas vezes é útil descrever o que deveria ser feito sempre que um escaninho fosse preenchido (chamado *procedimento se adicionado*) ou o modo como o valor para um escaninho pode ser calculado se necessário (chamado de *procedimento a estabelecer*). A utilização de tais procedimentos encaixados dentro de uma estrutura declarativa é denominada *procedimento de ligação*. Graficamente não há diferenças entre ela e as redes semânticas.

Vantagens :

i. Modelo de representação genérico (trata diversos domínios)

ii. Bom modelo para dar significado de palavras

iii. Trata herança de propriedades

iv. Boa notação gráfica, porém não pode ser representada em um programa deste modo

Desvantagens :

i. Grande número de relacionamentos quando houver um texto

ii. Em um sistema grande a complexidade fica exponencial

3.5 Sistemas de Produção

Sistema de Produção é um nome genérico dado a todos os sistemas baseados em *regras de produção*, isto é, pares de expressões consistindo de uma condição e uma

ação. A idéia inicial foi proposta poe Post, através dos sistemas de Post, que consiste numa tranformação sobre cadeias de caracteres.

Em geral, a arquitetura dos sistemas de produção consiste de três módulos:

- uma base de regras,
- uma memória de trabalho,
- um motor de inferência.

O motor de inferência controla a atividade do sistema. Esta atividade ocorre em ciclos, cada ciclo consistindo em três fases:

i) Correspondência dos dados: onde as regras que satisfazem a descrição da situação atual são selecionadas

ii) Resolução de conflitos: onde as regras que serão realmente executadas são escolhidas entre as regras que foram selecionadas na primeira fase.

iii) Ação: a execução propriamente dita das regras.

O teste para verificar se uma nova regra satisfaz uma dada situação depende de como o conhecimento é representado internamente na base de regras e na memória de trabalho. Este teste pode ser um processo de unificação entre duas fórmulas lógicas, a comparação entre dois quadros, ou simplismente um teste de igualdade sintática entre dois símbolos. O teste depende também da estratégia de controle adotada pelo motor de inferência:

Se o sistema trabalha em encadeamento para frente
Então sua condição é posta em correspondência com a descrição da situação na memória de trabalho
Senão (encadeamento para trás) a ação é comparada com os objetivos do sistema.

O uso de sistemas de produção é bastante utilizado para tratar sistemas especialistas. Em tais sistemas, a semântica da linguagem fica perdida devido a sua característica sintática de representação do conhecimento. O uso de sistemas de produção em BD textuais fica inadequado, pois procuramos um modelo de representação com boa caracterização semântica.

Vantagens :

i. Modularidade do conhecimento

ii. Tratamento de unificação (podendo ser modificado para tratar dados textuais).

iii. Uso de regras de inferências

iv. Sintaxe bem definida, podendo representar a estrutura sintática dos textos, assim como nas gramáticas sintáticas.

Desvantagens :

i. Ineficiência para caracterização semântica da linguagem utilizada

ii. Complexidade do fluxo de controle entre os dados textuais e o motor de inferência.

3.6 Dependência Conceitual

A análise de uma frase dentro de uma representação de dependência conceitual é semelhante ao processo de análise sintática de algumas tipos de gramáticas semânticas. Em dependência conceitual (DC) o processo de análise de uma frase é fortemente acionado por um conjunto de expectativas que são utilizadas com base no verbo principal da frase. O primeiro passo para mapear uma frase dentro de uma representação em Dependência Conceitual envolve um processamento sintático, que por sua vez envolve a extração do substantivo principal e do verbo (com sua categoria sintática). Logo em seguida, a parte conceitual assume e faz uso de um dicionário do tipo *verbo-ação*, que contém uma produção para cada local onde o verbo pode aparecer.

Este modelo foi construído para representar o significado de frases em linguagem natural. Com este intuito, foi criado uma teoria de como representar o significado das frases em linguagem natural de modo a:

- Facilitar a geração de inferências das frases
- Ser independente da linguagem em que a frase foi originalmente declarada

Por causa dessas duas observações acima, a representação usando Dependência Conceitual para uma frase é construída não de termos primitivos correspondentes a palavra utilizada na frase , mas de termos primitivos conceituais que podem ser combinados para formar os significados de palavras em qualquer linguagem em particular. Diferentemente das redes semânticas, que apenas fornecem uma estrutura onde os nós representam a informação em qualquer nível, a Dependência Conceitual fornce tanto uma estrutura como um conjunto específico de termos primitvos, a partir dos quais as representações de informações particulares podem ser constrídas.

Um exemplo simples de como o conhecimento é representado em Dependência Conceitual é apresentado pela frase

Frase: **Gynt pegou o livro de Machado de Assis.**

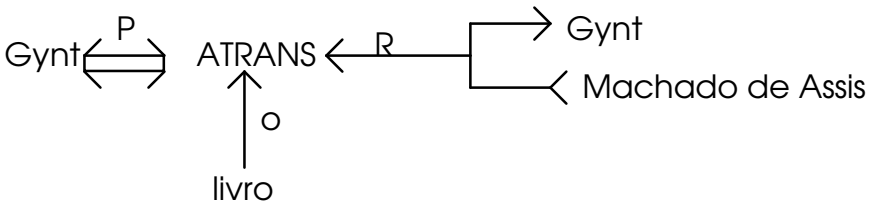

Figura 7 - Representação de uma frase usando DC

onde os símbolos utilizados no modelo acima possuem os seguintes significados

- Setas indicam a direção da dependência
- Seta dupla indica um elo de mão dupla entre o agente e a ação
- **P** indica o tempo no passado
- **ATRANS** é um dos termos primitivos utilizados pela teoria. Ele indica tranferência de posse.
- **o** indica a relação de objeto
- **R** indica a relação de recipiente

A dependência conceitual fornece não apenas uma estrutura onde o conhecimento pode ser representado, mas também um conjunto específico de blocos de construção dos quais as representações podem ser construídas. Estes conjuntos básicos assim como toda a teoria e alguns exemplos de respresentação do conhecimento usando Dependência Conceitual podem ser encontrado em [Rich93, Wins84].

Vantagens :

i. Os termos primitivos não são construídos de palavras tiradas das frases, mas de conceitos que formam o significado

ii. Independente da linguagem natural utilizada

iii. Muitas inferências já estão contidas na própria representação

Desvantagens :

i. Requer que todo o conhecimento seja decomposto em seus primitivos

ii. Ao tratar com muitas frases ou textos as relações tornam-se muito complexas e de compreensão global extremamente difícil

iii Criação de conceitualização própria e limitada aos termos primitivos da teoria

4. Textual Database Model — TDM

Conforme foi visto na seção 3, diversos modelos foram propostos para representar o conhecimento, porém não há uma adequação desejada destes modelos para BD textuais. Por isso, criamos o TDM (Textual Database Model). O TDM é um modelo criado para tratar especificamente o problema de representar o conhecimento em BD textuais. Os bancos de dados que manipulam textos possuem características singulares, tais como: quase não apresentam alterações, inclusões e deleções em seus termos formadores (linha, parágrafos e capítulos). Note que a natureza estática destes BDs deve-se ao fato de que um texto uma vez pronto dificilmente será alterado, principalmente quando o texto é uma narrativa, o que é o nosso caso. Entretanto, isto não diminui muito a complexidade de manipulação de textos, pois as consultas podem ser feitas por eventos ou estruturas e não apenas utilizando-se conectivos lógicos sobre um dicionário pré-estabelecido.

O TDM possui uma estrutura baseada em um formalismo lógico, formado por axiomas da lógica de predicados e teoria dos modelos. A partir deste formalismo elaboramos um esquema gráfico que representa o conhecimento textual. Este esquema gráfico tem como finalidade o recebimento de um texto (sua entrada) e a partir deste texto representá-lo de forma que possam ser feitas recuperações de informações, teste de similaridade entre textos (através de atribuição de graus de similaridade). Para que se obtenha um poder máximo de expressividade do modelo, torna-se necessário que seja escolhido a priori o nível de representação desejado, pois a partir da definição do nível de representação é que torna-se possível encontrar o foco de atenção do texto. Por último, são mostradas as vantagens e desvantagens em se utilizar o TDM como um modelo de representação do conhecimento.

4.1 Esquema Gráfico

O TDM pode ser visto como um DAG (Direct Acyclic Graph) composto por arcos "E" (relacionamentos "Part_of") e arcos "OU" (relacionamentos "Is_a"). Os nós "E" representam os estados que são considerados solucionados apenas se *todos* os seus sucessores forem solucionados. Os nós "OU" representam os estados que são considerados solucionados se *um* de seus sucessores for solucionado. O uso de relacionamentos "Is_a" e "Part_of" torna-se importante, pois a partir destes relacionamentos é possível fazer generalizações e especificações que serão de grande importância ao fazer a comparação entre os textos. Um exemplo do uso do TDM pode ser visto no exemplo mostrado na figura 8.

Resumo de um texto narrativo:

Peer Gynt é um irresponsável e sonhador. Tem um grande sonho : ser um imperador destemido. Vive criando casos, sempre com a ajuda de sua mãe – seduz a noiva do rei da floresta, Solveig. Torna-se um sujeito canastrão. Por último torna-se um contador de histórias onde ele é sempre o herói.

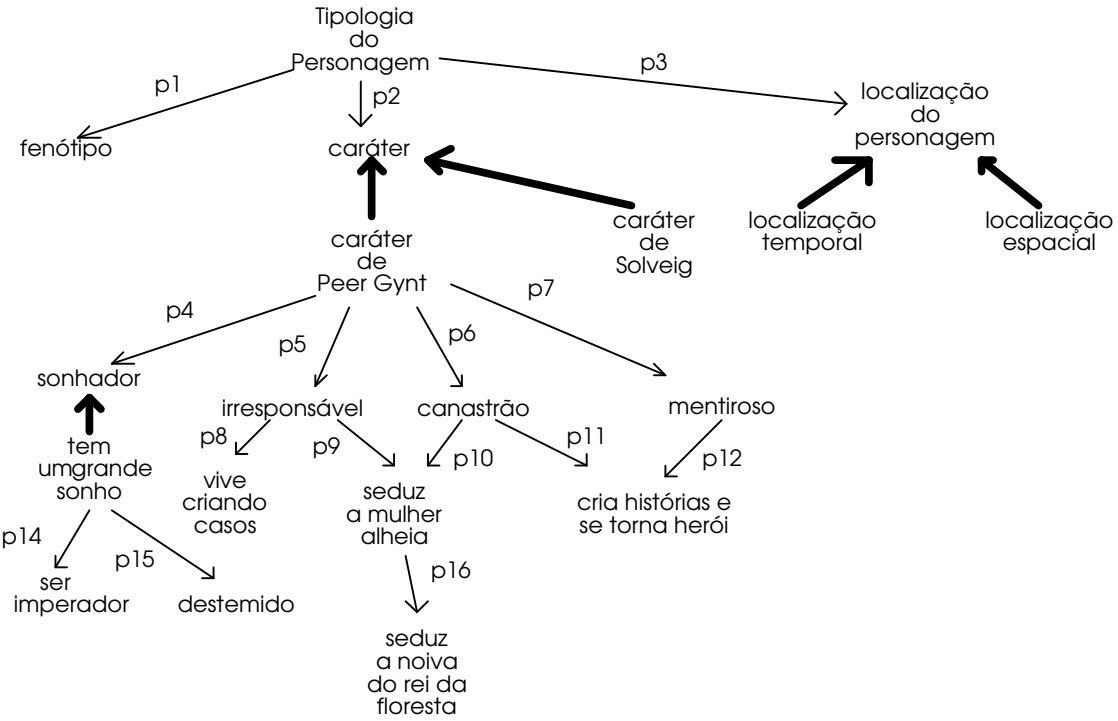

Figura 8 - Um exemplo de TDM para um texto narrativo

As setas largas representam os relacionamentos do tipo "Is_a" e as setas estreitas os relacionamentos do tipo "Part_of".

4.2 Estrutura Formal

A estrutura do TDM é formada por uma coleção de axiomas e regras que podem ser vistas como uma codificação lógica de um grafo acíclico E/OU. Estes axiomas e regras representam as relações, restrições e heurísticas que podem ser utilizados na manipulação do texto modelado.

Um TDM T pode ser definido como sendo uma estrutura lógica formado por um dupla :

$$\mathbf{T = <S_t , G_t>}$$

♦ $\mathbf{S_t}$ é o conjunto formado por todos os axiomas e regras que descrevem o texto. i.e. S_t é a representação estrutural de um texto.

$$\mathbf{S_t = <A_a , A_p , R >}$$

onde A_a são os axiomas de abstração;
A_p são os axiomas das partes e
R são as regras heurísticas de manipulação

Uma representação textual é composta por diversos axiomas de abstração, axioma das partes e por diversas regras.

Cada axioma de abstração A_a é um predicado da forma:

abs(E_j,E_i) ou **E_j abstrai E_i**

Formalmente,

Sejam dois eventos quaisquer E_i, $E_j \in H_E$. Têm-se que E_i abstrai E_j , se e somente se, para todo evento E_i, temos E_j, mas não vale o contrário. Valendo também a transitividade em lógica de segunda-ordem. Ou seja, diz-se que E_i abstrai E_i, se e somente se, são satisfeitos os seguintes axiomas:

(I) $\forall E_i, E_j\ [abs(E_j, E_i) \leftrightarrow (\forall x.E_i(x) \rightarrow E_j(x))]$

(II) $\forall E_i, E_j, E_k\ [abs(E_i, E_j) \wedge abs(E_j, E_k) \rightarrow abs(E_i, E_k)]$

Caso seja necessário que os eventos dos axiomas acima sejam do mesmo tipo, é necessário apenas modificar o axioma (I) da seguinte forma (usando lógica poli-sortida):

(I) $\forall E_i, E_j\ [abs(E_j, E_i) \leftrightarrow ((\forall x:t)\ .E_i(x) \rightarrow E_j(x))]^{+1}$, $t \in$ conjunto de tipos

Note que a redução deste axioma para lógica de primeira-ordem poderá ser realizada usando relativização:

$[(\forall x:t)\ \varphi]^{+} := \forall x[tipo_t(x) \rightarrow \varphi]$

O simétrico da abstração é a especialização, que pode ser definida como sendo:

esp(E_i , E_j), ou E_i especializa E_j.

Ou seja, **esp(E_i , E_j) $\rightarrow$ abs(E_j,E_i).**

Cada axioma das partes A_p é um predicado da forma:

$(\forall x.E_0(x) \rightarrow E_1(p_1(x)) \wedge \ldots \wedge E_n(p_n(x))) \wedge \kappa$,

onde E_o é um evento qualquer
$E_1, \ldots, E_n$ são os eventos que são componentes direto de E_0
$p_1, \ldots, p_n$ são os passos ou funções de mapeamento

Todo passo ou função de mapeamento p_i, $1 \leq i \leq n$, deve ser rotulado, visando uma distinção entre eles.

---

[1] O símbolo + é utilizado para mostrar que a fórmula está codificada em lógica poli-sortida.

O conjunto de regras R, que também faz parte do modelo estrutural S_t, é formado por predicados ou inferências que visam capturar o conhecimento ímplicito do TDM ou ainda fazer a remoção dos caminhos improváveis de uma consulta ao BD textual. Algumas das regras criadas podem ser vistas logo abaixo:

Regra da Desconectividade - **(DESC)**

Seja C_A o conjunto formado por todos os eventos participantes dos axiomas de abstração e C_P o conjunto de todos os axiomas das partes. Então temos que

$$DESC = \{ \forall E_i / E_i \notin C_A \vee E_i \notin C_P \}$$

Ou seja, DESC é o conjunto formado por todos os eventos desconexos, que não possui nehuma relação ligando-o a outro evento. Note que em BD textuais é comum ter este tipo de evento desconexo, pois o projetista do TDM pode ter colocado o evento desconexo apenas como oração ou partícula de realce. Ao ser feito generalizações da estrutura do TDM todos os eventos desconexos serão eliminados.

Regra da Similaridade - **(SIMP)**

Dado um TDM inicial T_i, um TDM a ser comparado T_j e um objetivo de comparação O_c, a similaridade é dada pelo teste de similaridade pura (SIMP).

Similaridade Pura - **(SIMP)**

Seja CT_i um conjunto de eventos pertencentes ao TDM inicial T_i e CT_j um conjunto de eventos pertencentes ao TDM T_j, onde $1 \leq i,j \leq n$ e n é o número de eventos em cada TDM. A similidaridade entre dois textos (já mapeados em seus respectivos TDMs) é dada por

$$[\exists(CT_i) \in T_i \wedge \exists(CT_j) \in T_j / CT_i \wedge CT_j \models O_v]$$

Ou seja, SIMP é teste booleano de similaridade obtido a partir de um conjunto de eventos do TDM inicial T_i e um conjunto de eventos do TDM a ser comparado T_j; tal que o objetivo da comparação O_c é conseqüência lógica dos conjunto CT_i e CT_j. desta forma podemos comparar partes de textos ou textos completos apenas modificando o objetivo de comparação O_c.

- G_t é o conjunto formado pelos arcos "Is_a", "Part_of", os nós de eventos e as probabilidades associadas a ocrrência de determinados eventos.

Conforme visto anteriormente, o modelo gráfico do TDM pode ser visto como um DAG E/OU, com as seguintes mudanças:

Os nós são eventos (ações e metas)

Os arcos são relações do tipo "Is_a" e "Part_of". Estes arcos são representados formalmente como axiomas lógicos e a nível de implementação como predicados ao estilo PROLOG.

4.3 Forma de Raciocínio

* O TDM basea-se no raciocínio por generalizações e especificações.
* O centro de todo o modelo está na definição do *foco de atenção*, pois cada TDM possui seu próprio foco que pode ser usado para generalizações e testes de similaridade entre TDMs

* As buscas de informações são efetuadas neste modelo da mesma forma que nos grafos E/OU. Sendo os arcos deste grafo utilizados para tratar herança (arcos "Is_a" e "Part_of")

* O controle da busca é feito com o uso de probabilidades associadas aos eventos

* É possível, de uma maneira, restrita realizar o raciocínio meta-nível, ou seja, fazer uso de *introspecção*

4.4 Vantagens e Desvantagens

Vantagens :

i. Modelo direcionado para o foco de atenção (tema) de cada texto

ii. Tratamento de generalizações e especificações

iii. Facilidade de adquirir novas heurísticas

iv. Controle do espaço de busca dos estados possíveis a resposta de uma solicitação

v. Não requer que o conhecimento seja decomposto em termos primitivos

vi. Possui uma semântica formal baseada em lógica

vii. Pode tratar quantificadores

viii. Uso de regras e heurísticas acopladas à estrutura formal

ix. Tratamento de similaridade entre textos

x. As relações são representadas de acordo com o nível de representação escolhido pelo projetista ao representar o conhecimento através do TDM

Desvantagens :

i. Não há o tratamento de objetos

ii. O número de relações pode ficar muito grande, dependendo do nível escolhido pelo projetista

iii. Uso informal de probabilidades

5. Conclusões

A finalidade inicial deste trabalho foi a elboração de um elo entre os diversos paradigmas de representação do conhecimento e BD textuais. Para isso, foram escolhidos alguns dos principais modelos de representação do conhecimento (lógicas - lógica de predicados e lógica modal de ação, gramáticas - gramática sintática e gramática semântica, redes semânticas, "frames", sistemas de produção e dependência conceitual). Todavia, a maioria destes modelos não se mostraram satisfatórios ao ser utilizado com BD textuais, sendo então feito um estudo comparativo mostrando as principais vantagens e desvantagens de cada modelo.

Contudo, devido a inadequação da maioria dos modelos de representação do conhecimento existente, criamos um modelo próprio para representar BD textuais - TDM (Textual Database Model). Este modelo é dividido em duas partes: o esquema gráfico e o esquema formal. No esquema gráfico, o usuário mapea seu texto em uma estrutura baseada em um grafo de relacionamentos hierarquicamente organizados. Toda a manipulação deste garfo, assim como as heurísticas e a sua formalização foram vistos no esquema formal. Por último mostramos as principais vantagens e desvantagens do TDM.

Bibliografia

[BaFe81] Barr A. e Feigenbaum E. A. "The Handbook of Artifical Intelligence", volume I, William Kaufman Inc., 1981

[Berw86] Berwick R. C. "The Acquisition of Syntatic Knowledge", MIT Press, segunda edição, 1986

[Bitt91] Bittencourt G. "Representação de Conhecimento", VIII Simpósio Brasileiro em Inteligência Artificial, Brasília, BR, 1991

[Bode87] Boden M. A. "Artificial Intelligence and Natural Man", segunda edição, Basic Books Inc., 1987

[ChWi76]Charniak E. e Wilks Y. "Computational Semantics: An Introduction to Artificial Intelligence and Natural Language Comprenhension", North-Holland Publish Company, 1976

[Elco83] Elcock, E. W. "How Complete are Knowledge Representation System", IEEE, 1983

[EvAr92] Evsukoff A. e Aragon D. "Uma Formalização de Redes Semânticas para um Sistema Híbrido de Representação de Conhecimento", SBIA92, Rio de Janeiro, RJ., 1992

[Ellm89] Ellman T. "Explanation Based Learning: Survey of Programs and Perpestives", ACM Computing Survey, volume 21, número 2, 1989

[Gard88] Gärdenfors P. "Knowledge in Flux: Modeling the Dynamics of Epistemic States", The MIT Press, Cambridge, Ma., 1988

[GaVa89] Gardarin G. e Valduriez P. "Relational Databases and Knowledge Bases", Addison-Wesley Publish Company, 1989

[Gold88] Goldman R. e Charniak E. "A Probabilistic ATMS for Plan Recognition", AAAI-88, Workshop on Plan Recognition, 1988

[Hash90] Hashim, S. H. "Exploring Hypertext Programming: Writing Knowledge Representation and Problem-Solving Programs", Windcrest ed., 1990.

[Maes86] Maes, P. "Introspection in Knowledge Representation", Proceedings of ECAI 7, Brighton, UK, 1986

[Mars91] Marshall R. "Manipulating Full-Text Scientific Database: A Logical Based Semantic-Pragmatic Approach", The Computer Journal, volume 34, número 3, 1991

[Matt91] Mattos, N. M. "An Approach to Knowledge Base Management", Lecture Notes in Artificial Intelligence, Springer-Verlag, 1991

[Mins86] Minsky M. "The Society of Mind", Simon & Schuster, NY, 1986

[Nils80] Nilsson N. J. "Principles of Artificial Intelligence", Tioga Publish, Palo Alto, Ca., 1980

[Nogu94] Nogueira J. H. M. "Análise e Implementação de uma Teoria Formal de Reconhecimento de Planos", (a ser publicado como Relatório Técnico, Departamento de Informática, PUC, Rio de Janeiro, RJ, 1994)

[Nogu91a] Nogueira J. H. M. "Automatização do Raciocínio do Senso Comum", Anais da 1a. Reunião de Iniciação à Pesquisa, UECE, 1991

[Nogu91b]Nogueira J. H. M. "Desmistificando a Inteligência Artificial", Anais do 43o. Congresso da Sociedade Brasileira para o Progresso da Ciência, Rio de Janeiro, RJ, 1991

[Nogu92a]Nogueira J. H. M. "Paradigma de Sistemas Especialistas e Redes Neurais", Anais do III Encontro de Computação, Fortaleza, CE, 1992

[Nogu92b] Nogueira J. H. M. "Prova Automática de Teoremas", Anais do 44o. Congresso da Sociedade Brasileira para o Progresso da Ciência, São Paulo, SP, 1992

[Nogu92c] Nogueira J. H. M. "Uso da Lógica na Representação do Conhecimento", Anais do 44o. Congresso da Sociedade Brasileira para o Progresso da Ciência, São Paulo, SP, 1992

[Rich93] Rich E. "Artificial Intelligence", second edition, McGraw-Hill Company, 1993

[RoBe92] Rodrigues O., Benevides M. "Prolog Modal de Ação", Anais do IX Simpósio Brasileiro de Inteligência Artificial, Rio de Janeiro, RJ, 1992

[SAB94] Salton G., Allan J. e Buckley C. "Automatic Structuring and Retrieval of Large Text Files", Communications of the ACM, volume 37, número 2, 1994

[Smea92] Smeaton A. F. "Progress in Application of Natural Language Procesing to Information Retrieval Task", The Computer Journal, Cambridge University Press, volume 35, número 3, 1992

[Smit82] Smith P. J. "Reflection and Semantics in a Procedural Language", Ph.D. Thesis, M.I.T., Cambridge,MA, 1982.

[Sowa88] Sowa, J. F. "Knowledge Representation in Database, Expert Systems, and Natural Language", Working Conference on the Role of Artificial Intelligence in Database and Information Systems, Guangzhou, China, 1988

[Sowa84] Sowa J. F. "Conceptual Structures: Information Processing in Mind and Machine", Addison-Wesley Publish Company, 1984

[Vick86] Vickery, B. C. "Knowledge Representation a Brief Review", The Journal of Documentation, volume 42, número 3, 1986

[Wins84] Winston P. H. "Artificial Intelligence", segunda edição, Addison-Wesley Publish Company, Ma., 1984