Do processamento morfológico à análise sintáctica de *corpora* multilíngue

Marcos Garcia, Pablo Gamallo

Universidade de Santiago de Compostela

Resumo: O presente trabalho tem como objectivo apresentar dous recursos disponibilizados, com licença GPL, pela linha de investigação ProLNat. O primeiro deles é o conjunto de ficheiros de treino de diversos módulos de análise morfossintáctica para Português Europeu e Galego de FreeLing. Neste momento, o sistema atinge valores de precisão na anotação morfossintáctica de mais de 96% e de 97%, respectivamente, para as duas variedades referidas. A segunda das ferramentas é uma *suite* de análise sintáctica com base nos princípios da gramática de dependências. O sistema compõe-se de um formalismo de geração deste tipo de gramáticas e de um compilador que cria parsers a partir das próprias gramáticas. Actualmente, são disponibilizados analisadores para cinco línguas: Inglês, Francês, Galego, Português e Castelhano. A análise é robusta e permite processar (parcialmente) grandes quantidades de *corpora*. Adicionalmente, será apresentado um modelo de correcção de PoS-tagging com base nas duas ferramentas referidas. O método consiste na aplicação, sobre a saída do PoS-tagger, de regras de pós-processamento morfossintáctico motivadas linguisticamente.

1 Introdução

O processamento da linguagem natural engloba um vasto conjunto de tarefas que vão desde tratamentos superficiais do texto (tokenização, segmentação de orações, etc.) até análises profundas do ponto de vista sintáctico e semântico. Tarefas como a extracção e a recuperação de informação, a tradução automática, etc., podem precisar deste tipo de tratamentos, que incluam análises morfossintácticas precisas (categoria, lema, género, número, etc.) bem como parsers robustos que permitam analisar grandes quantidades de *corpora*.

O presente trabalho tem como objectivo principal apresentar dous recursos desenvolvidos pela linha de investigação ProLNat (do Grupo Gramática do Espanhol da USC), que permitem (entre outras tarefas) analisar morfossintacticamente grandes quantidades de texto, e gerar parsers robustos, através de gramáticas de dependências, capazes de extrair informação sintáctico-semântica. Ambos os sistemas são disponibilizados baixo licença GPL.

A primeira das ferramentas é uma adaptação de diversos módulos de análise morfossintáctica de FreeLing (Carreras *et al*., 2004) (desenvolvido pelo Grupo TALP, da Universitat Politècnica de Catalunya) para Português Europeu (PE) e Galego (GA). Actualmente, são disponibilizadas ferramentas de tokenização, de segmentação de orações, de lematização (com base em léxicos) e de PoS-tagging. Adicionalmente, estão a ser desenvolvidos os módulos de reconhecimento de numerais, datas e quantidades, bem como identificadores de expressões multipalavra. Diversas avaliações realizadas aos módulos adaptados do FreeLing indicaram que a ferramenta tem desempenhos próximos do estado-da-arte, variando entre 94% e 98% de precisão em PoS-tagging, em função dos *corpora* de treino e de teste.

O segundo dos recursos é DepPattern (Gamallo e González, 2009; 2010), uma *suite* de

análise sintáctica multilíngue com base na gramática de dependências (Nivre, 2005, por exemplo). Esta ferramenta compõe-se de um formalismo para escrever gramáticas, bem como de um compilador que gera, com base nas próprias gramáticas, parsers de análise sintáctica robusta. DepPattern recebe como *input* a saída do analisador morfossintáctico (FreeLing ou TreeTagger (Schmid, 1994), a escolher pelo utilizador), pelo que as línguas analisáveis variam em função do PoS-tagger: Actualmente, DepPattern disponibiliza analisadores para cinco línguas: Inglês, Francês, Galego, Português e Castelhano.

Para além da análise sintáctica, DepPattern possibilita a utilização de informação morfossintáctica e léxica para corrigir a análise do PoS-tagger, o que permite melhorar a precisão do etiquetador (Garcia e Gamallo, 2010) e ampliar a cobertura das regras gramaticais.

O resto do artigo organiza-se da seguinte maneira: a Secção 2 faz uma apresentação da adaptação de FreeLing para Português Europeu e Galego; a Secção 3 mostra as principais características de DepPattern; finalmente, são destacadas as principais conclusões deste artigo.

2 Análise Morfossintáctica

A análise morfossintáctica é uma etapa crucial para o processamento linguístico em níveis superiores, tais como a análise sintáctica ou a síntese de voz, por exemplo. As primeiras tarefas deste processo têm a ver com a tokenização (divisão do texto em palavras), a segmentação de orações, a lematização ou a etiquetação morfossintáctica.

FreeLing permite realizar estas e outras tarefas sobre *corpora* de diferentes línguas, e possibilita também, de maneira relativamente simples, a sua adaptação para outros idiomas. Nesta secção apresentaremos os aspectos mais importantes do desenvolvimento de FreeLing para Português Europeu e Galego. Para a primeira das variedades, a versão aqui apresentada é a primeira disponível, enquanto para Galego o trabalho é realizado a partir de versões anteriores, realizadas pelo Seminario de Lingüística Informática da Universidade de Vigo.

2.1 Módulos Adaptados

A primeira das tarefas que FreeLing realiza é a tokenização, processo pelo qual um texto plano se converte num vector de palavras. O principal problema a ter em conta nesta etapa tem a ver com os tokens que têm ambiguidade entre contracção e palavra simples (por exemplo *nos*, que pode ser (i) contracção da preposição *em* com o artigo *os* ou (ii) um pronome pessoal). O sistema não possui informação morfossintáctica, pelo que nesta altura não consegue saber se estas formas devem ser divididas ou não. Assim sendo, neste primeiro processo, o tokenizador não vai separar as contracções, mantendo portanto a ambiguidade.

O seguinte módulo de FreeLing é o segmentador de orações, que devolve uma nova oração cada vez que detecta uma fronteira ortográfica. Para evitar que esta ferramenta receba casos ambíguos (o ponto final de uma abreviatura pode ser interpretado como uma fronteira), o

processo anterior (o tokenizador) desambigua estes casos graças a uma lista de abreviaturas.

Solucionadas estas ambiguidades, a configuração do segmentador de orações consiste na selecção das marcas ortográficas que indicam fronteira de oração; assim, salvo pequenas diferenças (como a utilização dos símbolos iniciais de interrogação e exclamação —não utilizados em PE, mas opcionais em GA), a configuração é similar para as duas variedades.

As seguintes tarefas realizadas por FreeLing compõem um meta-módulo de análise morfológica, que contém ferramentas de identificação de numerais, datas e quantidades, de reconhecimento de expressões multipalavra, de lematização, ou de desambiguação morfológica.

Até ao momento, o trabalho realizado centrou-se na adaptação dos sub-módulos de pesquisa em dicionário e de lematização verbal e nominal (com base em regras de afixação). O primeiro deles procura num léxico todas as possibilidades de análise de cada um dos tokens encontrados no *input*, enquanto a afixação permite tratar formas que não estão no dicionário (verbos com pronomes clíticos, sufixos diminutivos e aumentativos, etc.).

O dicionário de Português Europeu contém mais de 1.257.000 formas, enquanto o de Galego supera as 577.000.[1] Uma vez que o sistema não integra um lematizador próprio (unicamente para casos de prefixação e sufixação), a qualidade de lematização dependerá do tamanho dos léxicos. Neste sentido, foram realizadas avaliações da lematização nas duas variedades adaptadas, com resultados de 98,583% (PE) e de 99,41% (GA), sobre *corpora* de 50.000 e 6.200 tokens, respectivamente.

Com base na aplicação destes sub-módulos (pesquisa em dicionário e análise de afixos), o sistema permite tratar a ambiguidade que existe nas formas que podem ser analisadas como contracções ou como tokens simples.[2] Este tipo de entradas pode ser analisada através do dicionário (incluindo a contracção como mais uma forma de análise das formas) ou da afixação (criando regras de lematização que interpretem as contracções como elementos sufixados). As duas análises permitem evitar a circularidade provocada pela interacção do tokenizador com a análise morfossintáctica, cujas decisões são interdependentes. Assim, todas as possibilidades propostas pelo dicionário e pelas regras de afixação serão avaliadas pelo etiquetador morfossintáctico, que poderá dividir as contracções na sua saída.

Antes da aplicação do módulo de etiquetação morfossintáctica, FreeLing utiliza um desambiguador morfológico, com base no treino sobre *corpus*, que assigna uma probabilidade para cada um possíveis tags de cada token, e tenta saber quais as etiquetadas possíveis dos tokens desconhecidos.

O último módulo de processamento morfossintáctico é o PoS-tagger, que decide a etiqueta a atribuir a cada token. FreeLing inclui dous métodos: o clássico HMM (Brants, 2000) e um modelo híbrido, que combina informação estatística com restrições escritas manualmente

1 O primeiro foi extraído do LABEL-LEX (SW) (Eleuterio *et al*., 2003), enquanto o segundo é uma ampliação do publicado em anteriores versões de FreeLing pelo Seminario de Lingüística Informática da Universidade de Vigo.

2 As contracções não ambíguas (por exemplo, *do*) estão incluídas no dicionário, pelo que são divididas no *output* deste módulo.

(Padró, 1998). Nas adaptações aqui apresentadas foi utilizado o primeiro dos métodos, puramente estatístico, treinado sobre *corpora* etiquetado manualmente.

Para Português Europeu, o *corpus* utilizado foi uma adaptação do Bosque 8.0,[3] que contém uns 138.000 tokens revisados manualmente por linguistas. A versão de Galego foi treinada com um *corpus* do projecto Gari-Coter (Barcala *et al*., 2007), de 237.000 tokens extraídos de textos jornalísticos.

Para além do *corpus*, um outro factor com muita relevância no desempenho da anotação morfossintáctica tem a ver com o tagset utilizado. Uma vez que um dos objectivos do treino de FreeLing foi o seu uso como base de um analisador sintáctico, decidiu-se utilizar um tagset com informação morfossintáctica detalhada (categoria morfossintáctica; tempo, modo e pessoa dos verbos; género, número e grau de nomes e adjectivos, etc.). O tagset tem 255 e 277 etiquetas diferentes para PE e GA, respectivamente, e baseia-se nas recomendações do Grupo EAGLES (Leach e Wilson, 1996) bem como em tagsets de outras línguas incluídas em FreeLing.

2.2 Avaliação do PoS-tagger

Para conhecer o desempenho dos módulos PoS-tagging treinados, foram realizadas diversas avaliações com métodos diferentes. De modo geral, a avaliação deste processo realiza-se dividindo o número de tokens etiquetados correctamente entre o número total de tokens do *corpus*. Como foi dito, a informação dos tags utilizados é muito pormenorizada, o que implica uma precisão menor na etiquetação. Uma vez que o PoS-tagger não é apenas utilizado como *input* do analisador sintáctico, mas também para outras objectivos que não precisam de informação tão detalhada, foram realizadas avaliações com um tagset mais simples (*SingleTags)*. Este contém unicamente os dous primeiros elementos do tag completo (categoria morfossintáctica e tipo, salvo para os verbos —cuja etiqueta contém o modo), e compõe-se de 28 tags diferentes, para além dos símbolos de pontuação.

Etapas do processamento anteriores à etiquetação podem provocar desalinhamento entre o texto anotado automaticamente, e o *corpus* de teste. Assim, expressões multipalavra como os nomes próprios de mais de um elemento, podem ser incorrectamente tokenizados, pelo que o método de avaliação pode realizar-se de várias maneiras. Durante os testes realizados aos etiquetadores morfossintácticos, utilizaram-se os seguintes métodos:

- *NoTok*: Este método considera que os erros de tokenização não devem ser avaliados de maneira conjunta com o PoS-tagger. Assim, se são detectados erros de split (*Presidente_Mário_Soares NP* vs *Presidente NP / Mário NP / Soares NP*), unicamente é avaliado o tag do primeiro token (pelo que seria contabilizado um acerto.
- *Tok*: O método *Tok* considera também os erros de tokenização, pelo que se houver diferenças entre os dous texto, são contabilizados todos os erros (no caso anterior, três).

3 Bosque. Uma floresta integralmente revista por linguistas: http://www.linguateca.pt/Floresta/corpus.html#bosque.

Note-se que, no caso de que a tokenização e a assinação do tag em palavras com mais de um token sejam correctas, unicamente é atribuído um acerto.

- *NoLoc*: Esta avaliação só tem em conta os tokens correctamente alinhados, ignorando os restantes. No exemplo anterior, não se contabilizariam nem erros nem acertos.
- *OnlyTag*: Este método, como o anterior, avalia unicamente os tokens alinhados correctamente. À diferença de *NoLoc*, no *corpus* de teste utilizado foram divididas as formas de mais de um token (locuções, nomes próprios de mais de um token, etc.). Na avaliação das expressões multipalavra, portanto, são contabilizados os erros e acertos de etiquetação de cada um dos tokens que a formam.

Na Tabela 1 podemos ver os resultados das avaliações realizadas com o PoS-tagger de Português Europeu; os valores correspondem-se com a média de cinco execuções sobre extractos aleatórios de 10.000 tokens, com o sistema treinado nos restantes 130.000. O método *OnlyTag*, porém, foi treinado sobre 90.000 tokens, e testado com o um *corpus* de 50.000.

Avaliação	Tag Completo	SingleTags
NoTok	94,788%	96,012%
Tok	94,470%	95,728%
NoLoc	95,044%	96,263%
OnlyTag	94,324%	95,537%

Tabela 1: Avaliação PoS-tagger PE

A avaliação do PoS-tagger para Galego (cujos resultados estão na Tabela 2) foi realizada sobre um *corpus* de textos jornalísticos de 6.200 tokens, etiquetado manualmente. O treino foi realizado com o *corpus* completo, de quase 238.000 tokens.

Os resultados dos etiquetadores treinados encontram-se próximos do estado-da-arte, situado sobre o 97%, em função dos *corpora* de treino e teste, do tagset, ou da língua analisada (Megyesi, 2001; Branco e Silva, 2004).

Avaliação	Tag Completo	SingleTags
NoTok	97,695%	98,037%
Tok	97,191%	97,562%
NoLoc	97,724%	98,067%
OnlyTag	97,503%	97,914%

Tabela 2: Avaliação PoS-tagger GA

FreeLing contém mais ferramentas de análise que também podem ser adaptadas para outras línguas; neste momento, estão em desenvolvimento módulos como os identificadores de numerais, quantidades e datas ou de expressões multipalavra para PE e GA.

3 DepPattern: Análise Sintáctica em Gramática de Dependências

A gramática de dependências é um formalismo cuja utilização tem vindo a aumentar em diversas tarefas do processamento da linguagem natural. É, portanto, cada vez mais frequente

encontrar analisadores de dependências baseados em regras para diversas línguas (Atserias *et al*., 2005; Bick, 2006).

O formalismo que utiliza DepPattern tem como base os próprios princípios da gramática de dependências, mas incorpora ao mesmo tempo noções da linguística de *corpus* de Sinclair (Sinclair, 2001) e da gramática de padrões (Hunston e Francis, 1999).

Do trabalho de Sinclair, DepPattern incorpora a possibilidade de análise das expressões semi-fixas (aquelas cujo significado não é composicional, como *ter [algo] em conta*), permitindo a criação de regras gramaticais especializadas na identificação de unidades léxicas descontínuas e variáveis sintacticamente. Assim, para além das regras básicas de escolha livre (*open choices*), com base nos princípios regulares da composicionalidade semântica, DepPattern permite analisar as expressões semi-fixas como uma única unidade lexical.

A *Pattern Grammar* tem como base a ideia de "padrão", uma organização léxico-morfo-sintáctica de uma palavra, da qual selecciona aspectos semânticos. Os padrões são estruturas sintácticas de superfície, que permitem descrever a gramática de qualquer língua. Exemplos de padrões podem ser **ADJ de-inf**: *fácil de fazer* ou **V que-conj**: *Disse que cantasses.* DepPattern baseia-se na gramática de padrões porque fundamenta a sua análise na utilização de cadeias de PoS-tags acrescentadas com informação morfológica e léxica. Esta análise permite identificar e gerar as relações de dependência inerentes às estruturas das cadeias de elementos.

3.1 O Formalismo

As gramáticas escritas com DepPattern são um conjunto de regras que têm como objectivo estabelecer relações núcleo-dependente com base em padrões de etiquetas morfossintácticas, bem como de informação morfológica e léxica. As regras contêm, para além do padrão de etiquetas, o nome da relação de dependência:

Exemplo de Regra Simples
SpecL: DET NOUN
%

Este exemplo é uma regra que estabelece uma relação de dependência (*SpecL*) entre um determinante e um nome. Os nomes das dependências e das etiquetas são especificados em ficheiros individuais, pelo que podem ser modificados por cada utilizador.

DepPattern tem dous grandes tipo de dependências: *HeadDep* e *DepHead*, em função da posição do núcleo e do dependente. No exemplo anterior, e sendo o nome o núcleo da dependência, o tipo de relação será definida como *SpecL DepHead*.

Para além destas regras básicas, DepPattern permite acrescentar a informação das cadeias, com informação morfológica, léxica, ou com operadores de concordância, recursividade, etc.

Exemplo de Regra com Operadores
SpecR: NOUN ADJ<type:Q>
Agreement: gender, number
%

Neste exemplo, o tipo de dependência indica que o elemento à esquerda é o núcleo, e que um adjectivo qualificativo (type:Q) será o dependente. Além disso, o operador *Agreement* implica que os dous elementos da relação tenham concordância em género e em número.

DepPattern aplica as regras sequencialmente, e funciona com base no princípio de unicidade (*uniqueness principle*). Este princípio estabelece que cada palavra tem um só núcleo, isto é, funciona uma única vez como dependente. Assim, a aplicação de uma regra provoca o desaparecimento do dependente do *input* da seguinte regra. Vejamos um exemplo.

Exemplo do Princípio de Unicidade
AdjnL: ADJ NOUN
Agreement: gender, number
%
SpecL: DET NOUN
Agreement: gender, number
%

Estas duas regras permitem analisar frases como *Um lindo carro*. A segunda regra não poderia analisar a frase, já que entre os dous elementos da dependência há um adjectivo. Uma vez que este actua como dependente na primeira regra, o princípio de unicidade elimina-a da entrada da segunda, pelo que esta é aplicada.

O princípio de unicidade reduz o espaço de aplicação das regras posteriores, sendo a sua definição simplificada. Depois de aplicadas todas as regras, o sistema gera uma representação em forma de triplets, que contêm o nome da dependência, o núcleo e o dependente (e outra informação morfossintáctica, em função do tipo de *output* escolhido):

(SpecL carro_NOUN_2; um_DET_0)
(AdjnL carro_NOUN_2; lindo_ADJ_1)

Existem, contudo, fenómenos linguísticos que são dificilmente analisáveis sob o princípio de unicidade (pensemos nos adjectivos predicativos, que têm dous núcleos: um verbo e um nome). Para tratar estes casos, DepPattern permite utilizar blocos de regras, ou operadores que não apagam o dependente.

Além disso, e com o fim de analisar as expressões semi-fixas, o formalismo contém mais dous tipos de dependências: *DepHead_lex* e *HeadDep_lex*, que criam unidades léxicas compostas potencialmente descontínuas (inserindo elementos opcionais entre o núcleo e o dependente). O exemplo antes referido, *ter [algo] em conta*, pode ser analisado assim:

Exemplo de Análise: ter [algo] em conta
TermR_lex: [VERB<lemma:ter>] [ADV]? PRP<lemma:em> NOUN<lemma:conta>
NEXT
ComplR_lex: VERB<lemma:ter> [ADV]? PRP<lemma:em> [NOUN<lemma:conta>]
%
AdjnR: VERB ADV
%

Este exemplo permite analisar, como foi dito, a expressão *ter em conta*, com um advérbio opcional entre o verbo e a preposição. Poder-se-iam acrescentar mais elementos opcionais, para tratar frases como *teve muitas questões hoje em conta*. DepPattern permite definir classes de palavras em ficheiros externos, pelo que a regra poderia utilizar uma classe de verbos entre os

quais se incluíssem, para além de *ter*, outros como *tomar* ou *levar*. Desta maneira, a regra analisaria frases como *ter/tomar/levar [algo] em conta*.

3.2 O Compilador

Uma vez escrita uma gramática, o sistema inclui um compilador, que gera através da própria gramática, um parser para analisar de maneira robusta os textos.

Os parsers têm como *input* texto plano, pelo que não precisa ser pré-processado antes de ser analisado por DepPattern. Antes da execução, o utilizador deve decidir que PoS-tagger (entre FreeLing e TreeTagger) deve utilizar DepPattern, bem como o *output* do parser. A este respeito, é preciso indicar que existem três saídas predefinidas (às que podemos acrescentar outras, usando *scripts* de conversão): O primeiro *output* é a análise básica, que mostra as dependências encontradas no parsing. O segundo, *full analysis*, para além das dependências analisadas, contém informação morfossintáctica e léxica de cada um dos tokens parseados.

Finalmente, uma terceira saída (*corrector*), tem o mesmo formato do PoS-tagger (neste caso, FreeLing), e pode ser utilizada, como veremos, para criar regras de correcção do etiquetador morfossintáctico.

3.3 Correcção de PoS-tagging

Os dous PoS-tagger que DepPattern permite utilizar (TreeTagger e FreeLing) funcionam com informação estatística (*decision trees* e Hidden Markov Models, respectivamente). Uma vez que estes métodos não utilizam directamente dados linguísticos na sua análise, a anotação produz de maneira sistemática alguns erros cujos padrões podem ser facilmente definidos.

Com base na saída *corrector*, uma gramática de DepPattern pode incluir regras que corrijam alguns desses erros sistemáticos, ou que façam um pré-processamento do texto, de modo a que outras regras posteriores tenham uma maior cobertura. Uma frase como *chegou a boas horas* é assim analisada por FreeLing:

```
chegou chegar VMIS3S0
a o DA0FS0
boas bom AQ0FP0
horas hora NCFP000
```

A preposição *a* é etiquetada pelo sistema como um artigo, mas não concorda em número com o nome do qual deveria depender. Para corrigir este tipo de casos, DepPattern permite utilizar um tipo de regras, *Single*, cujo operador *Corr* modifica a informação desejada (morfossintáctica, ou o próprio lema ou token):

```
Exemplo de Regra de Correcção
Single: DET<token:[Aa]> [NOUN<number:P>|ADJ<number:P>]
Corr: tag:PRP, type:P, lemma:a
%
```

Este exemplo de regra de correcção modifica qualquer determinante cujo token seja *A* ou *a* que preceda um nome ou um adjectivo em plural, por uma preposição (com lema *a*). A aplicação

desta regra permitiria, portanto, corrigir o erro referido. Contudo, antes de incluir uma regra deste tipo numa gramática, é preciso verificar a sua precisão, já que um padrão pouco robusto pode provocar mais erros do que correcções.

Para além da utilização do *output* de correcção, regras deste tipo podem ser aplicadas no início de uma gramática de análise sintáctica. Deste modo, podemos não apenas corrigir os erros de etiquetação do PoS-tagger (o que ampliaria a cobertura de regras de dependências), mas também modificar possíveis erros na identificação de nomes próprios, datas, ou outro tipo de entidades que possam ser relevantes para os objectivos da análise de dependências.

4 Conclusões

O presente trabalho apresentou dous recursos desenvolvidos pela linha de investigação ProLNat, focados no processamento morfossintáctico e na análise sintáctica multilíngue.

O primeiro deles, a adaptação de FreeLing para Português Europeu e Galego, permite realizar processamento morfossintáctico (nomeadamente tokenização, segmentação de orações, lematização e PoS-tagging) com níveis de desempenho próximos do estado-da-arte.

Com base na etiquetação morfossintáctica, DepPattern apresenta um formalismo para a criação de gramáticas de dependências, bem como um compilador que gera parsers robustos que permitem analisar grandes quantidades de *corpora*.

Além disto, com base no formalismo DepPattern, foi também apresentado um modelo de correcção de erros sistemáticos do PoS-tagger, que possibilita não apenas corrigir este tipo de erros, mas também realizar um pré-processamento do texto orientado para tarefas específicas como a extracção de informação.

Ambas as ferramentas são disponibilizadas sob licença GPL, pelo que o seu uso, distribuição e modificação são livres.

Referências

Atserias, Jordi, Elisabet Comelles e Aingeru Mayor (2005): "TXALA, un analizador libre de dependencias para el castellano". Procesamiento del Lenguaje Natural, 35: 455-456.

Barcala, Fco. Mario, Eva Mª Domínguez Noya, Pablo Gamallo Otero, Marisol López Martínez, Eduardo Miguel Moscoso Mato, Guillemo Rojo, María Paula Santalla del Río e Susana Sotelo Docío (2007): "A corpus and Lexical Resources for Multi-word Terminology Extraction in the Field of Economy in a in a Minority Language". Zygmunt Vetulani (ed.) *Human Language Technologies as a Challenge for Computer Science and Linguistics*. *Proceedings of the 3rd Language & Technology Conference:* 359-363 Wydawnictwo Poznaskie Sp. z o.o: Poznan.

Bick, Eckhard (2006): "A constraint grammar-based parser for spanish". *4th Workshop on Information and Human Technology*.

Branco, António e João Silva (2004): "Evaluating Solutions for the Rapid Development of State-of-the-Art POS Taggers for Portuguese". Lino, Maria Teresa, Maria Francisca Xavier, Fátima Ferreira, Rute Costa e Raquel Silva (eds.) *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*:507-510. ELRA: Pais.

Brants, Thorsten (2000): "TnT - A Statistical Part-of-Speech Tagger". *Proceedings of the 6th Conference on Applied Natural Language Processing, (ANLP 2000)*, ACL.

Carreras, Xavier, Isaac Chao, Lluís Padró e Muntsa Padró (2004): "FreeLing: An Open-Source Suite of Language Analyzers". *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*.

Eleutério, Samuel, Elisabete Ranchhod, Cristina Mota e Paula Carvalho (2003): "Dicionários Electrónicos do Português. Características e Aplicações". *Actas del VIII Simposio Internacional de Comunicación Social*: 636-642. Santiago de Cuba.

Gamallo, Pablo e Isaac González (2009): "Una gramática de dependencias basada en patrones de etiquetas". *XXV Congresso Anual de la Sociedad Española para el Procesamiento del Lenguaje Natural.* Donostia.

Gamallo, Pablo e Isaac González (2010): "A Grammatical Formalism based on Part-of-Speech Tags". *International Journal of Corpus Linguistics* (no prelo).

Garcia, Marcos e Pablo Gamallo (2010): "Using Morphosyntactic Post-processing to Improve PoS-tagging Accuracy". *Proceedings of the International Conference on Computational Processing of Portuguese (PROPOR 2010): Extended Activities*. Porto Alegre.

Hunston, Susan e Gill Francis (1999): *Pattern Grammar*. John Benjamins: Amsterdam.

Leach, Geoffrey e Andrew Wilson (1996): "Recommendations for the Morphosyntactic Annotation of Corpora". Relatório Técnico. Expert Advisory Group on Language Engineering Standard (EAGLES).

Megyesi, Beáta (2001): "Comparing Data-Driven Learning Algorithms for PoS Tagging of Swedish". *Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing:* 151-158.

Nivre, Joakim (2005): "Dependency Grammar and Dependency Parsing". Relatório Técnico. Växjö University: School of Mathematics and Systems Engineering.

Padró, Lluís (1998): *A Hybrid Environment for Syntax-Semantic Tagging*. Tese de Doutoramento. Dept. Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya.

Schmid, Helmut (1994): "Probabilistic Part-of-Speech Tagging Using Decision Trees". *Proceedings of the International Conference on New Methods in Language Processing:* 44-49.

Sinclair, John (2001): *Corpus, Concordance, Collocation*. Oxford University Press: Oxford.