Tratamento automático de termos compostos

Ulisses Pinto, José João Dias de Almeida
Departamento de Informática
Universidade do Minho
Campus de Gualtar, 4700 Braga
ulisses,jj@di.uminho.pt

Resumo

Em Linguagem Natural são frequentes as situações em que certas sequências de palavras tem um significado diferente daquele que seria inferível a partir dos significados das partes.

Estas situações correspondem a diferentes fenómenos linguísticos como idiomatismos, lexias complexas, locuções, etc. que por simplicidade serão o designados por **termos compostos**.

Para processamento de linguagem natural, à necessidade de pré-agrupamento das palavras em termos compostos para um tratamento correcto destes.

Neste artigo apresenta-se uma ferramenta PTC , para detecção e tratamento automático de termos compostos baseada num dicionário externo.

Na sua versão actual o PTC permite:

- controlar o tipo de flexões possíveis e a que palavras podem ser aplicadas
- devolver informação morfo-sintáctica associada ao termo composto. Essa informação resulta do dicionário e das flexões encontradas
- tratar as ambiguidades léxicas e de sobreposição de termos compostos
- percorrer textos e anotar os termos compostos
- tratar sobreposições de termos compostos

Uma descrição da estrutura interna de representação bem como a arquitectura geral da ferramenta será apresentada, juntamente com exemplos de utilização.

Palavras chave: termos compostos, PLN, morfologia, verbos operacionais

Índice

1 Introdução

1.1 Terminologia

Dada a sua complexidade e variedade, os conceitos e terminologia portuguesa associados a lexias complexas, idiomatismos, locuções etc, não conseguiu ainda consenso.

Neste documento usaremos **termos compostos** para designar genericamente as sequências de palavras que têm características não dedutíveis a partir das características das partes constituintes

$$TC \neq f(P_1, P_2, P_n)$$

Alguns exemplos de compostos

- Ideomatismos (Ex.: *pezinhos de lã*)
- Lexias complexas (Ex.: *máquina de escrever*)
- Locuções (Ex.: *para com, de modo que*)

1.2 Motivação

O processamento de linguagem natural necessita normalmente do pré-agrupamento das palavras em termos compostos. Cite-se por exemplo as áreas de:

tradução automática a generalidade dos termos compostos não pode ser traduzido "à letra". No processo de tradução tem de haver um pré-processamento que encontre os termos compostos e só então pode ser feita uma correcta produção de equivalentes.

> Exemplo:
> `short circuit -- *circuito pequeno`
> é uma tradução palavra-a-palavra incorrecta. A produção de equivalentes tem de ser orientada ao termo.

localização de informação (information retrieval): para aceder selectivamente a partes de uma grande colecção de documentos, é necessário normalizá-los e indexá-los. Não faz sentido que as palavras constituintes de um termo composto sejam agrupadas com a palavra isolada.

> Exemplo: a palavra "gato" no composto "gato pingado" não está relacionada com o mamífero gato, por isso quem pesquisar informação sobre gatos não tem interesse em encontrar a entrada relativa a "gato pingado"

Na pesquisa, pode haver utilização de termos compostos, havendo por isso a necessidade de existência de entradas indexadas dos mesmos.

> Exemplo: Para encontrar textos sobre "base de dados" interessa também "bases de dados", mas não interessa apenas "base" ou apenas "dados"

tratamento sintáctico de linguagem natural a não consideração dos termos compostos como locuções, leva a situações em que a análise sintáctica falha ou em que há necessidade de uma análise muito mais complexa.(Ex.:*... para com ...*)

tratamento semântico de linguagem natural No caso do tratamento semântico o caso é ainda mais grave, já que (por definição) o significado do composto não é dedutível a partir da semântica das palavras que o constituem, logo a única hipótese de tratamento semântico correcto desses casos é ter a indicação explícita da semântica do termo composto. (Ex.:*Apesar de tudo fomos embora*).

1.3 Questões básicas a considerar

O reconhecimento de um termo composto não se limita evidentemente a uma simples comparação de *strings*, já que as próprias palavras que compõem o termo composto podem ser flexionadas. Em alguns casos é importante:

- restringir o tipo de flexões que podem ser feitas a partir das palavras.
- dar a possibilidade de indicar *features* para o termo definido (de outro modo a sua utilidade em gramáticas seria muito limitada)
- permitir a herança de *features* de algumas das palavras que a constituem.

  Exemplo: a análise do termo composto "acertei em cheio" deve retornar informação sobre o tempo, o número e a pessoa, sendo esta informação herdada de "acertar" `[CAT=v,T=preper,N=s,p=1]`

Ambiguidade Há problemas no armazenamento dos termos compostos devido a ambiguidades léxicas dos termos constituintes. O reconhecimento levanta também problemas porque uma palavra pode ter mais que um radical diferente o que obriga a seguir simultaneamente vários caminhos diferentes na árvore de termos compostos.

Categorização e semântica Aos termos compostos são associadas categorias gramaticais e respectivas semânticas, não sendo, em grande parte dos casos possível atribuir uma categoria atómica como *adjectivo, nome comum*, etc., correspondendo-lhes por vezes categorias gramaticais de nível superior - *frase nominal, frase verbal, frase,* etc.

2 Formato de definição de termos compostos

2.1 Morfologias

A principal dificuldade na implementação dum sistema deste tipo é o tratamento de flexões. Por ex. "andar à nora" deve incluir "andei à nora", "andou à nora", etc. mas não "andei às noras". Isto significa que devemos dar a indicação de qual(ais) a(s) palavra(s) passíveis de serem flexionadas, o que pode ser expresso na seguinte notação:

`andar* à nora`

que tem o significado de

`[todas as flexões associadas a andar] à nora.`

Normalmente é útil associar ao composto as suas *features* que podem ser indicadas explicitamente ou herdadas de uma das palavras que a compõem. O operador "*" indica flexão universal com herança de *features*.

No caso de existirem duas palavras flexionáveis no mesmo termo composto (Ex.:*"acesso directo" no plural é "acessos directos"*), devem associar-se dois operadores de flexão. O operador "%" é usado para flexão universal sem herança e o operador "&" para herança sem flexão.

> Exemplo:
> `andar* depenado%G-N`
>
> Ambas as palavras são flexionáveis, a palavra "andar" é flexionável e é a base para extracção das *features* do termo composto. "Depenado" é flexionável em Género (G) e Número (N). Se pretendermos indicar que este termo composto é uma frase verbal (FV) optaríamos por:
> `andar* depenado%G-N/CAT=FV`
>
> A anotação /CAT=FV explicita *features*(neste caso a categoria) a adicionar (reescrever) ao termo composto.

Sendo assim podemos definir uma entrada no dicionário como:

```
entrada   = (palavra[sinal[nfeatures]])+ ['/' def_feat]
sinal     = '*' | '%' | '&'
nfeatures = feat ['-' feat]*
def_feat  = feat '=' valor (',' feat '=' valor)*
feat      = sequência de letras que são normalmente iniciais da
                feature que representam
```

2.2 Grafo como resultado da análise de termos compostos

Durante a análise a extracção de um termo pode conduzir a uma situação ambígua que iremos representar por um grafo acíclico (ver figura).

Para uma sequência palavras que formem um termo composto pode existir uma subsequência de palavras que também seja um termo composto. Pode ainda acontecer que as palavras devam ser interpretadas separadamente.

> Exemplo:
> Com o Dicionário:
> `a pé`
> `a pé firme`
> A representação de "foi a pé firme" é

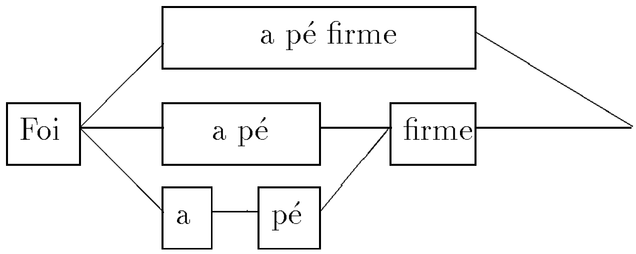


fig. Grafo de análise

2.3 Interface

2.3.1 Como programa etiquetador

Neste modo de utilização, o programa recebe texto de entrada gerando um novo texto em que cada uma das expressões encontradas é marcada com chavetas e respectiva categoria.

Exemplo:
`a Joana abriu os cordões à bolsa alegremente`
tem como resultado
`a Joana {abriu os cordões à bolsa}[CAT=v,T=pp,P=3,N=s] alegremente`

Exemplo: resultado da anotação dum pequeno texto (sem classificação):

O João {{a pé}firme}, foi com os {gatos pingados} {a altas horas} {abaixo de Braga} {abanar o capacete}, {abriu o apetite}, indo a uma {casa de tias}, já que {a carne é fraca}, entrou {à bruta} e {ficou entregue à bicharada}.

{Andou por maus caminhos}, {armou-se em parvo} e levou um {balde de água fria}. Para {salvar a pele} pôs-se a mexer, o {pobre diabo} que era {pessoa de bem} {perdeu a cabeça} e {passou um mau bocado}.

{Fez {o diabo a quatro}}, descobriram-se-lhe {os podres} e as coisas foram {de mal a pior}. {Deu {tudo por tudo}}, {correu riscos} desnecessários, ficou {para sempre} a {ovelha ranhosa}, {vendeu caro a vida} mas ele no fundo até {tinha pinta}.

2.3.2 Como biblioteca C:

A existência desta biblioteca permite utilizar o módulo de reconhecimento de compostos como componente de outros programas. A função básica desta biblioteca é `get_next_lexia`, que a partir de uma *string*, retorna um conjunto de pares posição, lexia:

$$get\_next\_lexia : posição \rightarrow (lexia \times posição)*$$

lexia – a próxima lexia (pode ser um termo simples ou composto)

posição – a posição imediatamente a seguir ao fim da lexia, (para facilitar a próxima chamada a `get_next_lexia`).

Sendo assim para uma dada *string* de input há na realidade uma lista de lexias possíveis e respectivas posições de continuação correspondendo às várias hipóteses de análise.

```
get_next_lexia(char *pos, comp_sol solutions[]);
```

A estrutura `comp_sol` é definida da seguinte forma:

```
typedef struct comp_sol {
   char *lex_end;
   char is_comp;   /*indica se se trata de um termo composto*/
   sol_type sol;
} comp_sol;


void arv_from_disk(char *filename);
```

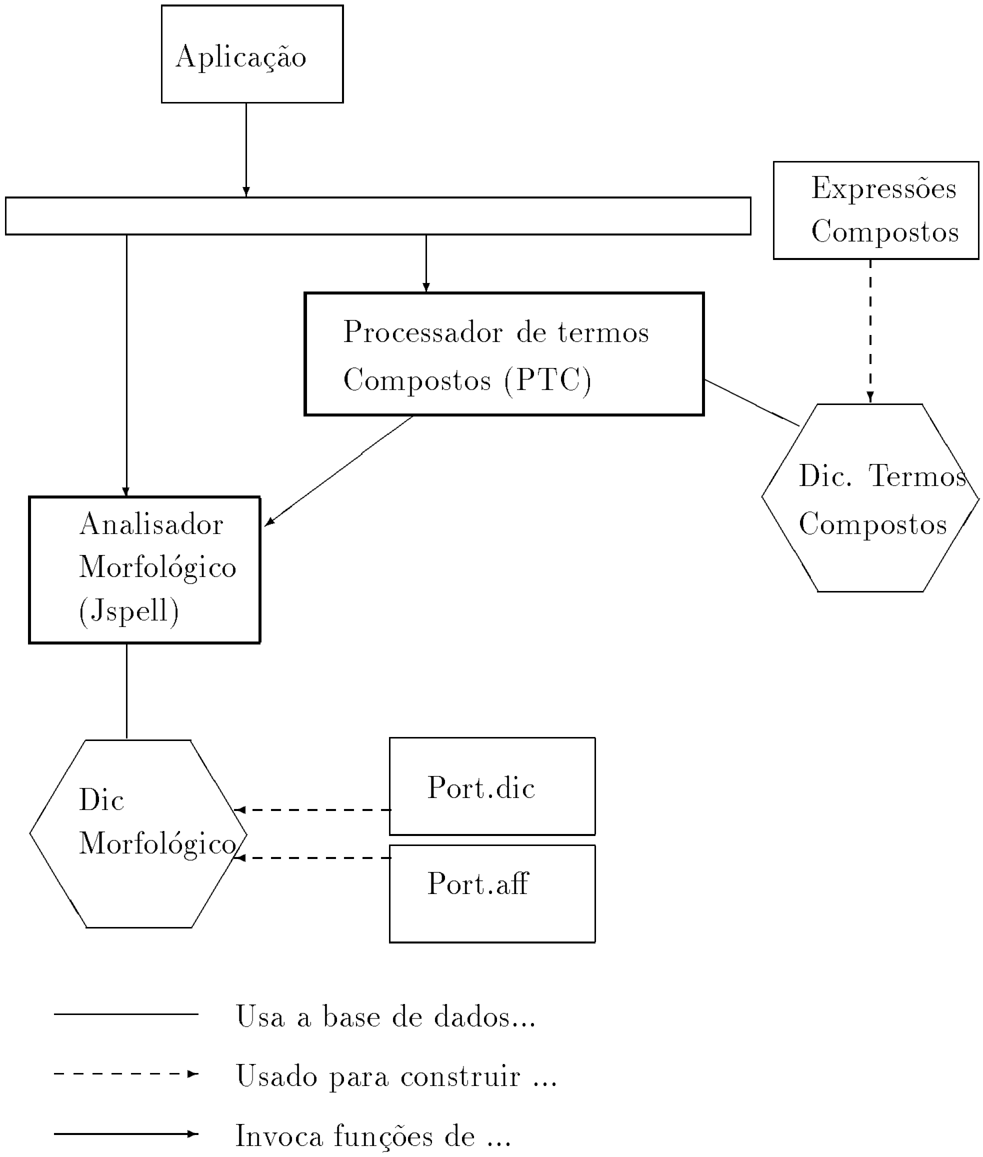
Esta função é invocada para inicializar o Ptc .

Programa exemplo de uso da biblioteca

```
#include "jslib.h"
#include "ptc.h"

main()
{
   ...
   comp_sol solutions[MAXPOSSIBLE];

   init_jspell("-d port -T lat -W 0 -g -a");
   arv_from_disk("expr.out");

   while (gets(X))
    { p_i = X;
      while (p = get_next_lexia(p_i, solutions))
         for(i = 0; solutions[i].sol[0] != '\0'; i++)
         {
            ...
         }
    }
}
```

2.4 Arquitectura do sistema Ptc e sua invocação

A utilização do sistema Ptc passa primeiro pela criação de um ficheiro intermédio que contém as expressões num formato mais manuseável que o ficheiro de texto inicial que continha a definição de compostos dada pelo utilizador. Mais tarde o ficheiro criado será usado no reconhecimento de termos compostos num texto.



Construção de dicionário de termos compostos

`build_ptc expr-por.txt port.hash expr.out`

`expr-por.txt` – lista de termos compostos
`port.hash` – dicionário morfológico (criado pelo jbuild/JSPELL)
`expr.out` – o dicionário de termos compostos criado.

A invocação do programa para etiquetação das expressões será:

`ptc port.hash expr.out`

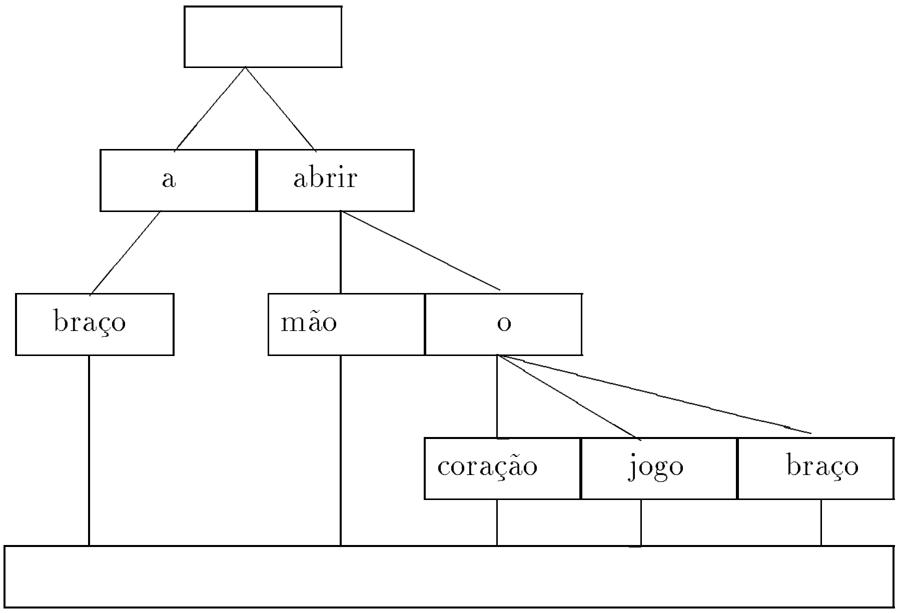
3 Estrutura interna de representação

As palavras são armazenadas sob a forma de uma árvore. Cada nó tem uma quantidade variável de células, (igual ao número de termos cujas palavras anteriores são comuns e diferem neste nível). Por razões de eficiência, optou-se por guardar nos nós da árvore a forma normalizada das palavras, sendo acrescentadas nas folhas da árvore as condições que restringem as flexões de acordo com o dicionário original.

Exemplo: os compostos

```
a braços
abrir mão
abrir o coração
abrir o jogo
abrir os braços
```

dão origem a uma árvore do tipo:



Na árvore aparece "braço" e não "braços" porque "braços" não é forma normalizada.

Nas folhas são indicadas as *features* do termo composto e de cada uma das palavras que o constituem, bem como a indicação da palavra a partir da qual será feita a herança de *features*.

Cada folha tem uma estrutura tipo:

```
n_def_feat [features]+ / feature_glob
```

`n_def_feat` - número de ordem da palavra cujas *features* serão herdadas
`features` - *features* que restringem as flexões possíveis da palavra
`feature_glob` - *features* do composto indicado explicitamente no dicionário inicial.

Exemplo: o composto `bom senso&` dá origem à descrição em *features*:
`2[CAT=adj][CAT=nc, G=m]/`
2 – indica que vai haver herança de *features* da 2ª palavra,

`[CAT=adj]` – são as *features* de "bom"
`[CAT=nc, G=m]` – são as *features* de "senso".

A seguir a "/" não aparece nada porque no dicionário também não há classificação global do composto.

4 Análise de resultados e trabalho Futuro

Na concepção do programa de reconhecimento de compostos, houve alguma preocupação de eficiência.

Num teste realizado com um dicionário de termos compostos de frases de tipo idiomáticos com cerca de 4500 frases aplicado a um sub-corpus com cerca de 1.5 milhões de palavras (retirado do corpus Natura-Público(Natura, 1995)) tendo sido encontrados 4716 termos idiomáticos[1] em 53 minutos.

O computador utilizado foi um 486-DX2, a 80Mhz com 16Mb de RAM, com o sistema operativo Linux.

Quanto a trabalho futuro são várias as linhas de acção previstas, correspondendo a fenómenos/características que gostaríamos de melhorar.

Restrições

Já foi referido a importância da existência de restrições para limitar certo tipo de flexões. Se assumirmos que o JSPELL já retorna a indicação que a palavra era reflexiva, podemos ter uma entrado do tipo:

```
fazer*R=1 engraçado
```

em que "R=1" indica que a reflexividade tem de existir. Outra hipótese possível consiste em indicar a condição no fim da frase:

```
fazer* engraçado/$1.R=1
```

esta segunda aproximação tem a vantagem de poder ser usada também para restringir concordâncias

```
ex. gato*N pingado%N/$1.N=$2.N
```

A condição final $1.N$ =2.N impede a inclusão de termos compostos como "gatos pingado"

Alternativos e opcionais

Existem muitas expressões que podem ser escritas de forma ligeiramente diferente, mas que significam o mesmo. Essa diferença pode resultar da possibilidade de remoção de uma palavra ou de substituição por outra(s).

Torna-se portanto importante a existência de um mecanismo de indicação de palavras alternativas e palavras opcionais.

Ex. de entrada com alternativas: " andar* (no | ao) engate"

Esta entrada incluirá tanto "andar no engate" como "andar ao engate".

[1] O número de lexias complexas, locuções existentes seria muito maior que o número de termos compostos de tipo idiomático. As frases idiomáticas foram escolhidas por serem estruturalmente mais complexas e assim constituírem um teste mais rico ao poder expressivo do sistema

Padrões na definição de termos compostos:

Muitas vezes um termo composto pode ter no seu interior um componente variável, em que é impraticável enunciar todas as hipóteses e que pode ser constituído por mais que uma palavra.

Esse componente pode ser por exemplo uma frase nominal.

Exemplo: o composto

```
fazer de alguém gato sapato
```

deve incluir os casos:

```
fazer do irmão gato sapato
fazer da irmã gato sapato
fez dele gato sapato
faz do meu rico tio gato sapato
```

A introdução desta potencialidade implica naturalmente uma interacção com o nível gramatical.

Verbos operativos e classes

Há certo tipo de verbos que aparecem com bastante frequência nos termos compostos e que funcionam um pouco como verbos auxiliares, na medida em que o seu conteúdo semântico no termo é virtualmente nulo.

Ex. no composto "chegar para as encomendas", o verbo "chegar" não tem aqui a sua semântica habitual.

Acontece também que muitos destes verbos podem substituir-se sem alterar o significado do termo.

Ex. "vir tarde e a más horas" é equivalente a "chegar tarde e a más horas"

em vez de ter duas entradas independentes ou apenas uma entrada do tipo

```
{vir | chegar} tarde e a más horas
```

poder-se-ia incluir estes dois verbos numa classe sendo usada essa classe querendo indicar que é aceitável qualquer um dos verbos que a constituem.

Estruturação

Em muitos casos a sobreposição pura e simples de *features* não é satisfatória, já que desta forma se pode perder informação importante. Para se lidar com um resultado único em que existam *features* com o mesmo nome é necessária a existência de algum nível de estruturação.

Adivinhar a categoria e *features* de um termo

Dada a enorme quantidade de expressões existentes, seria extremamente útil haver uma forma de determinar automaticamente a sua classificação gramatical. Isto teria de ser feito com base em pistas. Por exemplo um verbo seguido de um FN, pode ter como categoria um FV. Há no entanto muitos casos em que isto não é possível, tendo então o utilizador de indicar explicitamente a categoria.

Agradecimentos

Os autores agradecem a Álvaro Sanroman pelo esforço de clarificação das questões terminológicas envolvendo *colocações, frases feitas, fraseologia, idiomatismos, lexemas, lexias, locuções, modismos, solidariedades léxicas, etc.*

Bibliografia

Almeida, J.João and Ulisses Pinto. 1994. Manual de utilizador do JSpell. Manual, Universidade do Minho, Julho.

Baptista, Jorge M. 1994. Estabelecimento e formalização de classes de nomes compostos. Master's thesis, Fac. Letras, Universidade de Lisboa.

Natura, Projecto. 1995. Processamento de linguagem natural. http://www.di.uminho.pt/ jj/pln/pln.html.

Simões, Guilherme. 94. *Dicionário de expressões populares portuguesas*. Publicações D. Queixote.

Xavier, M. F. and M. H. Mateus. 1992. *Dicionário de Termos Linguísticos*, volume II. Lisboa: Cosmos.