

Boletim Técnico da Escola Politécnica da USP
Departamento de Engenharia de Computação e
Sistemas Digitais

ISSN 1413-215X

BT/PCS/9701

Uma Experiência na
Construção de um Tradutor
Inglês-Português

Jorge Kinoshita
João José Neto

São Paulo - 1997

O presente trabalho é parte da tese de doutorado apresentada por Jorge Kinoshita, sob orientação do Prof. Dr. João José Neto: "Aspectos de Implantação de uma Ferramenta de Auxílio à Tradução Inglês-Português", defendida em 25/03/97, na EPUSP.

A íntegra da tese encontra-se à disposição com o autor e na Biblioteca de Engenharia de Eletricidade da Escola Politécnica da USP.

Kinoshita, Jorge

Uma experiência na construção de um tradutor
inglês-português / J. Kinoshita, J. José Neto. --
São Paulo : EPUSP, 1997.

p. -- (Boletim Técnico da Escola Politécnica da
USP, Departamento de Engenharia de Computação e Sis-
temas Digitais, BT/PCS/9701)

1. Linguagem natural (Inteligência artificial)
2. Tradução por computador I. José Neto, João II.
Universidade de São Paulo. Escola Politécnica. De-
partamento de Engenharia de Computação e Sistemas
Digitais III. Título IV. Série

ISSN 1413-215X

CDD 006.35
418.02

Uma experiência na construção de um tradutor inglês-português

Jorge Kinoshita

João José Neto

Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, trav. 3 no. 158
05508-900 - São Paulo - SP
fone 818-5392
e-mail: jkinoshi@usp.br

Resumo

Neste trabalho propomos a arquitetura de uma ferramenta de auxílio à tradução Inglês-Português do tipo *transfer*, isto é, que realiza a análise sintática da sentença em Inglês, obtém uma estrutura em árvore e gera a tradução da sentença a partir desta árvore. Implementamos o protótipo de uma ferramenta que apresenta a capacidade de aprendizado através da interação com o usuário durante a análise sintática. A gramática foi construída através da união de conceitos provenientes da teoria X-barra e da gramática de ligações. Na fase de síntese, as melhores palavras são escolhidas através de exemplos ou da generalização de exemplos.

1. Introdução

Apresentamos um mecanismo de traduzir uma sentença do inglês para o português observando como as palavras se organizam aos pares dentro de uma sentença. Estes pares de palavras podem ser dispostos em uma árvore e servir de base para a tradução. Os pares de palavras são conhecidos como relações sintagmáticas. Por exemplo, na sentença:

The girl will come whenever she wants.

temos uma relação entre a palavra “she” e “want”. A sentença do exemplo é composta das seguintes relações sintagmáticas:

[[the] girl]
 [[girl] come]
 [[will] come]
 [come [whenever]]
 [whenever [wants]]
 [[she] wants]

que unidas correspondem à árvore [[[The] girl] [will]come [whenever [[she] wants]]]] que pode ser dada na forma indentada:

```
[      [[The] girl]
      [will]
come
      [whenever
        [[she] wants]]
      ]
]
```

Vamos chamar esta estrutura de árvore de palavras.

Um banco de dados que contenha exemplos de relações pode ser útil tanto na fase de análise para se resolver casos de ambigüidade estrutural quanto na fase de transferência, a fim de se ter a tradução das relações. Por exemplo, cada uma das relações pode ser traduzida individualmente como:

relação em inglês	tradução
[[the] girl]	[[a] menina]
[[girl] come]	[[menina] vir/T]
[[will] come]	[vir/futuro]
[come [whenever]]	[vir/T [[sempre] que]]
[whenever [wants]]	[[sempre] que [querer/pressubj]]
[[she] wants]	[[ela] querer/T1]

A tradução das relações sintagmáticas podem estar dispostas na seguinte árvore:

[[[a] menina] vir/futuro [[sempre] que [[ela] querer/pressubj]]]

que é traduzido como:

a menina virá sempre que ela quiser.

Dessa forma, podemos ver todo o processo de tradução baseado em relações sintagmáticas. O mecanismo de usar sempre duas palavras como contexto para a tradução nem sempre é válido. Existem casos em que é necessário cobrir um contexto maior. Um exemplo deste caso é no uso de expressões idiomáticas. Estes casos podem ser tratados diretamente na gramática. Existem casos ainda mais complexos que necessitam de uma análise do texto para se resolver ambigüidades. Por exemplo, a sentença pode referenciar outros elementos do texto através de pronomes. A correta tradução depende de se saber quem é o referente. Embora não estejamos tratando destes casos (até que ponto eles devem ser tratado em uma ferramenta interativa de auxílio à tradução?), uma possível heurística é escolher aqueles referentes do texto que possuem o maior número de relações sintagmáticas com as palavras que se relacionam com o pronome. Uma heurística que trate do processamento de texto é complexa e foge do escopo desta tese.

Para automatizarmos a tradução dessa forma é necessário:

- levantar a árvore de palavras. Isto é feito através de uma gramática. Encontramos na literatura duas gramáticas que serviram de base ao nosso analisador sintático: a teoria X-barra e a gramática de ligações.
- associar uma tradução a cada relação sintagmática da árvore de palavras.

2. A teoria X-barra

Em qualquer sentença podemos identificar grupos de palavras que estão relacionadas entre si, formando blocos coesos. Por exemplo: na sentença "O menino que corria pretendia ganhar as Olimpíadas" podemos identificar grupos de palavras como: "O menino que

corria", "pretendia ganhar as olimpíadas" ou "ganhar as Olimpíadas". Já determinados grupos não apresentam uma coesão como "menino que", "menino Olimpíadas", etc. Estes grupos coesos de palavras são conhecidos na literatura ([Lyons-79], [Lopes- 75]) como *sintagmas*. Cada sintagma possui uma palavra que domina as outras. Por exemplo: em "o menino que corria" a palavra dominante, ou seja, aquela que dá o sentido mais geral para este sintagma é "menino". As outras palavras especificam ou modificam a palavra "menino". Se omitirmos o modificador "que corria" ainda teremos uma sentença válida como "O menino pretendia ganhar as Olimpíadas". Por outro lado, se omitirmos "menino" a sentença já passa a soar de forma mais estranha ("O que corria pretendia ganhar as Olimpíadas"). A palavra mais importante é conhecida como núcleo do sintagma (em inglês: *head*). Ela comanda as demais palavras. A concordância em pessoa, gênero e número é ditado pelo núcleo. Por exemplo, o sintagma "A menino que corremos" soa estranho. Estes fatos já foram observados por diversos linguistas, e em particular por aqueles ligados à teoria X-barra.

Apenas as palavras pertencentes a certas categorias gramaticais podem servir de núcleo a sintagmas. Um exemplo, é o caso dos substantivos (ex: "menino"). Ao redor de um substantivo podemos agrupar categorias como o artigo (ex: "o menino", "um menino") o adjetivo (ex: "menino amarelo"), etc. Há também palavras que não podem servir de núcleo a sintagmas como por exemplo, artigos, pronomes possessivos, etc. A teoria X-barra identificou apenas 4 categorias que podem servir de núcleos a sintagmas: nome (substantivo), adjetivo, verbo e preposição. A teoria X-barra propõe regras para a formação de sintagmas. Estas regras poderiam ser peculiares a cada um destes quatro tipos de sintagmas, porém foi formulado que existe um princípio geral que vale para os quatro tipos de sintagmas. O nome X-barra deriva justamente desta observação. A letra X corresponde a uma variável que pode assumir uma destas quatro categorias, de forma abreviada N (nome), A (adjetivo), V (verbo) e P (preposição).

A regra de formação de um sintagma cujo núcleo é uma palavra da categoria X estabelece que outros sintagmas podem concatenar-se à esquerda ou à direita do núcleo formando o sintagma X' da seguinte forma:

$$X' \rightarrow \langle \text{modificadores_esquerda} \rangle, X, \langle \text{modificadores_direita} \rangle.$$

O nome "X-barra" para esta teoria deriva justamente deste tipo de regra. A representação de X com uma barra acima é difícil na maioria dos editores de texto e é comum se encontrar a representação X'. Dizemos que X' é a projeção de X. Os sintagmas colocados como <modificadores> modificam X. Por exemplo: no caso de termos <modificadores_esquerda> = "yellow", N = "boy" e <modificador_direita> = vazio temos o sintagma "yellow boy" correspondente a N'.

X' também é projetado em um segundo nível como:

$X'' \rightarrow \langle \text{especificadores} \rangle, X'$.

Tanto no português quanto no inglês, os especificadores situam-se à esquerda de X'. Os especificadores correspondem a sintagmas que especificam X. Exemplo, no caso de um sintagma nominal (aquele que tem N como núcleo), os artigos correspondem a especificadores (ex: *the*). Podemos assim obter N'' como "*the yellow boy*". Ainda não existe um consenso absoluto quanto à distinção entre *especificador* e *modificador à esquerda* como pode ser visto em [Raposo-92].

A teoria X-barra procurou generalizar este mecanismo de formação de sintagmas para definir a formação de sentenças e dessa forma nasceram outras duas classes: a I que explica a formação de sentenças simples, e C, que explica a formação de sentenças (em geral, as orações subordinadas) que são introduzidas por uma partícula (ex: "*se ele viesse ...*"). Palavras da categoria I correspondem a modais, como *will* e *do*. Ao se usar tais modais o tempo do verbo principal do predicado (dado pelo sintagma verbal VP) é colocado na forma infinitiva. Nem sempre em inglês é feito uso de modais (ex: *the boy runs*). Assim, nem sempre a categoria I corresponde a palavras. Pode-se também associar uma estrutura que indica a concordância em número e pessoa do sujeito com o predicado, bem como o tempo em que o predicado é flexionado. As palavras pertencentes à categoria C são aquelas que introduzem orações subordinadas, tais como *if*, *that*, etc. Desta forma, a teoria X-barra procura explicar toda a formação de sentenças a partir de seis categorias gramaticais apenas.

Infelizmente, isto não pode ser aplicado na prática desta forma tão simples. Fizemos algumas experiências práticas construindo uma gramática com base na teoria X-barra implementada em DCG e os resultados ficaram aquém do esperado. O problema reside na enorme ambigüidade que surge nos diversos níveis do processamento de sentenças. Existem

diversas palavras em Inglês que podem servir a mais de uma categoria morfológica. Observemos um caso simples: Em "*The boy will arrive*", a palavra "will" pode corresponder ao substantivo: "testamento". "boy will" pode corresponder a "testamento do menino", por uma regra que informa que um Nome pode modificar outro Nome (ex: "computer science" = "ciência de computador"). Este fato pode levar a traduções pobres ou incorretas, ou então a um número grande de *backtrackings*. Concluimos que a teoria X-barra pode fornecer uma base para orientar a construção de nossa gramática, porém uma gramática demasiadamente genérica proporciona um grande número de soluções, dentre as quais, em geral, apenas uma delas é aceitável. Portanto é necessário eliminar as soluções erradas.

3. A influência da Gramática de Ligações

A gramática de ligações (*Link Grammar*) [Sleator-91] permite a realização da análise sintática de uma enorme variedade de sentenças. O resultado de uma análise sintática efetuada com base nesta gramática é um grafo que mostra como as palavras se interligam dentro das sentenças. [Sleator-91] levantou um dicionário que associa a cada palavra da língua inglesa uma regra que explica como ela se liga a outras palavras. Por exemplo, o substantivo "menino" pode se ligar a um artigo ("o-menino"), bem como a um verbo, servindo como o núcleo do sujeito ("menino-corria"), e assim por diante. Podemos ver que existem vários tipos de ligações. Estes tipos recebem um nome. Por exemplo, uma ligação do tipo "S" estabelece uma relação entre o sujeito e predicado como na ligação entre as palavras "menino" e "corria". Uma regra associada a uma palavra deve explicar quais os tipos de ligações que esta palavra pode realizar com outras palavras. Um caso curioso ocorre com o verbo "hope". Na sentença "*I hope this program works*" podemos observar a oração subordinada "*this program works*" conectando-se ao verbo "hope" sem a necessidade de nenhuma partícula (por exemplo "*that*"). Isto não é possível para o verbo "sell": não podemos dizer "*I sell this program works*". Parece-nos impossível ligar o verbo "sell" a uma oração subordinada sem o auxílio de nenhuma partícula. Este fato é estabelecido no dicionário da gramática de ligações. A ligação do tipo CL representa a ligação entre um verbo e uma oração subordinada sem uma partícula que introduza tal oração. No dicionário

da gramática de ligações, pode-se ver que a regra associada à palavra *hope* permite uma ligação do tipo CL enquanto que a regra associada à *sell* não permite.

Retornando à teoria X-barra, podemos entender um sintagma como sendo composto de um núcleo, mais sintagmas subordinados. Por exemplo, o sintagma verbal é composto por um verbo como núcleo e sintagmas subordinados. Um caso de sintagma subordinado é a oração simples ("ip"). Dessa forma poderíamos ter a regra:

`vp_mod_e --> ip.`

que especifica que um modificador a esquerda de um sintagma verbal ("vp_mod_e") é uma oração. Esta regra pode reconhecer construções erradas como "I sell this program works". Isto pode ser evitado se associarmos um "flag" a esta regra de tal forma que ela é habilitada para o verbo "hope" mas desabilitada para o verbo "sell". A regra modificada se torna:

`vp_mod_e(Verbo_Principal) --> {cl(Verbo_Principal)},ip.`

onde `cl(X)` retorna verdadeiro se a palavra X na gramática de ligações contém uma ligação do tipo CL.

Este mecanismo pode ser generalizado para os diversos tipos de ligações encontrados no dicionário da gramática de ligações. Resumindo, a nossa proposta é se ter as regras gerais como as propostas pela teoria X-barra associadas a condições relacionadas às palavras. Estas condições foram observadas no dicionário da gramática de ligações.

4. A interação com o usuário

Na fase de análise permitimos a interação com o usuário, com o objetivo de atualizarmos a base de dados que descreve como as diversas palavras se ligam aos outros sintagmas dentro das sentenças. Caso esta base de dados esteja completa, tal interação poderá ser suprimida. Dessa forma, a ferramenta pode trabalhar em dois modos: no modo aprendizado, e no modo normal.

No modo aprendizado, o usuário pode interagir com o sistema durante a fase de análise sintática.

Observemos as seguintes sentenças:

a) He is going to school.

b) He is going to cry.

School pode servir como verbo, no sentido de educar: "She is well schooled in literature". Dessa forma a) poderia ser erroneamente traduzida como "Ele está indo educar" da mesma forma que b) poderia ser traduzido como "Ele está indo chorar". A fim de ter uma tradução mais próxima da realidade, podemos permitir que o usuário interfira no processo de tradução. Caso sejam apresentada ambas as opções ao usuário:

ir educar

ir para escola

então ele poderá escolher qual a opção que mais adequada ao seu caso. Ao escolher a opção "ir para escola" dois fatos podem ser aprendidos:

- *school* é um substantivo que não requer necessariamente um determinante.
- existe uma relação entre o verbo *go* e a preposição *to*.

No modo "aprendizado", a regra de formação de sintagma pode habilitar a associação do núcleo com qualquer outro sintagma subordinado, independentemente de esta associação estar registrada no dicionário da gramática de ligações. Caso esta associação seja feita, pede-se a confirmação do usuário. Ao confirmar a formação do sintagma, uma informação foi obtida: o núcleo pode se associar a um novo tipo de sintagma. Esta associação deve ser então incorporada ao dicionário, para ser usada daí em diante. Dessa forma, o reconhecedor tende a se tornar cada vez melhor, através do uso.

5. A tradução das relações sintagmáticas

Uma vez feita a análise sintática, a árvore de palavras já está montada. O processo de síntese se baseia em extrair relações sintagmáticas da árvore de palavras e traduzi-las.

Levantar todas as relações da língua inglesa é uma tarefa trabalhosa e que consome muito espaço de armazenamento. Podemos observar que as palavras podem ser dispostas em grupos correlatos. Por exemplo: *maçã*, *laranja* e *limão* podem pertencer ao grupo das *frutas*. Todas estas palavras se comportam de uma forma semelhante dentro das sentenças. Por exemplo: todas podem ser comestíveis, ou seja, podem ser o núcleo do objeto direto do

verbo "comer". Podemos supor como uma palavra pode se combinar a outras observando apenas o fato de que ela pertence a um determinado grupo. Por exemplo, se soubermos que "*plimbis*" (nome inventado) é uma fruta, então podemos supor que "*plimbis*" é comestível. Dessa forma podemos deduzir relações entre palavras se conhecermos, além das palavras, a forma como elas estão agrupadas. Este agrupamento de palavras pode ser representado através de uma estrutura hierárquica, na qual passamos a ter grupos e subgrupos de palavras. Estamos utilizando uma estrutura hierárquica com base apenas em substantivos.

Um par de palavras correspondente ao nó-pai e nó-filho extraídos da árvore pode corresponder a uma relação sintagmática. Esta relação será chamada de *relação de entrada*. Por exemplo, da árvore proveniente da análise de "*I eat plimbis*" pode-se extrair a relação de entrada (verbo: eat - objeto: plimbis); que vamos abreviar por "eat-plimbis". A tradução de uma relação de entrada é feita com base na estrutura hierárquica de substantivos (ex: *plimbis* é uma fruta) e em exemplos de outras relações traduzidas (ex: *eat/chupar-orange/laranja*). Estes exemplos já estão previamente armazenados em um banco de dados. A tradução é feita procurando-se traduzir a relação de entrada de acordo com a tradução dos exemplos do banco de dados através do processo de generalização.

O processo de generalização é aplicado apenas às relações que envolvam substantivos. A outra palavra que se relaciona ao substantivo, aliada à informação do tipo de relação (ex: verbo-objeto) será chamada *método* (nomenclatura inspirada na programação por objetos). Este processo envolve os seguintes passos:

1 - verificar todas as relações do banco de dados que possuam o mesmo método da relação de entrada. Por exemplo, o resultado desta busca poderia ser: *eat/chupar-orange/laranja*, *eat/comer-pizza/pizza*

2 - para cada substantivo das relações encontradas verificar aquele que se assemelha mais ao substantivo da relação. Isto é feito observando qual substantivo possui o maior número de ancestrais em comum na hierarquia de substantivos. No caso é *orange*, que também corresponde a uma fruta. Vamos chamar esta relação de *relação próxima*.

3- a tradução da relação de entrada é feita tomando-se a tradução do método da relação próxima (no exemplo: *eat/chupar*) e tomando-se a tradução do substantivo do dicionário bilingüe. A tradução do substantivo é tomada observando-se também a hierarquia de palavras. Por exemplo: vamos supor que *plimbis* pudesse ser a fruta *plimbom* ou o avião

pluft. A tradução de *plimbis* escolhida é *plimbom*. Assim a tradução final é "chupar plimbom".

6. Exemplos

O nosso programa de auxílio à tradução está implementado na forma de um laboratório, encontrado-se, portanto, em constante evolução. Por enquanto vamos apresentar diversos exemplos que podem ser traduzidos através de nossa ferramenta. Existem por enquanto diversos casos ainda não tratados. Existem diversas regras da gramática ainda a serem inseridas, bem como diversos aprimoramentos necessários ao banco de dados.

Nos exemplos abaixo, vamos apresentar o resultado da análise sintática, ou seja, a árvore de palavras, e depois os passos seguintes que levam à síntese da sentença. Mesmo que a nossa ferramenta muitas vezes não gere a melhor tradução possível para a sentença fornecida, é fácil observar que isto se deve, na maioria dos casos, à falta de informação de na base de dados ou à falta de regras da gramática e não à deficiência do método empregado. Em muitos casos resolvemos ou propomos soluções. De qualquer forma, foi adquirida até agora uma boa experiência de como expandir a ferramenta. No primeiro exemplo, estamos apenas comentando a saída gerada após a entrada da sentença. No segundo exemplo, mostramos um caso de generalização sobre relações.

7.1. Primeiro exemplo

- Sentença

Corresponde à transcrição direta da sentença de entrada. Ex:

the boy runs

- Morfologia

Para cada palavra da sentença de entrada é apresentada sua possível categoria morfológica. Cada categoria está numerada dependendo da posição da palavra correspondente na sentença. Ex:

- 1 - `d(_OCB8,the)`
determinante "the", _OCB8 correspondente a uma variável em aberto do PROLOG indicando que o número singular ou plural está indefinido.
- 2 - `n(s,boy)`
"boy" = substantivo no singular (dado pelo parâmetro s).
- 3 - `n(pl,run)`
"run" = substantivo no plural (pl). corresponde a "corridas".
- 3 - `vt((3 , s),run / pres)`
"run" = verbo transitivo na terceira pessoa do singular, com a raiz "run" no tempo presente. Neste exemplo, observar que a *run* não é um verbo transitivo. Todos os verbos estão erroneamente, a princípio, declarados como transitivos. A característica quanto à transitividade pode ser aprendida.

- Sintaxe

Neste item apresentamos uma possível interação com o usuário. A interação ocorre caso o analisador não tenha reconhecido a sentença. Através desta interação, a base de dados é atualizada, introduzindo as cláusulas rela (indicam como uma palavra se relaciona com outros sintagmas e se referem principalmente à forma como um verbo transitivo se liga a cada tipo de objeto) bem como declarar um verbo como intransitivo quando não for encontrado complemento. A intransitividade é "descoberta" durante a análise. Ex:

```
run - vi?(s,n): s
>>vi(run,correr)
```

Durante o decorrer de nossos exemplos, sempre que uma cláusula for precedida do sinal ">>", isto indica que tal cláusula foi introduzida no banco de dados. Neste exemplo, foi introduzida a informação de que *run* pode se comportar como um verbo intransitivo, mas a informação de que ele é transitivo permaneceu no banco. Isto de fato acaba não influenciando na análise de futuras sentenças que contenham *run* porque não foi especificado de que maneira o verbo *run* se une ao seu objeto. Caso o usuário assim deseje, ele pode editar o banco de dados e remover a cláusula *v(run,correr)*.

- Árvore de Palavras

Neste item apresentamos a árvore de palavras resultante da análise sintática da sentença. Ex:

```
[
  [
    suj *                               /* indica sujeito */
    [
      [
        det(_OF94,the) - the
      ]
      nomi(s,boy,0,[],[]) - boy
    ]
    vt((3 , s),run / pres) - runs
  ]
]
```

- Síntese

Caso a ferramenta esteja trabalhando no modo de extrair relações (ex: a relação entre o núcleo do sujeito "boy" e o verbo "run") a partir da árvore (pelo comando *liga(junta)*) então a relação é aprendida pelo sistema apenas com a aprovação do usuário. Ex:

```
suj : (run , boy) ? (s/n):s
>> rela(run,suj - boy)
```

Esta relação em si é de pouca valia se a ela não for associada uma tradução. O usuário poderia associar traduções às cláusulas *rela* como por exemplo:

```
rela(run/apressar,suj - boy/garoto)
```

A síntese está dividida em duas fases. Na primeira são verificados as relações em inglês e suas respectivas traduções para o português, bem como é feita a expansão de *nomi*. Na segunda fase, é feita a concordância de gênero, pessoa e número. Ex:

```
1>
[suj * [[det(_OF94,the) - the],n(s,boy,0) - boy],vt((3 ,
s),run / pres) - runs]
2>
[det(m,s,o),n(m,s,menino),vt((3 , s),correr / pres)]
```

- Resultado

A sentença traduzida para o português é apresentada. O trabalho ainda não está totalmente desenvolvido. Não estamos lidando com acentuação ou a formatação da sentença em português.

Ex:

[[o, menino, corre]]

7.2. Segundo exemplo

Vamos apresentar o exemplo referente à tradução da palavra "work" com base em sua relação com os substantivos. Este exemplo baseia-se na seguinte estrutura hierárquica:

r (raiz da árvore)
animado
boy-menino
woman-mulher
inanimado
car-carro
machine-máquina

Do ponto de vista da implementação, esta estrutura é implementada através das cláusulas:

r/animado.
r/inanimado.
n(\$woman\$,f - \$mulher\$,animado).
n(\$boy\$,m - \$menino\$,animado).
n(\$car\$,m - \$carro\$,inanimado).
n(\$machine\$,m - \$maquina\$,inanimado).

----- Sentenca

the boy works

----- Morfologia

1 - d(_0D5C,the)
2 - n(s,boy,animado)
rela(work,suj - woman)
rela(work,suj - computer)
rela(work / funcionar,suj - computer / computador)

Observe a primeira relação traduzida, extraída da base de dados.

```
rela(work,suj - machine)
rela(work,suj - boy)
rela(work / trabalhar,suj - woman / mulher)
```

Observe a segunda relação traduzida.

```
3 - n(pl,work)
3 - vi((3 , s),work / pres)
```

----- Sintaxe

----- Arvore de Palavras

```
[
  [
    suj *
    [
      [
        det(_10C0,the) - the
      ]
      nomi(s,boy,animado,[],[]) - boy
    ]
    vi((3 , s),work / pres) - works
  ]
]
```

----- Sintese

```
1>
[suj * [[det(_10C0,the) - the],n(s,boy,animado) - boy +
menino],vi((3 , s),work / pres) - works + trabalhar]
```

Observe que neste caso, a tradução de boy e work foi feita neste passo a partir da generalização sobre as duas cláusulas provenientes da base de dados: *boy* assemelha-se mais a *woman* que a *computer*.

```
2>
[det(m,s,o),n(m,s,menino),vi((3 , s),trabalhar / pres)]
```

----- Resultado

```
[[o,menino,trabalha]]
```

7. Comparações com Projetos Simulares

Não encontramos referências específicas ao nosso mecanismo de generalizar as relações. As referências encontradas que mais se aproximam foram as que tratam da aquisição automática relações sintagmáticas.

- [Weischedel-93]: apresenta como diversos métodos estatísticos podem ser usadas em diversas fases do processamento de linguagem natural. Em particular, na fase da análise semântica é apresentado um mecanismo de extração das relações sintagmáticas a partir de um texto anotado. As relações extraídas são generalizadas segundo uma estrutura hierárquica. Existem diversas diferenças com o nosso trabalho: [Weischedel-93] não está preocupado em traduzir relações. O resultado da generalização de [Weischedel-93] é uma base de dados que contém diversas relações generalizadas, como, por exemplo (eat, FOOD) onde FOOD corresponde a uma classe de substantivos e não a apenas um substantivo. Este tipo de generalização pode levar a erros. Por exemplo: vamos supor que temos diversas relações-exemplos, tais como "stretch-arm", "stretch-bridge", etc. A super-classe que abrange *arm* e *bridge* ao mesmo tempo pode estar em um nível próximo da raiz da estrutura hierárquica (chamada de TUDO), ou seja, esta super-classe abrange praticamente todos os substantivos. Isto acarreta uma super-generalização inconveniente, levando a uma relação do tipo (stretch, TUDO). Esta relação em particular acaba sendo de pouca valia. A diferença com o nosso mecanismo de generalização está no momento em que esta generalização ocorre. O trabalho de [Weischedel-93] é bom para preparar a uma base de dados a priori, ou seja, antes que ela esteja sendo usada. Em nosso caso, a generalização é feita no momento da tradução e não a priori. Dessa forma, para a relação de entrada (stretch-arm) iremos procurar nos exemplos da base de dados a relação mais próxima, ou seja, (stretch-arm) e sua correspondente tradução. Isto evita possíveis erros de tradução devido à super-generalização com base em uma relação do tipo (stretch-TUDO). Devemos observar porém que [Weischedel-93] não está preocupado especificamente com o processo de tradução de relações sintagmáticas.

- [Hindle-93] se preocupa em buscar como as preposições são usadas de forma a distinguir onde o Sintagma Preposicional se encaixa em seqüências do tipo: [Verbo, Objeto Direto, Sintagma Preposicional]. Para tanto passa a levantar diversas relações sintagmáticas do tipo Verbo-Preposição ou Nome-Preposição. Este trabalho também não se volta à tradução de relações, e não se vale de uma

estrutura hierárquica para fazer a generalização sobre exemplos. A semelhança com o nosso trabalho está em fazer uso de um banco de relações para inferir o comportamento de verbos, preposições e nomes.

8. Conclusão

A construção de um programa tradutor que substitua o homem parece um sonho ainda distante, mas podemos verificar que, na busca deste sonho, vários passos já foram dados. Do ponto de vista de contribuições, procuramos apresentar as bases sobre as quais uma ferramenta de auxílio à tradução pode ser construída, não com o objetivo de substituir um tradutor humano, e sim com o de auxiliá-lo na execução do processo de tradução. A ferramenta aqui discutida se situa entre um simples dicionário eletrônico e um tradutor perfeito.

Embora a ferramenta, aqui apresentada, necessite de várias melhorias, como a formulação de mais regras gramaticais e aperfeiçoamento na base de dados, o que torna esta ferramenta peculiar é:

- a capacidade de aprendizado através da interação com o usuário.
- a escolha das palavras na tradução: ela pode ser feita através de exemplos ou da generalização de exemplos de relações entre palavras.
- a união de conceitos provenientes da gramática de ligações e da teoria X-barras em uma gramática DCG.
- a sua implementação na forma de um laboratório, o que lhe dá uma característica muito adequada para fins didáticos.

Bibliografia

- [Hindle-93] Hindle,D.;Rooth,M. "Structural Ambiguity and Lexical Relations". Computational Linguistics, Vol 19, No. 1, pgs 103 - 120, 1993.
- [Kinoshita-96] Kinoshita,J. "Aspectos de implementação de uma ferramenta de auxílio à tradução Inglês-Português" tese de doutorado a ser apresentada à Escola Politécnica da Universidade de São Paulo, 1996.
- [Lopes-75] Lopes,E. "Fundamentos da Linguística Contemporânea"; Editora Cultrix, São Paulo, 1975.
- [Lyons-79] Lyons, J. "Introdução à Linguística Teórica" Editora da Universidade de São Paulo, 1979. Título Original: "Introduction to Theoretical Linguistics" Cambridge University Press.
- [Raposo-92] Raposo, E.P. "Teoria da Gramática. A Faculdade da Linguagem" Editorial Caminho, SA, Lisboa -1992.
- [Sleator-91] Sleator,D.D.K.; Temperley,D. "Parsing English with a Link Grammar" Tese de doutorado. Canegie Mellon University. 1991
- [Weischedel-93] Weischedel,R. et al. "Coping with Ambiguity and Unknown Words through Probabilistic Models" Computational Linguistics, Vol 19, No. 2, pgs 360-382, 1993.

BOLETINS TÉCNICOS - TEXTOS PUBLICADOS

- BT/PCS/9301 - Interligação de Processadores através de Chaves Ômicron - GERALDO LINO DE CAMPOS, DEMI GETSCHKO
- BT/PCS/9302 - Implementação de Transparência em Sistema Distribuído - LUÍSA YUMIKO AKAO, JOÃO JOSÉ NETO
- BT/PCS/9303 - Desenvolvimento de Sistemas Especificados em SDL - SIDNEI H. TANO, SELMA S. S. MELNIKOFF
- BT/PCS/9304 - Um Modelo Formal para Sistemas Digitais à Nível de Transferência de Registradores - JOSÉ EDUARDO MOREIRA, WILSON VICENTE RUGGIERO
- BT/PCS/9305 - Uma Ferramenta para o Desenvolvimento de Protótipos de Programas Concorrentes - JORGE KINOSHITA, JOÃO JOSÉ NETO
- BT/PCS/9306 - Uma Ferramenta de Monitoração para um Núcleo de Resolução Distribuída de Problemas Orientado a Objetos - JAIME SIMÃO SICHMAN, ELERI CARDOSO
- BT/PCS/9307 - Uma Análise das Técnicas Reversíveis de Compressão de Dados - MÁRIO CESAR GOMES SEGURA, EDIT GRASSIANI LINO DE CAMPOS
- BT/PCS/9308 - Proposta de Rede Digital de Sistemas Integrados para Navio - CESAR DE ALVARENGA JACOBY, MOACYR MARTUCCI JR.
- BT/PCS/9309 - Sistemas UNIX para Tempo Real - PAULO CESAR CORIGLIANO, JOÃO JOSÉ NETO
- BT/PCS/9310 - Projeto de uma Unidade de Matching Store baseada em Memória Paginada para uma Máquina Fluxo de Dados Distribuído - EDUARDO MARQUES, CLAUDIO KIRNER
- BT/PCS/9401 - Implementação de Arquiteturas Abertas: Uma Aplicação na Automação da Manufatura - JORGE LUIS RISCO BECERRA, MOACYR MARTUCCI JR.
- BT/PCS/9402 - Modelamento Geométrico usando do Operadores Topológicos de Euler - GERALDO MACIEL DA FONSECA, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/9403 - Segmentação de Imagens aplicada a Reconhecimento Automático de Alvos - LEONCIO CLARO DE BARROS NETO, ANTONIO MARCOS DE AGUIRRA MASSOLA
- BT/PCS/9404 - Metodologia e Ambiente para Reutilização de Software Baseado em Composição - LEONARDO PUJATTI, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/9405 - Desenvolvimento de uma Solução para a Supervisão e Integração de Células de Manufatura Discreta - JOSÉ BENEDITO DE ALMEIDA, JOSÉ SIDNEI COLOMBO MARTINI
- BT/PCS/9406 - Método de Teste de Sincronização para Programas em ADA - EDUARDO T. MATSUDA, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/9407 - Um Compilador Paralelizante com Detecção de Paralelismo na Linguagem Intermediária - HSUEH TSUNG HSIANG, LÍRIA MATSUMOTO SAITO
- BT/PCS/9408 - Modelamento de Sistemas com Redes de Petri Interpretadas - CARLOS ALBERTO SANGIORGIO, WILSON V. RUGGIERO
- BT/PCS/9501 - Síntese de Voz com Qualidade - EVANDRO BACCI GOUVÊA, GERALDO LINO DE CAMPOS
- BT/PCS/9502 - Um Simulador de Arquiteturas de Computadores "A Computer Architecture Simulator" - CLAUDIO A. PRADO, WILSON V. RUGGIERO
- BT/PCS/9503 - Simulador para Avaliação da Confiabilidade de Sistemas Redundantes com Reparo - ANDRÉA LUCIA BRAGA, FRANCISCO JOSÉ DE OLIVEIRA DIAS
- BT/PCS/9504 - Projeto Conceitual e Projeto Básico do Nível de Coordenação de um Sistema Aberto de Automação, Utilizando Conceitos de Orientação a Objetos - NELSON TANOMARU, MOACYR MARTUCCI JUNIOR
- BT/PCS/9505 - Uma Experiência no Gerenciamento da Produção de Software - RICARDO LUIS DE AZEVEDO DA ROCHA, JOÃO JOSÉ NETO
- BT/PCS/9506 - Método OO - Método de Desenvolvimento de Sistemas Orientado a Objetos: Uma Abordagem Integrada à Análise Estruturada e Redes de Petri - KECHI HIRAMA, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/9601 - MOOPP: Uma Metodologia Orientada a Objetos para Desenvolvimento de Software para Processamento Paralelo - ELISA HATSUE MORIYA HUZITA, LÍRIA MATSUMOTO SATO
- BT/PCS/9602 - Estudo do Espalhamento Brillouin Estimulado em Fibras Ópticas Monomodo - LUIS MEREGE SANCHES, CHARLES ARTUR SANTOS DE OLIVEIRA

BT/PCS/9603 - Programação Paralela com Variáveis Compartilhadas para Sistemas Distribuídos - LUCIANA BEZERRA
ARANTES, LIRIA MATSUMOTO SATO

BT/PCS/9604 - Uma Metodologia de Projeto de Redes Locais - TEREZA CRISTINA MELO DE BRITO CARVALHO, WILSON
VICENTE RUGGIERO

BT/PCS/9605 - Desenvolvimento de Sistema para Conversão de Textos em Fonemas no Idioma Português - DIMAS
TREVIZAN CHBANE, GERALDO LINO DE CAMPOS

BT/PCS/9606 - Sincronização de Fluxos Multimídia em um Sistema de Videoconferência - EDUARDO S. C. TAKAHASHI,
STEFANIA STIUBIENER

BT/PCS/9607 - A importância da Completeza na Especificação de Sistemas de Segurança - JOÃO BATISTA CAMARGO
JÚNIOR, BENÍCIO JOSÉ DE SOUZA

BT/PCS/9608 - Uma Abordagem Paraconsistente Baseada em Lógica Evidencial para Tratar Exceções em Sistemas de
Frames com Múltipla Herança - BRÁULIO COELHO ÁVILA, MÁRCIO RILLO

BT/PCS/9609 - Implementação de Engenharia Simultânea - MARCIO MOREIRA DA SILVA, MOACYR MARTUCCI JÚNIOR

BT/PCS/9610 - Statecharts Adaptativos - Um Exemplo de Aplicação do STAD - JORGE RADY DE ALMEIDA JUNIOR, JOÃO
JOSÉ NETO

BT/PCS/9611 - Um Meta-Editor Dirigido por Sintaxe - MARGARETE KEIKO IWAI, JOÃO JOSÉ NETO

BT/PCS/9612 - Reutilização em Software Orientado a Objetos: Um Estudo Empírico para Analisar a Dificuldade de Localização
e Entendimento de Classes - SELMA SHIN SHIMIZU MELNIKOFF, PEDRO ALEXANDRE DE OLIVEIRA GIOVANI

BT/PCS/9613 - Representação de Estruturas de Conhecimento em Sistemas de Banco de Dados - JUDITH PAVÓN
MENDONZA, EDIT GRASSIANI LINO DE CAMPOS