TAGGERS EM GRAMÁTICAS DE ANÁLISE E DE SÍNTESE

SUSANA GENELIOUX
VITÓRIA MURUJO
(ILTEC)

Nesta comunicação defendemos que a integração de taggers nos sistemas de Processamento de Linguagem Natural (PLN) permite optimizar o seu funcionamento, quer do ponto de vista da adequação linguística, quer do ponto de vista do desempenho computacional.

A estratégia que defendemos é particularmente adequada no caso dos sistemas baseados em gramáticas, isto é, dos sistemas que permitem expressar directamente teorias linguísticas mediante determinados formalismos computacionais, como as chamadas gramáticas de unificação. Procuraremos demonstrar que, neste caso, a integração de taggers permite melhorar notavelmente a robustez dos sistemas, a velocidade de processamento e a eficiência das gramáticas.

Analisaremos em particular o caso de um sistema de PLN em desenvolvimento, cuja arquitectura comporta uma gramática de análise, uma gramática de síntese e um módulo de transferência cujo objectivo é a construção de um protótipo de sistema de tradução automática.

1. O que é um tagger?

Um dos mais recentes e bem sucedidos avanços na área da linguística computacional foi a criação de etiquetadores (taggers) com a capacidade de processar qualquer tipo de texto com elevada rapidez e fiabilidade. Um etiquetador, de um modo geral, é um programa que, dada uma sequência arbitrária de unidades lexicais, é capaz de lhe atribuir uma etiqueta consoante a sua categoria. Esta classificação pode ser feita segundo critérios morfológicos, sintácticos, semânticos, lexicais, etc.

Tem havido um aumento do interesse em taggers na última década. Um dos pontos de maior destaque neste tipo de aplicações é a obtenção de um alto nível de adequação na etiquetagem de textos, dado um conjunto de etiquetas (tagsets) previamente definido. A tarefa central de um tagger é escolher, de entre um conjunto de possíveis etiquetas, a etiqueta correcta para cada palavra no contexto em que está inserida. O número e a escolha das etiquetas pode variar não só consoante as línguas, mas também consoante as necessidades de uma dada aplicação.

Grande parte das aplicações em linguagem natural necessita de um mecanismo de classificação lexical que, perante uma palavra, obtenha várias informações a seu respeito como, por exemplo, a determinação da categoria gramatical.

Os taggers devem obedecer a vários requisitos fundamentais como independência, modularidade, flexibilidade e abrangência (Armstrong, S. et al., 1996).

Um etiquetador requer a separação entre os recursos linguísticos (léxicos e definição de tagsets) e os programas que os utilizam, ou seja, deve ser **independente** da língua.

Por vezes, alguns taggers são feitos tendo em conta apenas uma língua específica, o que dificulta a sua extensão e modificação, com o objectivo de serem utilizados por diferentes línguas. Para que tal não aconteça, o tagger deverá ter uma arquitectura **modular**. Deverá existir uma clara separação de subtarefas e uma boa definição de interfaces entre módulos, de forma a permitir a integração de recursos externos à aplicação. A modularidade também providencia uma boa base para os utilizadores explorarem individualmente as ferramentas que compõem a aplicação, e talvez até incluirem novos componentes.

Um tagger deverá ser **flexível**, ou seja, deverá permitir que os utilizadores possam adaptar os programas e os recursos de acordo com as suas necessidades.

Estas ferramentas deverão, acima de tudo, ser o mais **abrangentes** possível, ou seja, devem dar conta do maior número de unidades da forma mais adequada.

Existem vários modelos que têm sido utilizados para a elaboração de taggers, nomeadamente modelos estatísticos, modelos escondidos de Markov (Armstrong, S. et al., 1996) e modelos baseados em redes neuronais (Marques, N. & Lopes, G., 1995).

2. A importância dos taggers em aplicações para o processamento da linguagem natural

Os taggers são ferramentas de extrema importância no processamento da linguagem natural (PLN). Grande parte dos trabalhos feitos sobre a língua natural baseia-se em *corpora* falados ou escritos, de acordo com a aplicação, os quais,

antes de serem analisados, devem ser correctamente etiquetados consoante os objectivos que se pretende atingir com a sua utilização. Assim, os taggers servem de base a um vasto leque de aplicações de PLN como, por exemplo, sistemas de tradução automática, indexação automática, sistemas de recuperação de informação, bases de representação do conhecimento, sistemas de análise e/ou geração, correctores ortográficos, correctores sintácticos, sistemas de reconhecimento de fala, entre outros.

Como já foi referido na secção anterior, um tagger não é mais do que um etiquetador de *corpora*. Na maioria dos casos, os taggers são construídos tendo em vista a sua integração numa aplicação de PLN. O objectivo que se pretende que o tagger atinja determina certas características aquando da sua implementação. Por exemplo, a linguagem de programação utilizada, o conjunto de etiquetas que vai ser atribuído a cada unidade e a sua cobertura linguística, ou seja, o número e o tipo de unidades que vai tratar.

Tendo em conta estas características, podemos designar dois tipos de taggers diferentes: os **taggers gerais** e os **tagger específicos**. Os **taggers gerais** devem poder etiquetar todas as unidades do léxico de uma língua, independentemente do seu critério de classificação. Os **taggers específicos** destinam-se a etiquetar subconjuntos de léxico. Poderá ser útil, por exemplo, construir um tagger que etiquete apenas formas verbais, formas nominais, datas, etc.

3. A integração de taggers em sistemas de tradução automática

Apesar desta comunicação pretender evidenciar, de um modo geral, as vantagens da utilização de taggers em sistemas de PLN, a estratégia que defendemos insere-se no contexto de um projecto de investigação, mais concretamente do projecto GLEP (Gramática de Larga Escala do Português), projecto financiado pelo programa PRAXIS XXI (PCSH/C/CLC/123/96).

O GLEP é um projecto que tem como objectivo a implementação de um protótipo de um sistema de tradução automática de inglês para português de texto escrito, desenvolvido sobre a plataforma ALEP (Advanced Language Engineering Platform) e baseado em gramáticas de unificação. Esta plataforma foi criada para o desenvolvimento de sistemas de PLN que, entre outras aplicações, permite a construção de sistemas baseados em gramáticas.

Em sistemas de tradução automática como aquele que é desenvolvido no âmbito do projecto GLEP, cuja arquitectura é composta por uma gramática de análise, uma componente de transferência e uma gramática de síntese (para uma descrição dos vários tipos de sistemas de tradução automática, c.f. Eliseu et al. 1998, neste volume), a informação morfológica representa um papel fundamental no processo de análise linguística, como aliás acontece em qualquer sistema de PLN.

Existem várias formas de integrar taggers no sistema de tradução automática em causa, tirando partido de o ALEP ser uma plataforma aberta à integração de módulos externos. Devido à sua estrutura modular, o ALEP permite não só a integração de ferramentas/aplicações, mas também a substituição de módulos da sua arquitectura, sem que para isso seja necessário recorrer a qualquer compilação.

A tarefa concreta que se pretende que o tagger desempenhe determina a sua natureza e a sua função específica no contexto do sistema. Nos sistemas de tradução automática, a integração de taggers pode realizar-se a dois níveis: ao nível da análise e ao nível da síntese.

Ao nível da análise, um tagger pode, por exemplo, dar conta de todos os aspectos relacionados com a morfologia das unidades lexicais dos textos a traduzir. A atribuição de etiquetas a estas unidades, com informação sobre a sua classificação morfo-sintáctica, é um processo que serve de base a todas as fases de processamento linguístico que se seguem.

Ao nível da síntese, as etiquetas atribuídas por um tagger aos elementos linguísticos são importantes, por exemplo, para a componente de formação de palavras de qualquer sistema de tradução automática, funcionando como "reverse engineering" da componente de análise morfológica.

No caso do projecto GLEP, o tagger foi integrado na gramática de análise do sistema desenvolvido. Apesar de, como já foi referido, uma ferramenta deste tipo poder determinar a classe morfo-sintáctica de uma unidade lexical, o tratamento de questões relacionadas com a formação de palavras é feito através de regras específicas de "word-structure", regras estas que fazem parte da componente linguística do ALEP.

O tagger integrado no ALEP tem objectivos bastante específicos, que contrastam com as funções desempenhadas pela maioria dos taggers. A sua finalidade, no âmbito do GLEP, é dar conta de construções textuais cujo tratamento não pode ser feito pelas técnicas tradicionais de análise linguística, pois são expressões que não têm estrutura linguística interna. Podemos designar estas construções por "messy details". Exemplos típicos são números, códigos, siglas, datas, nomes próprios, títulos, legendas, ou outras (sequências de) formas "lexicais" que podem ocorrer de variadas formas, tornando impossível um tratamento adequado pelos meios tradicionais, ou seja, codificando cada entrada individualmente ou implementando regras sintácticas que contemplem estas unidades.

Construções deste tipo, em qualquer aplicação para o processamento da língua natural, devem ser processadas eficientemente. Por essa razão, uma das alternativas possíveis é a utilização de ferramentas implementadas em linguagens de programação que permitam o tratamento de expressões regulares. Desta forma, o sistema torna-se mais adequado do ponto de vista linguístico, ao dar

conta destas expressões cuja ocorrência é significativa em qualquer tipo de discurso.

O tratamento dos "messy details" da forma que foi descrito anteriormente tem grandes vantagens no funcionamento do sistema de tradução automática: diminuição significativa das entradas lexicais e das regras sintácticas, melhoramento do tempo de processamento, maior cobertura linguística e possibilidade de tratamento de texto real.

Todos os aspectos relacionados com o tagger integrado no sistema de tradução automática do GLEP e com o seu funcionamento são abordados na secção seguinte.

Um dos desenvolvimentos futuros deste projecto é a integração de um tagger no processo de síntese do sistema de tradução. Este tagger terá como função a geração de uma forma lexical, tendo como input os lemas e a classe morfológica pretendida. Desta forma, o tagger funcionará de forma inversa à forma como funciona na gramática de análise, melhorando a velocidade de processamento e a eficiência do sistema. Com a utilização de um tagger, não será necessário recorrer a regras de formação de palavras do próprio sistema, o que em ALEP provoca um aumento excessivo no tempo de processamento. Se conseguirmos que um módulo externo ao sistema consiga processar este tipo de informação de forma adequada e rápida, e se ao integrarmos esse módulo em ALEP essas características se mantiverem, conseguiremos optimizar o desempenho do sistema.

4. O tagger na gramática de análise

O tagger integrado na gramática de análise foi desenvolvido na Universidade de Essex, por um elemento da equipa inglesa do projecto LS-GRAM (LRE_61029). Deste projecto a nível europeu, resultaram implementações em ALEP de gramáticas de análise para várias línguas.

Descrição do tagger:

O tagger da equipa inglesa denomina-se *tagit_en*, por convenção. Foi determinado a nível dos participantes do projecto LS-GRAM que todos os taggers desenvolvidos no âmbito deste projecto tomassem este tipo de designação, alterando as duas últimas letras consoante a língua a que se destinassem. O *tagit_en* foi desenvolvido na linguagem de programação Perl, que permite o tratamento de expressões regulares.

Este tagger é interactivo e possui uma componente gráfica que facilita a sua utilização. Estas características trazem grandes vantagens, pois o tagger, ao interagir com o utilizador em tempo de processamento, permite desambiguar questões que o tagger sozinho não consegue. Uma outra vantagem do tagger interactivo é a de se um texto já tiver sido tratado pelo tagger, quando esse

mesmo texto "passar" pelo tagger uma segunda vez, o seu processamento vai ser muito mais rápido. Por exemplo os candidatos a nomes próprios são identificados como expressões regulares e vão ser aceites ou rejeitados em tempo de processamento, em interação com o utilizador.

Tarefas de integração:

A integração de uma ferramenta como o tagger referido anteriormente é feita em várias fases. Numa primeira fase, devem ser feitas todas as alterações necessárias ao próprio tagger de forma a adequá-lo às necessidades requeridas. De seguida, devem ser realizados testes ao tagger isolado, de forma a avaliar a sua funcionalidade fora da plataforma. Por fim, depois de executar todas as especificações necessárias no ALEP, o tagger fica integrado.

Na primeira fase, antes do tagger estar a funcionar isoladamente, ou seja, fora do ALEP, surge logo à partida uma dificuldade: interpretar o código que nos é apresentado. A análise do código em Perl vai ser indispensável para que se compreenda quais as expressões que já são tratadas e para adequar o tagger às necessidades do projecto. Após esta análise, devem ser feitas as especificações necessárias nos ficheiros fonte, para que possam ser reconhecidas todas as expressões desejadas. A fase de teste é indispensável para garantir que o tagger trate, da forma pretendida, todos os "messy details".

Passada a fase de testes do tagger isolado, foram feitas as alterações no código da própria plataforma de implementação. Em primeiro lugar, o ficheiro que executa o ALEP tem que conter, além de uma variável com o "caminho" para o tagger, todas as especificações que o próprio tagger exige. Em segundo lugar, têm de ser feitas alterações nas próprias operações do ALEP. O ALEP tem uma certa estrutura de operações no que diz respeito à gramática de análise, que vão desde o tratamento de texto, nomeadamente a sua transformação e etiquetagem, até à aplicação das regras sintácticas. A plataforma ALEP tem uma componente de Text Handling (TH), que permite um pré-processamento dos dados que servem como input ao sistema de tradução automática. Os dados passam primeiro por uma cadeia de processamento, que consiste numa etiquetagem, baseada no formato SGML, dos elementos do texto a traduzir. O texto é convertido para o formato EDIF (Eurotra Document Interchange Format), que se baseia no formato SGML. Seguem-se então outros processos de reconhecimento: reconhecimento de parágrafos, de frases e de palavras. O output destes processos consiste na etiquetagem dos elementos reconhecidos: etiqueta <P> para parágrafos, <S> para frases, <W> para palavras (nos casos de análise morfológica é utilizada a etiqueta <M> para os morfemas) e <PT> para os sinais de pontuação.

Exemplo:

A frase "Brian loves animals", após ter passado pela componente de TH, terá o seguinte formato (abreviado):

<P><S> <W>Brian</W><W>loves</W><W>animals</W><PT>.</PT></S></P>

Na realidade, existe mais informação sobre cada item no output das operações de TH. O formato real de "Brian" seria

<W TYPE="WORD" CASE="FIRST">Brian</W>

ou seja, a forma como esta componente trata os nomes próprios.

O módulo de TH da plataforma ALEP possibilita a integração de, por exemplo, programas de etiquetagem definidos pelo utilizador, como o *tagit_en*. Quando o tagger consegue identificar um padrão no input dado, a expressão é etiquetada com a etiqueta <USR>. Voltando ao exemplo anterior, a frase teria a seguinte sequência de processamento:

O input do módulo de TH seria um ficheiro de texto com a frase "Brian loves animals.". Este ficheiro vai ser transformado num outro em que a frase surge no formato EDIF.

<DOC LEVEL= "AS" LG= "EN" WP= "ASCII"><BDY LAY= "0"><P><S>Brian loves animals.</S> </P> </BDY> </DOC>

O ficheiro em formato EDIF será o input do *tagit_en*. As figuras que se seguem ilustram o funcionamento do *tagit_en*, bem como a sua interacção com o utilizador.

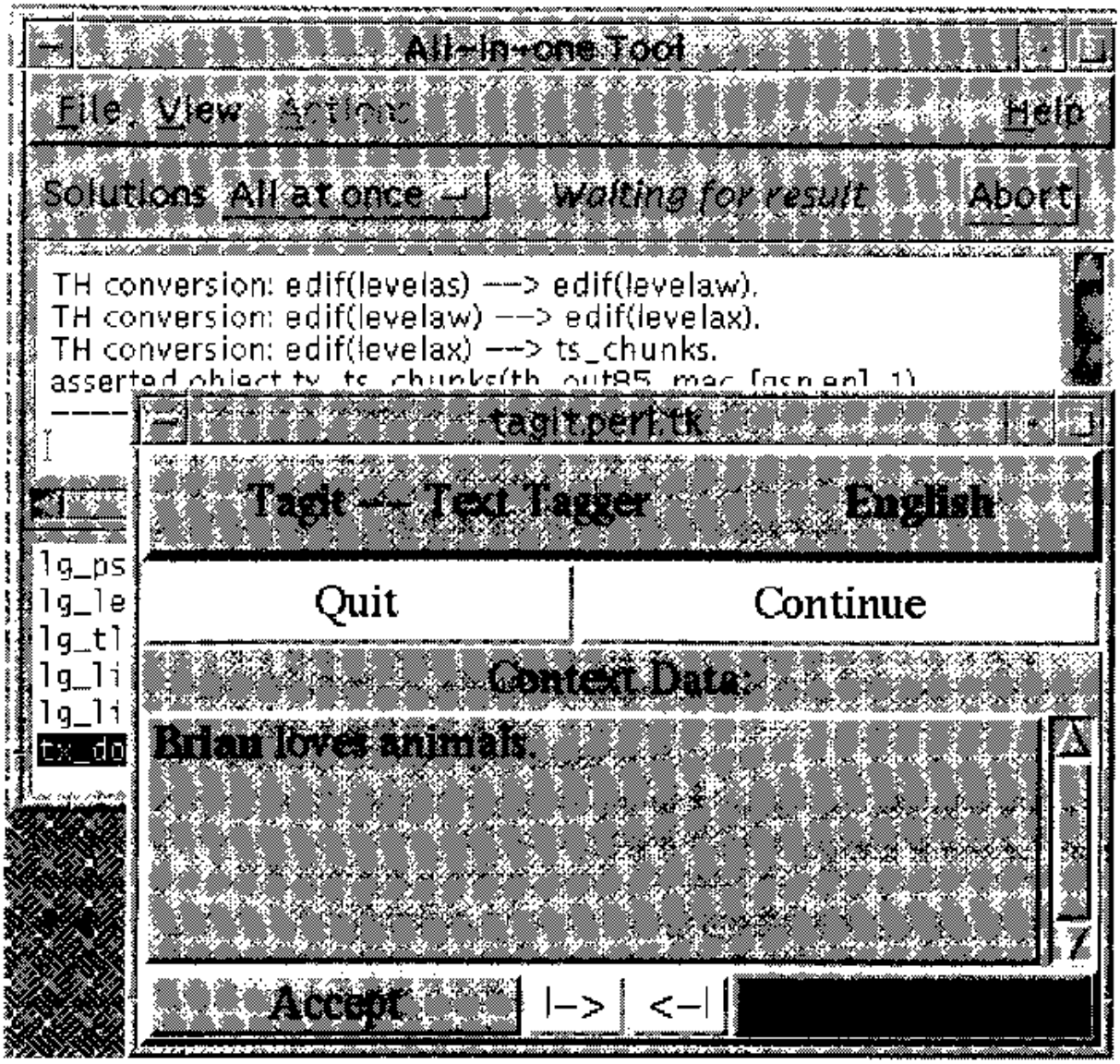

Fig. 1 - Primeira janela interactiva do *tagit_en* integrado numa operação do ALEP

Após a identificação do candidato a nome próprio, o tagger deixa a cargo do utilizador a classificação do elemento realçado.

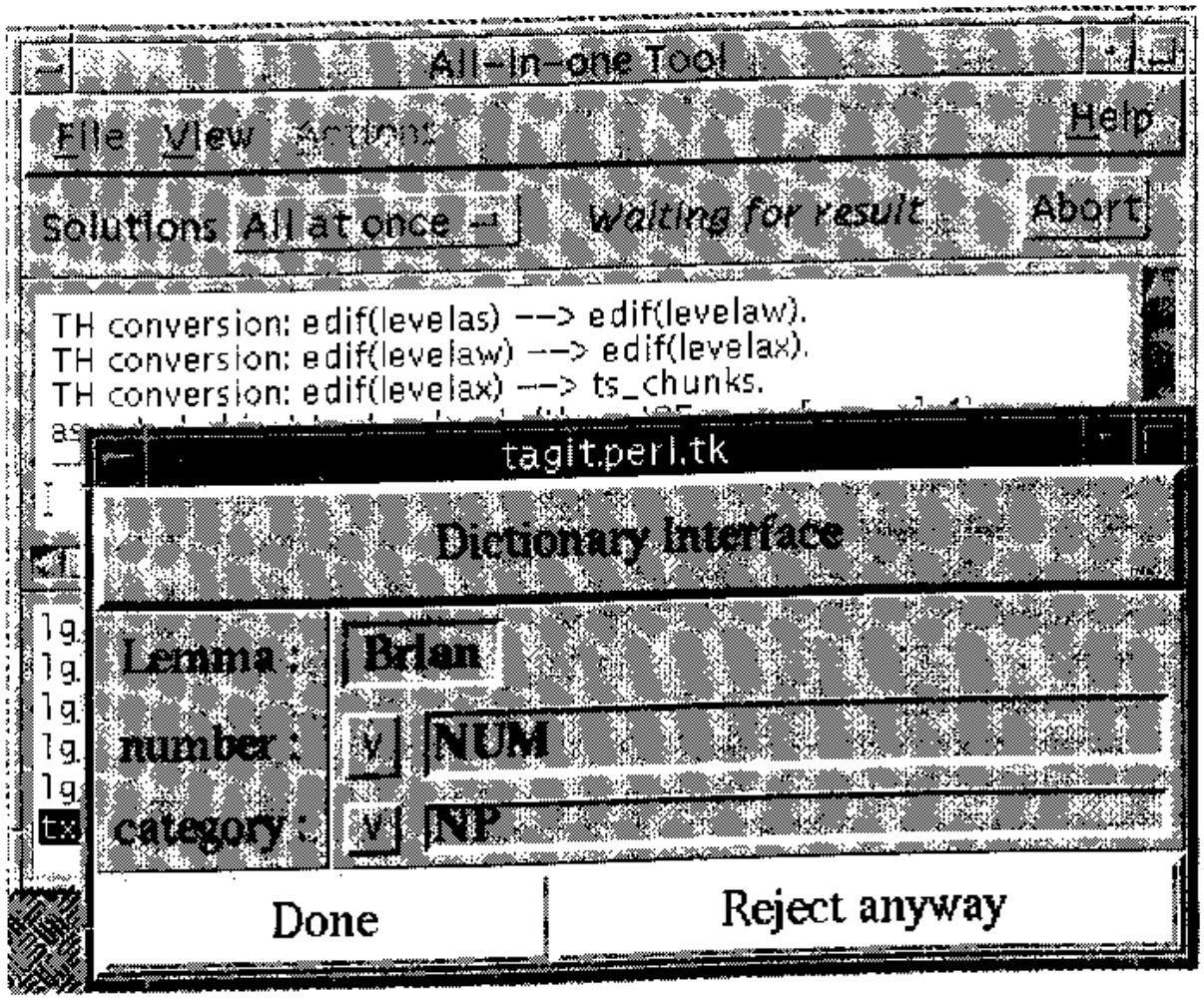

Fig 2 - Segunda janela interactiva do tagger integrada numa operação do ALEP

O output do *tagit_en* será um ficheiro em formato EDIF, com o nome próprio etiquetado correctamente.

<DOC LEVEL= "AW" LG= "EN" WP= "ASCII"><BDY LAY= "0"><P><S><**USR TYPE= "PNAME" ORIG= "Brian" VAL= "Brian" NUMBER= "NUM" CATEGORY= "NP"**>**Brian**</USR><W TYPE= "WORD" CASE= "NO">loves</W><W TYPE= "WORD" CASE= "NO">animals</W> <PT TYPE= "fs">.</PT></S></P></BDY></DOC>

Esta informação servirá como input da componente TH-LS (Text-Handling to Linguistic Structure) do sistema. Esta componente dispõe de regras, denominadas regras de *ts_ls* (text structure to linguistic structure), que transformam o output do TH em descrições linguísticas parciais. Estas regras possibilitam o fluxo de informação entre estes dois tipos de estruturas. É necessário elaborar regras de *ts_ls* que dêem conta das etiquetas atribuídas pelo tagger.

Exemplo de uma regra de *ts_ls*:

```
% Tag 'USR' lift rule (proper noun NPs)
ts_ls_rule(
 ld:{
  spec=en:{
    level_spec=level_spec:{
          lex=y,
          ws=y,
          ps=n}},
  sign=sign:{
     phon=phon:{
        string=[Word|Rest],
        rest=Rest},
     morph=morph:{
         cat=n,
         tag_name=TYPE},
     synsem=synsem:{
         locl=locl:{
            cat=cat:{
               head=noun:{
                  pred=minus,
                  mod=beardsley:{}},
               subj=[],
               comps=[],
               spr=[]},
            content=lq_cont:{
                rd_cont=r_npro:{
                    restr=[inst_psoa:{
                        rel=rel:{
                          rel_name=VAL}}]}}},
         nonlocl=nonlocl:{
             inher=nonlocl1:{
                slash_in=[],slash_out=[]}}}}},
'USR', ['TYPE'=TYPE='PNAME', 'ORIG'=VAL, 'NUMBER'=NUM,
'CATEGORY'='NP'], Word).
```

Após terem sido realizadas todas as alterações que foram descritas, é a altura de se testar o funcionamento do tagger integrado. Estes testes servem para verificar se o *tagit_en* etiqueta de forma correcta os "messy details", bem como

para verificar se toda a sequência de operações e de aplicação de regras não foi afectada pela utilização de uma ferramenta externa ao ALEP.

Os testes são também importantes para avaliar os tempos de processamento, de forma a determinar os índices de eficiência da ferramenta integrada.

5. Trabalho futuro

A experiência adquirida na integração de um tagger na gramática de análise vai permitir que, num futuro próximo, se integre um tagger na gramática de síntese do protótipo do sistema de tradução automática desenvolvido no âmbito do projecto GLEP. Este tagger terá a função de gerar, a partir de um lema e de um conjunto de traços, a forma lexical pretendida. Esta integração vai permitir eliminar da gramática regras de *word structure,* cuja utilização dificulta a boa performance do protótipo, diminuindo significativamente os tempos de processamento.

Referências Bibliográficas

Atwell, E., J. Hughes, Clive Souter (1994), AMALGAM: Automatic Mapping Among Lexico-Grammatical Annotation Models, *Proceedings of ACL workshop on The Balancing Act: Combining Symbolic and Statistical Approaches to Language,* New Mexico State University, Las Cruces, New Mexico, USA [Online], disponível: http://www.scs.leeds.ac.uk/nlp/papers.

Almeida, J. e U. Pinto (1994), *Jspell - Um módulo para análise léxica genérica de linguagem natural,* Universidade do Minho, [Online], disponível: http://www.di.uminho.pt/~jj/pln/pln.html.

Armstrong, S., P. Bouillon, G. Robert, *Tagger Overview.* ISSCO, University of Genebra, _Online_, disponível: http://issco-www.unige.ch/staff/robert/tatoo/tagger.html.

Chanod, J. and P. Tapanainen (1995), *Creating a tagset, and guesser for a French tagger.* Rank Xerox Research Centre, [Online], disponível: http://www.rxrc.xerox.com/grenoble/mltt/fsNLP/tagger.html.

Declerck, T and H. Maas (1997), *The Integration of a Part-of-Speech Tagger into ALEP Platform.* in Proceedings of the 3rd ALEP User Group Workshop, Saarbrücken, Germany.

Eliseu, A., A. Cardoso, C. Magro, E. Gonçalinho (1998), Precisão e cobertura: dois requisitos em conflito na constituição de uma gramática para um sistema de tradução, *Actas do XIV Encontro Nacional da Associação Portuguesa de Linguística,* Lisboa.

Fouvry F., A. Bredenkamp, T. Declerck, B. Music (1996), *Efficient Integrated Tagging of Word Constructs.* University of Essex, Essex.

Fouvry, F. and A. Bredenkamp (s.d.), *Partial parsing in ALEP.* University of Essex, Essex.

Hagége, C., A. Meireles, B. Trindade, C. Diogo, F. Leite (1997), *Analisador Morfo-Sintáctico.*, Relatório Técnico, ILTEC.

Marques, N. e G. Lopes (1996), Using Neural Nets for Portuguese Part-of-Speech Tagging, *Proceedings of the Fifth International Conference on The Cognitive Science of Natural Language*, [Online], disponível: http://www-ia.di.fct.unl.pt/~nmm/artigos/csnlp96.ps.gz.

Marques, N., *Etiquetagem Morfo-Sintáctica em Português usando Redes Neuronais*,Relatório Técnico, Universidade Nova de Lisboa, [Online], disponível: http://www-ia.di.fct.unl.pt/~nmm/artigos/relnn.ps.gz.

Reis, R.(1996), *Etiquetador de Português*. ,Relatório Técnico, Universidade do Minho, [Online], disponível: http://www.di.uminho.pt/~jj/pln/pln.html.

Simpkins, N.K.(s.d.), *ALEP - An open architecture for language engineering*, Cray Systems, [Online], disponível: http://www.cray-systems.lu/alep/doc/OpenAlep.html.