CURSO DE MESTRADO EM ENGENHARIA INFORMÁTICA

ANÁLISE E DESENVOLVIMENTO DE FERRAMENTAS COMPUTACIONAS PARA O PROCESSAMENTO DE LÍNGUA PORTUGUESA

VOL. I

José Eduardo de Carvalho Rosado Saianda

Orientador: Prof. Doutor Paulo Miguel Torres Duarte Quaresma

Évora, Dezembro de 2011

CURSO DE MESTRADO EM ENGENHARIA INFORMÁTICA

ANÁLISE E DESENVOLVIMENTO DE FERRAMENTAS COMPUTACIONAS PARA O PROCESSAMENTO DE LÍNGUA PORTUGUESA

VOL. I

José Eduardo de Carvalho Rosado Saianda

Orientador: Prof. Doutor Paulo Miguel Torres Duarte Quaresma

Évora, Dezembro de 2011

AGRADECIMENTOS

Chegado o final da elaboração deste trabalho e ultrapassadas todas as fases boas e menos boas, mas todas elas produtivas, este é o momento para agradecer a todos os que me incentivaram, ajudaram e para ele contribuiram.

Antes de mais, gostaria de agradecer ao Professor Doutor Paulo Quaresma a sua orientação sempre pronta e a sua disponibilidade. Agradeço-lhe todos os apontamentos e correcções assinalados e todas as críticas construtivas que entedeu fazer durante a elaboração da minha tese.

Gostaria igualmente de agradecer ao Centro de Investigação em Tecnologias da Informação da Universidade de Évora (CITI-UE) a disponibilização de uma máquina virtual para alojamento do portal que serviu de base à elaboração deste trabalho.

Agradeço de uma forma muito especial aos meus pais e a toda a família pelo incentivo e apoio no decorrer da elaboração deste trabalho.

Não posso deixar de expressar o meu agradecimento muito especial à minha mãe pela disponibilidade que sempre mostrou aquando da revisão do Português desta tese e à Tita - minha irmã - pela paciência que teve na formatação da mesma.

Jamais poderia deixar de agradecer à Olga Gonçalves a tradução para Inglês do resumo deste trabalho.

Seria muito injusto não deixar aqui uma palavra de agradecimento a alguns amigos muito próximos que souberam ouvir os meus desabafos.

Obrigado Cláudia e Rui, Rita, Miguel, Carlos, Margarida e Lucinda.

Dedico este trabalho, com todo o carinho, à minha Família.

RESUMO

"Análise e desenvolvimento de ferramentas computacionais para o processamento de Língua Portuguesa"

Actualmente, devido à explosão da Internet e ao número cada vez maior de ferramentas existentes para o processamento da língua portuguesa, surge a necessidade de concentrar num único local essas mesmas ferramentas com vista a melhorar e rentabilizar a sua utilização.

Posteriormente à pesquisa das principais ferramentas *open source* existentes nesta área, verificámos que seria bastante útil dispormos de um único local de fácil acessibilidade ao cidadão comum.

Para tal, julgámos que devido à sua facilidade de utilização e ao elevado número de extensões disponíveis, o CMS Joomla! seria o mais adequado para levar a cabo a tarefa pretendida.

Após a análise do trabalho desenvolvido, foi-nos possível concluir que a grande maioria das ferramentas disponíveis no portal são gratuitas, o que poderá constituir um enorme incentivo à sua utilização.

ABSTRACT

"Analysis and development of computational tools for the processing of Portuguese language"

At present, due to the wealth of resources provided by the Internet and to the increasing number of tools for the processing of the Portuguese language, there is the need to gather on a single site those same tools so as to improve and benefit from their use.

After having researched the main open source tools in this field, the usefulness of a single site which could easily be accessed by the common citizen became apparent.

Therefore, CMS Joomla! was thought to be the most adequate tool to meet that purpose due to its user-friendly access and to the high number of extensions available.

Following the analysis of the work developed, we were able to conclude that the great majority of the tools available on its portal are free of charge, which may encourage their use.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE GRÁFICOS

ÍNDICE DE TABELAS

LISTA DE SIGLAS

ANSI - *American National Standards Institute* (Instituto Nacional Americano de Padronização)

API - *Application Programming Interface* (Interface de Programação de Aplicações)

CCSL - Centro de Competência em Software Livre

CFG - *Context Free Grammar* (Gramática Livre de Contexto)

CLUL - Centro de Linguística da Universidade de Lisboa

CMS - *Content Management System* (Sistema Gestor de Conteúdos)

CNPTIA - Embrapa Informática Agropecuária

DLL - *Dynamic Link Library*

DTMF - *Dual-Tone Multi-Frequency*

EWN - *EuroWordNet*

FTP - *File Transfer Protocol* (Protocolo de Transferência de Ficheiros)

GPL - *GNU General Public License*

GETerm - Grupo de Estudos e Pesquisas em Terminológicos

HMM - *Hidden-Markov-Model*

HTML - *HyperText Markup Language*(Linguagem de Marcação de Hipertexto)

IGM - *Institut D'Electronique et d'Informatique Gaspard-Monge*

INESC - Instituto de Engenharia de Sistemas e Computadores

IULA - *Institut Universitari de Lingüística Aplicada*

LabInfo - Laboratório de Organização e Tratamento da Informação Electrónica

LAEL - Lingüística Aplicada e Estudos da Linguagem

LDAP - *Lightweight Directory Access Protocol (*Protocolo que actualiza e pesquisa directórios, funcionando sobre TCP/IP)

LGPL - *GNU Lesser General Public License*

LGPLLR - *Lesser General Public License for Linguistic Resources*

LIA - *Laboratoire Informatique d'Avignon*

N-Grama - Sequência de n *tokens* que ocorrem numa janela com tamanho igual ou superior a n.

NILC - Núcleo Interinstitucional de Linguística Computacional

OCR - *Optical Character Recognition* (Reconhecimento óptico de caracteres)

Parser - Programa que subdivide uma entrada (*input*) para que um outro possa actuar sobre ela; analisador gramatical

PDF - *Portable Document Format*

PHP - *Hipertext Preprocesor*[1]

PoS - *Part-of-Speech*

SASKIA - Interface responsável pelo pré-processamento das colecções da Wikipédia, e pela realização de uma classificação inicial às entidades mencionadas, com base na informação extraída da Wikipédia

SEF - *Search Engine Friendly URLs*

TEI – *Text Encoding Initiative* - Consórcio que desenvolve e mantém um padrão para a representação de textos em forma digital

TMX - *Translation Memory eXchange* - padrão XML utilizado para a troca de dados da memória de tradução, criado pela Computer Aided Translation (CAT)

UFRGS - Universidade Federal do Rio Grande do Sul

UFSCar - Universidade Federal de São Carlos

URL - Uniform Resource Locator

USP - Universidade de São Paulo

WN - WordNet[2]

WYSIWYG - *What You See Is What You Get*

XHTML - *eXtensible Hypertext Markup Language* - reformulação da linguagem HTML baseada em XML

XML - *Extensible Markup Language* - Conjunto de regras de codificação de documentos numa forma legível pelo computador

[1] http://www.php.net

[2] http://wordnet.princeton.edu/

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Uma das necessidades inerentes a todos os seres, e muito especialmente os seres humanos, é o viver com os seus semelhantes, viver esse que implica, naturalmente, e desde sempre, o contacto o Outro com vista à plena realização do SER.

Nesta condição *sine qua non* da sobrevivência assume particular importância a comunicação e, obviamente, a linguagem, um dos meios – sem dúvida o mais importante – que o homem tem à sua disposição para se fazer compreender e manifestar que também pode compreender.

Pela língua e com a língua evoluiu o Homem, evoluíram os grupos, as sociedades, os países. Com ela se construíram impérios e também com ela outros se desmoronaram.

A reflexão sobre este instrumento poderosíssimo que o mesmo Homem tem à sua disposição para dominar o mundo ou o seu mundo, suscitou desde tempos muito antigos – Antiguidade Grega e Latina - estudos de natureza variada cuja pertinência não se põe em dúvida muitos séculos depois. Na nossa era.

Mas o facto de se reconhecer a intemporalidade da "Retórica" de Aristóteles, por exemplo, não significa que outros estudos, outras teorias não tenham surgido ao longo das eras e não tenham também dado o seu contributo para a evolução do estudo da língua e das línguas.

Considerando tempos mais próximos, ainda que do século passado, não podemos esquecer que este estudo foi inicialmente feudo da Linguística – diacrónica e sincrónica – sobretudo a partir de investigadores como Saussure, Benveniste e Bakhtine.

Previligiando o primeiro uma investigação de natureza mais estruturalista e também mais teórica da linguagem, e apontando os segundo e terceiro para uma vertente interdiscursiva e polifónica, todos foram pioneiros nos estudos que empreenderam e que, por sua vez, deram origem a outros estudos e outras interpretações.

Outras disciplinas nasceram.

Outras vertentes de investigação surgiram. Novas luzes permitiram que interpretações, talvez mais profundas e sem dúvida mais rápidas e rigorosas, fossem efectuadas.

Uma dessas vertentes - a que previligiamos no estudo empreendido - é a que se refere ao processamento de língua natural, **resultado** do cruzamento entre os progressos da Informática e da Linguística.

Na investigação levada a cabo verificámos a existência de uma enorme diversidade de ferramentas para o processamento da língua, facto que atesta o interesse dos investigadores, tanto do campo da informática como do da Linguística. Pelo facto de nos interessar particularmente a língua portuguesa, na pesquisa que empreendemos demos prioritariamente atenção às ferramentas já disponíveis para este efeito.

Pelo facto de não serem de livre utilização e também porque a sua dispersão obstava a que os potenciais utilizadores as encontrassem com facilidade, decidimos, para colmatar estes problemas, concentrar num único portal, e classificando-as em diversas categorias, as já citadas ferramentas open source mais importantes. Esperamos, assim, dar o nosso contributo para que novas e ainda mais profícuas investigações venham a concretizar-se e que os potenciais utilizadores fiquem com liberdade para:[3]

- executar o s*oftware* para a utilização que lhes interessar;
- estudar o funcionamento de um programa e adaptá-lo às suas necessidades;
- redistribuir cópias;
- melhorar o programa e tornar as modificações públicas de modo que a comunidade inteira beneficie da melhoria.

[3] http://ansol.org/filosofia/softwarelivre.pt.html

1.2 OBJECTIVOS

O principal objectivo deste trabalho consiste, pois, no inventário das principais ferramentas *open source* existentes para o processamento da língua portuguesa em várias vertentes. Pretendemos assim promover a sua disponibilização concentrando-as num único local acessível à comunidade.

1.3 METODOLOGIA

Para a concretização do portal em causa foi necessário proceder inicialmente ao levantamento das ferramentas existentes - descrito no capítulo II - a partir das seguintes fontes:

- Linguateca[4];
- NILC – Núcleo Interinstitucional de Linguística Computacional[5];
- Pablo Gamallo[6];
- LX-Center[7].

após o que se procedeu ao seu agrupamento em dez categorias, de acordo com o proposto pelo projecto LINGUATECA coordenado e executado pela FCCN, cujo responsável máximo é Pedro Veiga[8]:

- ajuda ao ensino;
- ajuda à redacção;
- alinhadores;

4 http://www.linguateca.pt/
5 http://www.nilc.icmc.usp.br/nilc/
6 http://gramatica.usc.es/~gamallo/
7 http://lxcenter.di.fc.ul.pt/services/en/LXServicesSuite.html
8 http://www.fccn.pt

- analisadores;
- conjugadores verbais;
- extractores de N-Gramas;
- ferramentas especializadas;
- processamento de fala;
- sumarizadores;
- tradução automática.

Posteriormente, foi instalado o CMS Joomla! num servidor[9] procedendo-se em seguida à elaboração do portal que se encontra descrita no capítulo III.

1.4 RESULTADOS ESPERADOS

Com a elaboração deste trabalho esperamos tornar mais fácil e acessível o estudo das principais ferramentas existentes para o processamento da língua portuguesa em várias vertentes. Apresentaremos dicionários, ferramentas de apoio à redacção, alinhadores e analisadores de textos, ferramentas para fins específicos e geradores de sumários e finalmente auxiliares ao processamento de fala e à tradução;

É nossa intenção igualmente elaborar um portal, que ficará disponível em http://saianda.xdi.uevora.pt/joomla/, e que conterá o maior número possível de ferramentas *open source* para processamento da língua portuguesa nos seus mais variados aspectos.

9 http://saianda.xdi.uevora.pt/joomla/

1.5 ORGANIZAÇÃO DA TESE

O próximo capítulo descreve o "estado da arte" relativamente às ferramentas existentes para o processamento da língua portuguesa.

O capítulo 3 descreve as ferramentas utilizadas no desenvolvimento do portal de ferramentas para processamento da língua portuguesa.

No capítulo 4 é efectuada uma análise estatística global acerca das ferramentas existentes, bem como uma análise por categoria no que se refere à disponíbilidade das ferramentas (para download, online ou ambas), bem como a sua gratuitidade.

No capítulo 5 são comentadas as conclusões que o trabalho desenvolvido permitiu extrair, bem como possíveis melhorias a introduzir.

2 DESCRIÇÃO DAS FERRAMENTAS EXISTENTES NO PORTAL DE FERRAMENTAS PARA PROCESSAMENTO DE LÍNGUA PORTUGUESA

A realização deste trabalho exigiu que fosse feito o levantamento das ferramentas existentes, agrupando-as em diversas categorias conforme se mostra em seguida, e de acordo, como já referimos, com a classificação proposta pelo portal da LINGUATECA.

2.1 AJUDA AO ENSINO

2.1.1 Curso de Português do Brasil

Este curso, criado pela Brasiliano[10] e que resulta de uma experiência intensiva em ensino de Português para estrangeiros, pretende auxiliar pessoas de diferentes nacionalidades que desejam familiarizar-se com termos e expressões utilizados no dia-a-dia.

Figura 1 - Dicionário de Português do Brasil disponível em CD-ROM - níveis 1 e 2.

2.1.2 Dicionários de Português-Inglês

Ferramentas desenvolvidas pela Ectaco [11] disponíveis online em http://www.ectaco.com/dictionaries/portuguese.asp, que auxiliam a aprendizagem da língua inglesa.

[10] http://www.brasiliano.net/

[11] http://www.ectaco.com/

2.1.3 Gramática interactiva do Português

Gramática da língua portuguesa disponível online em http://visl.hum.sdu.dk/itwebsite/port/portgram.html.

2.1.4 Minigramática

Ferramenta desenvolvida pelo NILC[12] que permite obter dados relativos à morfologia e à sintaxe da língua portuguesa. Com esta minigramática podemos

- fazer o levantamento de
  - classes gramaticais

- identificar
  - termos da oração
  - concordância
  - regência verbal
  - colocação pronominal

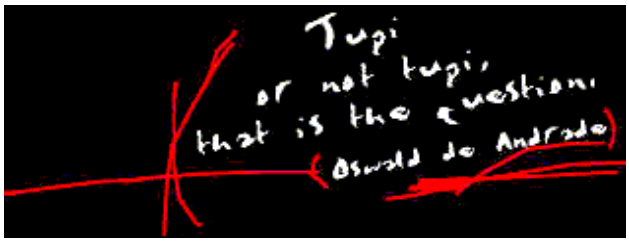


Figura 2 - Minigramática da Língua Portuguesa

[12] http://www.nilc.icmc.usp.br/minigramatica/mini/sejabemvindo.htm

2.2 AJUDA À REDACÇÃO

2.2.1 Aspell

Ferramenta desenvolvida por Kevin Atkinson[13] e disponível sob licença LGPL, que tem como objectivo realizar a correcção ortográfica de um texto ou funcionar apenas como uma biblioteca a utilizar por outras ferramentas.

2.2.2 CoGrOO

Corrector gramatical desenvolvido pelo CCSL[14], acoplável ao "*package*" OpenOffice[15], disponível sob licença LGPL, e que detecta os seguintes tipos de erros:

- colocação pronominal;
- concordância nominal;
- concordância entre sujeito e verbo;
- concordância verbal;
- utilização de crase;
- regulação nominal;
- regulação verbal;
- erros comuns da língua portuguesa escrita.

2.2.3 Correctores ortográficos

Ferramentas de correcção ortográfica desenvolvidas pela Universidade do Minho no âmbito do Projecto Natura[16], disponíveis sob licença GPL, para utilização conjunta com o Browser Mozilla FireFox[17] e com o "*package*" OpenOffice.

[13] http://kevin.atkinson.dhs.org/

[14] http://ccsl.ime.usp.br/

[15] Disponível para download gratuito em http://download.openoffice.org/index.html

[16] http://natura.di.uminho.pt/wiki/doku.php?id=dicionarios:main

[17] Disponível para download gratuito em http://www.mozilla-europe.org/pt/firefox/

2.2.4 FliP

Pacote de ferramentas disponível pela Priberam Informática[18] para processamento da língua portuguesa composto pelos seguintes módulos:

2.2.4.1 *Auxiliares de tradução*

Ferramentas que permitem obter sugestões de tradução de palavras ou expressões de ou para Português, Espanhol, Francês ou Inglês conforme mostrado na Figura 3;

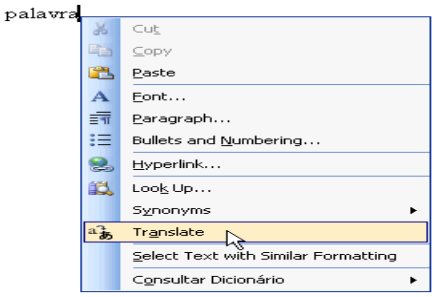


Figura 3 - Acesso às ferramentas de tradução do FliP.

Outro modo de se obter uma tradução consiste em seleccionar a opção "Idioma" ("Language") a partir do menu "Ferramentas"("Tools") e em seguida "Traduzir"("Translate").

Em ambos os casos, na caixa de diálogo que surge à direita, deverá seleccionar-se a língua de origem (que corresponde à língua na qual a palavra se encontra escrita) e a lingua de destino (que corresponde à língua para a qual se pretende obter a tradução da palavra seleccionada), conforme mostrado na Figura 4.

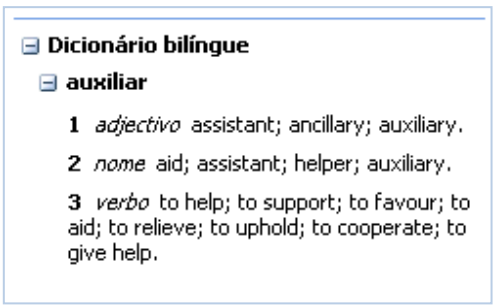


Figura 4 - Tradução de palavras de Português para Inglês utilizando o FliP

[18] http://www.flip.pt/

As traduções apresentadas encontram-se agrupadas por classe gramatical, como se pode observar no exemplo seguinte:

Figura 5 - Dicionário bilingue do FliP

e dizem respeito sempre às formas canónicas, ainda que a palavra a traduzir seja uma flexão. É o caso, por exemplo, de formas verbais conjugadas, de plurais ou de femininos.

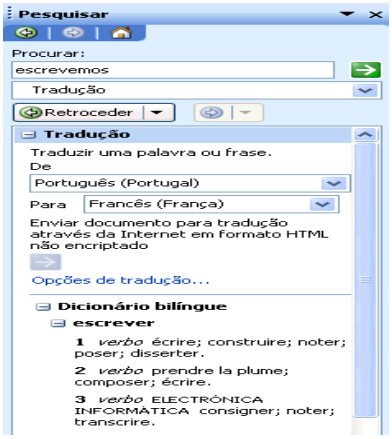


Figura 6 - Tradução de Português para Francês utilizando o Flip.

Os auxiliares de tradução funcionam essencialmente com palavras isoladas, mas também apresentam traduções para um conjunto muito significativo de locuções.

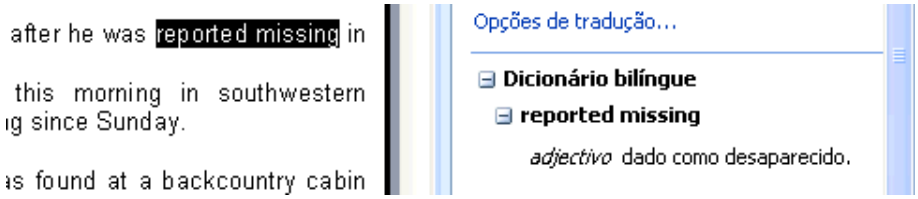


Figura 7 - Opções de tradução do FliP

A utilização do auxiliar de tradução é possível com formas flexionadas em espanhol e inglês, mas não com formas flexionadas em francês, a não ser que se trate de uma locução lexicalizada ou combinatória fixa.

2.2.4.2 *Configurador*

O configurador possui quatro funções distintas:

- selecção de dicionários temáticos a utilizar pelo corrector ortográfico (Figura 8);

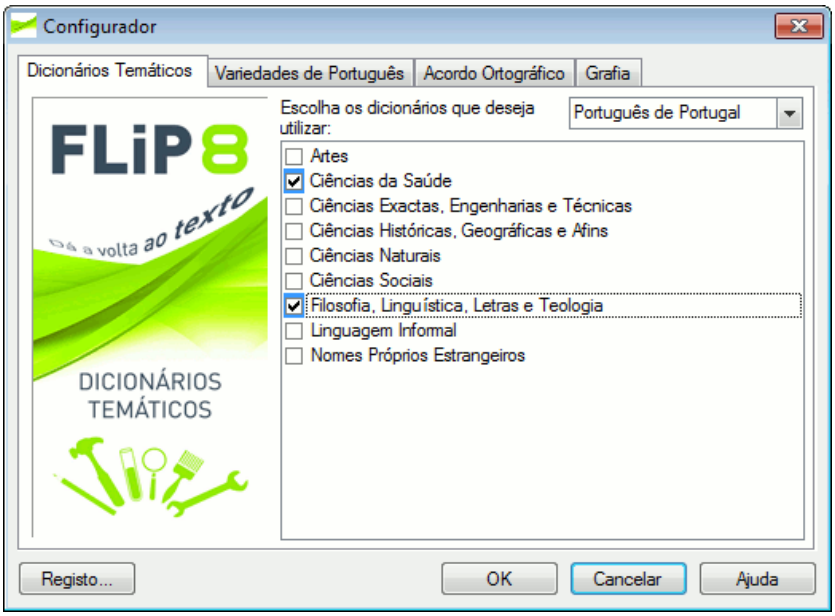


Figura 8 - Selecção de dicionários temáticos do FliP

- seleccão de várias possibilidades de português para o corrector ortográfico (Angola, Cabo Verde, Galiza, Guiné-Bissau, Macau, Moçambique, São Tomé e Timor) como se pode ver na Figura 9;

Figura 9 - Selecção de variedades de Português do FliP .

– selecção da utilização ou não da ortografia segundo o Acordo Ortográfico de 1990 (Figura 10);

Figura 10- Activação do acordo ortográfico no FliP .

– personalização da utilização da grafia nos casos em que o Acordo Ortográfico de 1990 permite duplas grafias, como se verifica no exemplo seguinte

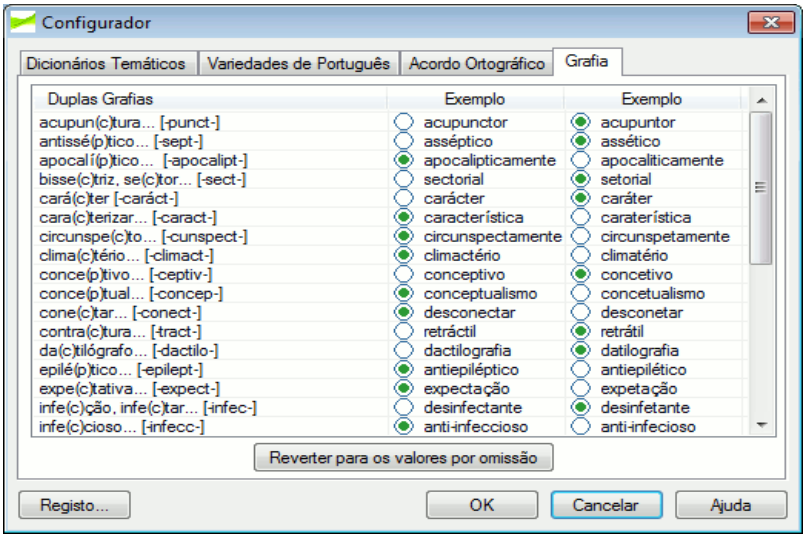


Figura 11 - Selecção das diferentes grafias que o Acordo Ortográfico permite.

2.2.4.3 *Conjugador*

A ferramenta FliP inclui três conjugadores verbais conforme mostra a Figura 12:

– Português Europeu;

– Português do Brasil;

– Espanhol.

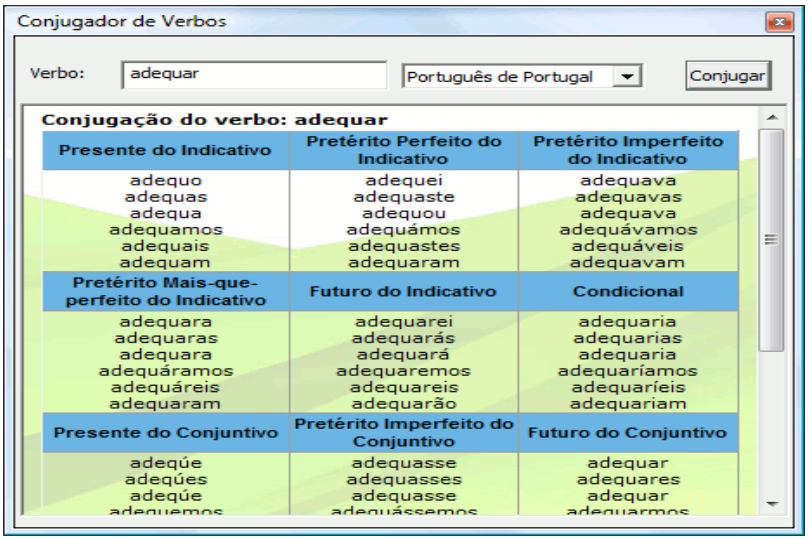


Figura 12 - Conjugador verbal do FliP

2.2.4.4 *Conversor para o Acordo Ortográfico*

Este módulo tem como objectivo a realização da conversão automática de um texto ou de uma parte seleccionada de um texto para a ortografia segundo o Acordo Ortográfico de 1990. Para levar a cabo esta funcionalidade, deverá aceder-se ao menu "FliP" e seleccionar a opção "Converter Texto para Acordo Ortográfico". Se o utilizador pretender converter apenas uma parte do documento, deverá seleccionar o texto a converter e marcar a opção "Converter apenas o texto seleccionado", como se mostra na Figura 13.

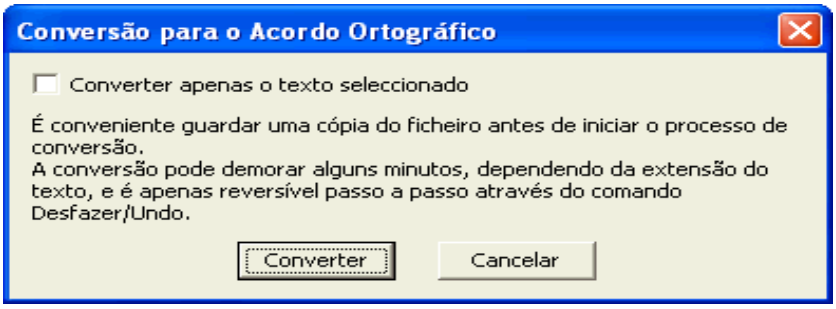


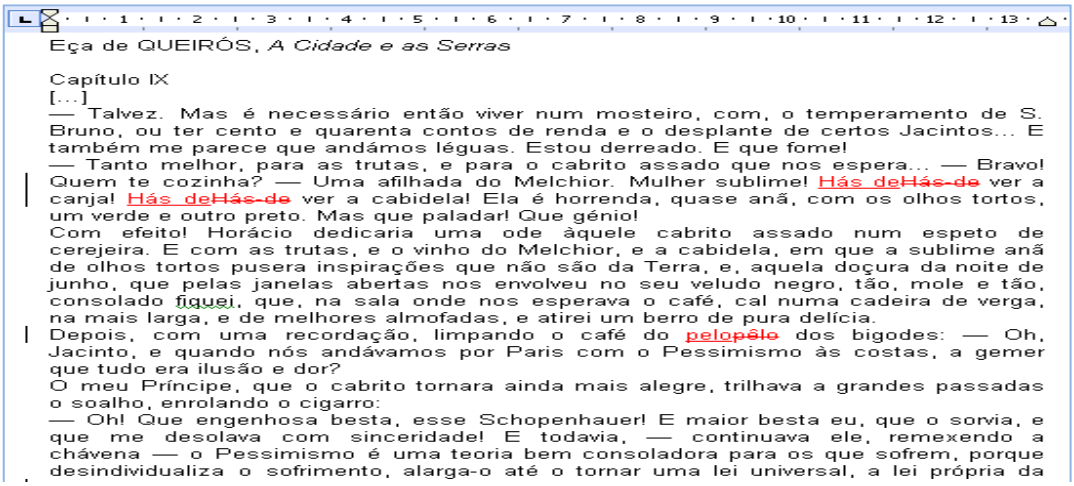
Figura 13 - Conversor do FliP para o Acordo Ortográfico.

A conversão pode demorar algum tempo, o que obviamemnte depende da extensão do texto. É aconselhável utilizar o editor de texto contido no FLiP – o FliPEd – uma vez que neste a operação de conversão é mais rápida do que em outros editores ou processadores de texto.

É aconselhável ao utilizador efectuar uma cópia do ficheiro antes de iniciar o processo de conversão, uma vez que esta operação apenas é reversível passo a passo através do comando "Desfazer/Undo". A conversão inversa (isto é, a conversão de um texto com a ortografia segundo o Acordo Ortográfico de 1990 para a ortografia anterior) não é possível.

Em alguns processadores de texto as alterações efectuadas através da conversão automática ficam registadas no documento. Para as rever, e aceitá-las ou rejeitá-las carregando com o botão direito do rato em cada alteração. Poder-se-á ainda aceitar as alterações na íntegra ou revê-las através da barra de ferramentas de Revisão.

O resultado obtido é ilustrado na Figura 14.


Eça de QUEIRÓS, *A Cidade e as Serras*

Capítulo IX
[...]
— Talvez. Mas é necessário então viver num mosteiro, com, o temperamento de S. Bruno, ou ter cento e quarenta contos de renda e o desplante de certos Jacintos... E também me parece que andámos léguas. Estou derreado. E que fome!
— Tanto melhor, para as trutas, e para o cabrito assado que nos espera... — Bravo! Quem te cozinha? — Uma afilhada do Melchior. Mulher sublime! Hás deHás-de ver a canja! Hás deHás-de ver a cabidela! Ela é horrenda, quase anã, com os olhos tortos, um verde e outro preto. Mas que paladar! Que génio!
Com efeito! Horácio dedicaria uma ode àquele cabrito assado num espeto de cerejeira. E com as trutas, e o vinho do Melchior, e a cabidela, em que a sublime anã de olhos tortos pusera inspirações que não são da Terra, e, aquela doçura da noite de junho, que pelas janelas abertas nos envolveu no seu veludo negro, tão, mole e tão, consolado figuei, que, na sala onde nos esperava o café, cal numa cadeira de verga, na mais larga, e de melhores almofadas, e atirei um berro de pura delícia.
Depois, com uma recordação, limpando o café do pelopêlo dos bigodes: — Oh, Jacinto, e quando nós andávamos por Paris com o Pessimismo às costas, a gemer que tudo era ilusão e dor?
O meu Príncipe, que o cabrito tornara ainda mais alegre, trilhava a grandes passadas o soalho, enrolando o cigarro:
— Oh! Que engenhosa besta, esse Schopenhauer! E maior besta eu, que o sorvia, e que me desolava com sinceridade! E todavia, — continuava ele, remexendo a chávena — o Pessimismo é uma teoria bem consoladora para os que sofrem, porque desindividualiza o sofrimento, alarga-o até o tornar uma lei universal, a lei própria da

Figura 14 - Palavras sugeridas pelo conversor de Acordo Ortográfico do FliP

A conversão respeita as regras estabelecidas na personalização das duplas grafias que podem ser alteradas no Configurador do FLiP.

2.2.4.5 *Corrector Ortográfico*

O corrector ortográfico detecta erros de ortografia e apresenta sugestões para a sua correcção.

O funcionamento do corrector ortográfico é baseado na comparação das palavras utilizadas num documento com uma lista de palavras conhecidas pelo módulo de correcção ortográfica. No processo de verificação ortográfica são utilizados os dicionários temáticos que estiverem seleccionados. Se uma determinada palavra não é reconhecida, esta é assinalada como errada e são apresentadas sugestões para a sua correcção.

Alguns processadores de texto permitem que a verificação ortográfica seja feita automaticamente enquanto se escreve. Neste caso, os erros ortográficos aparecem sublinhados a vermelho, como ilustra a Figura 15.

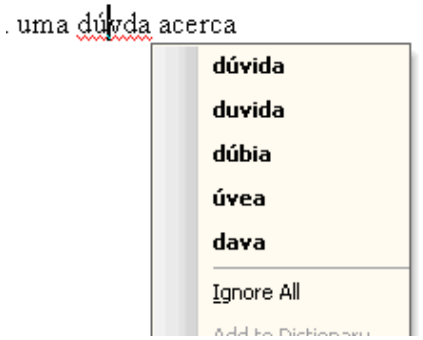


Figura 15 - Sinalização de erros ortográficos pelo FliP .

Carregando sobre a palavra assinalada com o botão direito do rato, poderá seleccionar-se uma das sugestões de substituição, ignorar a palavra ou adicioná-la ao dicionário de utilizador.

Normalmente o utilizador seleccionará uma das sugestões apresentadas para substituir a palavra incorrecta que se encontra no documento. Sempre que possível, as sugestões são apresentadas por ordem decrescente de probabilidade de se tratar da palavra correcta. Na ordenação das sugestões são considerados factores como a semelhança fonética, a semelhança gráfica (importante em textos reconhecidos por OCR), adjacências no teclado e frequência das palavras.

O conjunto das palavras conhecidas pelo corrector ortográfico é constituído pela reunião da lista de palavras geral, das listas de palavras dos dicionários temáticos seleccionados e das palavras adicionadas aos dicionários de utilizador activos.

Os tipos de erro corrigidos automaticamente englobam, entre outros, erros de acentuação (por exemplo, a palavra *abaco* é imediatamente corrigida para ábaco), erros de colocação do til (por exemplo, a palavra *órfao* é imediatamente corrigida para *órfão*), erros de colocação de cedilha (por exemplo, a palavra *françês* é imediatamente corrigida para *francês*), erros por semelhanças fonéticas (por exemplo, a palavra *anabolisante* é imediatamente corrigida para *anabolizante*), erros por uso de maiúsculas e minúsculas na mesma palavra (por exemplo, a palavra *CAMPAINHas* é imediatamente corrigida para *CAMPAINHAS*), erros por trocas muito frequentes de letras (por exemplo, a palavra *casamnetos* é imediatamente corrigida para *casamentos*), erros por ausência de consoante muda (por exemplo, *abstração* é imediatamente corrigida para *abstracção*) e erros de hifenização (por exemplo, a palavra *vicepresidente* é imediatamente corrigida para vice-*presidente* e a palavra *hemi-ciclo* é automaticamente corrigida para *hemiciclo*).

A utilidade da correcção automática é particularmente visível no caso da activação do novo Acordo Ortográfico. Com a opção Activar Acordo Ortográfico seleccionada, grafias que sofreram alterações são corrigidas automaticamente, nomeadamente os casos de presença de consoante muda (por exemplo, *abstracção* é imediatamente corrigida para *abstração*), os casos de hifenização ou falta dela (por exemplo, *auto-estrada* é imediatamente corrigida para *autoestrada* e *microondas* é imediatamente corrigida para *micro-ondas*) ou ainda os casos de presença de acentuação gráfica (por exemplo, *andróide* é imediatamente corrigida para *androide*).

2.2.4.6 *Corrector Sintáctico e Estilístico*

O corrector sintáctico funciona também como um verificador estilístico que chama a atenção para a utilização de registos de língua mais restritos (ex.: regionalismos, palavras ou expressões informais) e verifica a correcção da pontuação (avisando, por exemplo, quando é utilizada uma vírgula entre o sujeito e o verbo, ou quando se introduziram espaços antes de um sinal de pontuação).

Além disto, o corrector sintáctico assinala vários erros ortográficos que não podem ser detectados pelo corrector ortográfico pelo facto de este não dispor de informação de contexto. Um caso típico é o das palavras compostas ligadas por hífen. Expressões como "pequeno almoço" são detectadas e é sugerida a respectiva substituição por "pequeno-almoço".

Outro caso é o da confusão entre as palavras "à" (contracção da preposição *a* com o artigo *a*) e "há" (forma verbal do verbo *haver*). Em qualquer dos casos referidos não existe erro ortográfico se as palavras forem consideradas isoladamente; apenas a análise do contexto de ocorrência permite decidir sobre a correcção ortográfica do texto em análise.

Quando é detectado um erro, a frase ou parte da frase é sublinhada a verde e são apresentadas, sempre que possível, sugestões de correcção como se mostra na Figura 16.

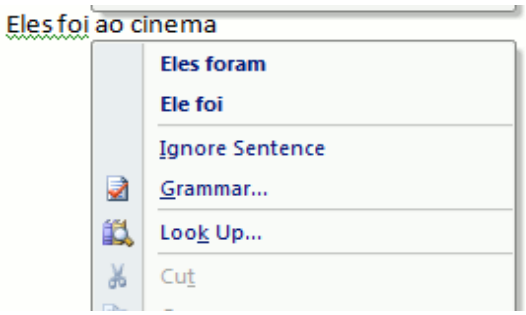


Figura 16 - Sinalização de erros em frases pelo FliP.

Alguns "*packages*" permitem a visualização da explicação dos erros gramaticais, como se pode ver na Figura 17.

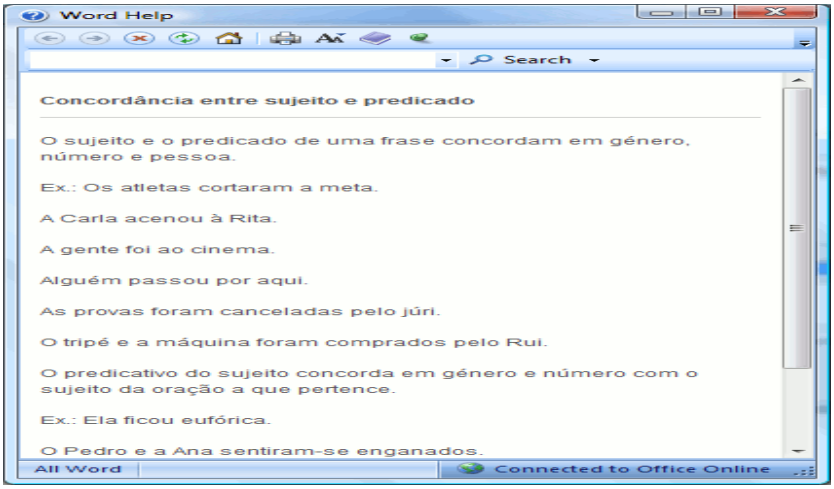


Figura 17 - Explicação de erros gramaticais no FliP.

Os diversos tipos de erro[19] detectados pelo corrector sintáctico podem ser configurados separadamente para cada um dos quatro estilos de escrita predefinidos: Formal, Corrente, Informal e Personalizado, conforme se pode ver na Figura 18.

[19] Os tipos de erro diferem ligeiramente consoante se trate da variante de português europeu ou da variante de português do Brasil, e divergem um pouco mais relativamente ao corrector sintáctico para espanhol.

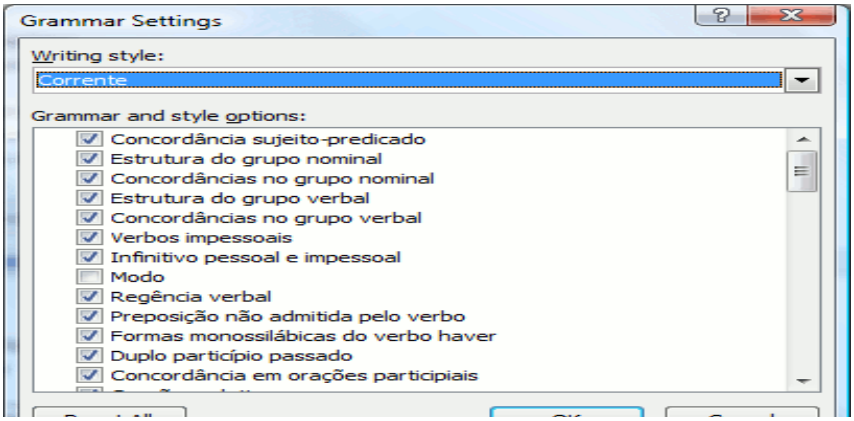


Figura 18 - Activação de características gramaticais no FliP

Perante esta diversidade, a configuração do estilo da verificação sintáctica e estilística deve ser feita em função do tipo de documento e da audiência ou destinatário ao qual é dirigido. Assim, por exemplo, configurado para um estilo Formal, o mais restritivo, o corrector sintáctico adverte o utilizador do emprego de palavras ou expressões informais ou de calão (o que não acontece quando configurado para uma escrita no estilo Informal).

2.2.4.7 *Dicionário Priberam da Língua Portuguesa*

O *Dicionário Priberam da Língua Portuguesa* é um dicionário de português europeu (de Portugal) que contém cerca de 97 000 entradas lexicais, incluindo locuções e fraseologias, que permite a consulta de definições, com sinónimos e antónimos por acepção, sub-entradas e locuções. A obra, que tem por base o *Novo Dicionário Lello da Língua Portuguesa* (Porto, Lello Editores, 1996 e 1999), foi adaptada pela Priberam para formato adequado à disponibilização electrónica e tem sido revista e modificada pela sua equipa de linguistas, estando em constante actualização e melhoramento.

Para aceder ao dicionário foi incluído um suplemento que entre outras funcionalidades, permite a consulta do dicionário Priberam directamente a partir do processador de texto, como se mostra na Figura 19.

Figura 19 - Acesso ao dicionário Priberam a partir do processador de texto

Se o utilizador pretende consultar o significado de uma palavra ou obter outras informações lexicais (etimologia, fonética, morfologia, etc.), deverá seleccionar a palavra e carregar em "Dicionário", como se mostra na Figura 20.

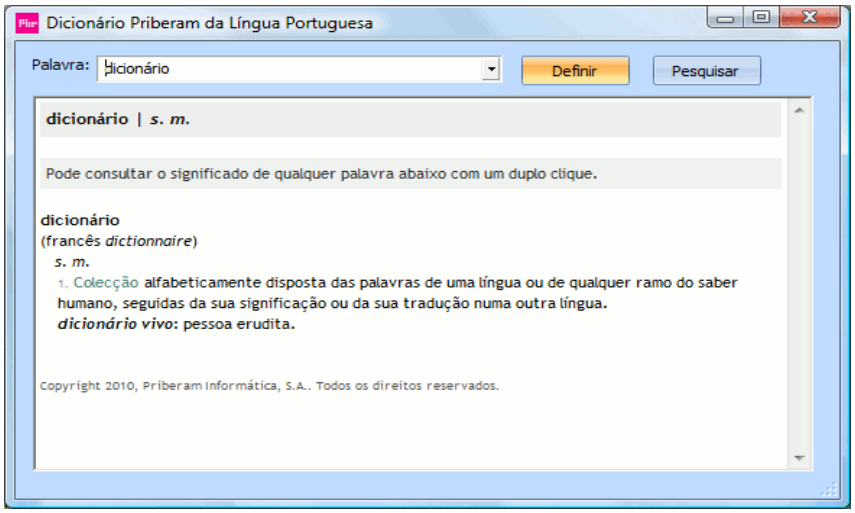


Figura 20 - Consulta de significados de palavras a partir do dicionário da *Priberam*

O *Dicionário Priberam da Língua Portuguesa* permite a consulta com ou sem as alterações gráficas previstas pelo Acordo Ortográfico de 1990, consoante a selecção feita pelo utilizador no "Configurador" do FLiP, como se pode ver na Figura 21.

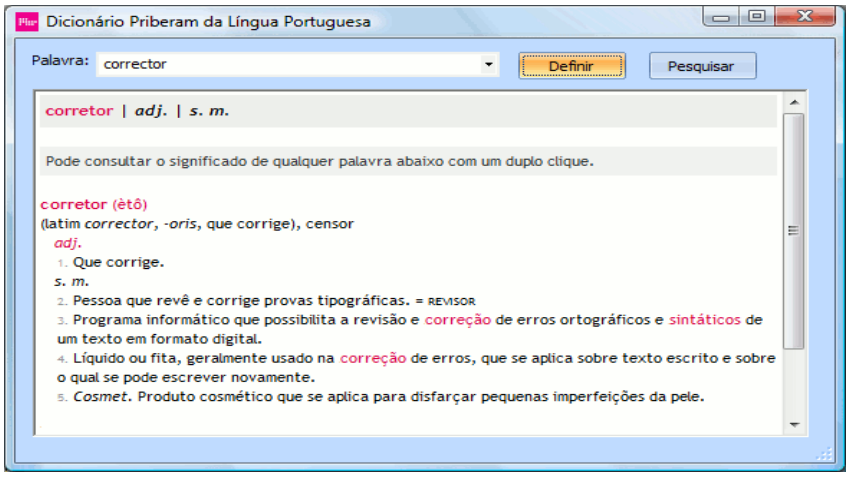


Figura 21 - Consulta de palavras com ou sem alterações previstas pelo Acordo Ortográfico

2.2.4.8 *Dicionário de Sinónimos*

O dicionário de sinónimos é uma ferramenta especialmente útil para evitar repetições de palavras. Através deste dicionário é possível obter, de uma forma quase imediata, sinónimos para uma dada palavra, divididos por acepções e categorias gramaticais, permitindo uma escolha lexical alargada.

Sempre que possível, os sinónimos são apresentados com a flexão correcta em relação à palavra pesquisada, permitindo a sua substituição imediata (Figura 22).

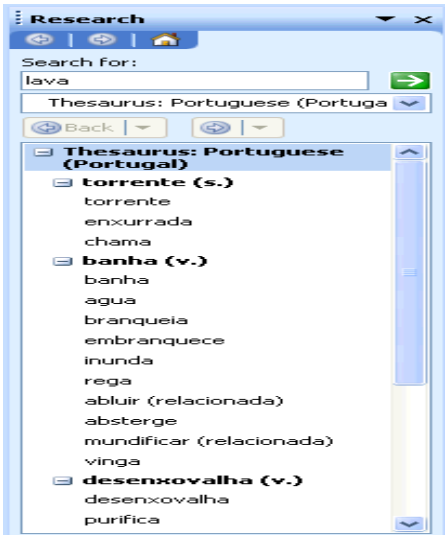


Figura 22 - Dicionário de sinónimos.

2.2.4.9 *Dicionários Temáticos*

O FLiP contém nove dicionários temáticos que podem ser activados individualmente e agrupam muitos domínios do conhecimento, permitindo a rápida verificação de textos de áreas específicas, como se pode ver na Figura 23.

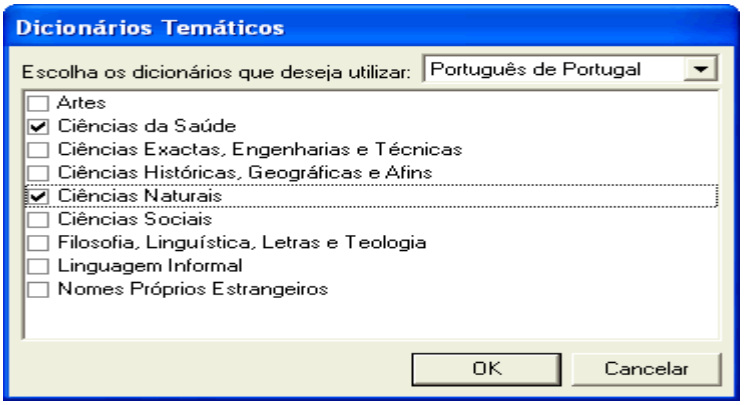


Figura 23 - Selecção de dicionários temáticos.

Ao activar um ou mais destes dicionários, o corrector ortográfico passa a incluir as palavras próprias dessa área, verificando-as e apresentando sugestões sempre que houver incorrecções, conforme se pode ver na Figura 24.

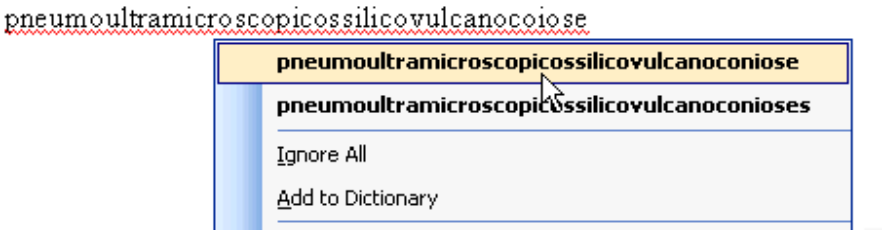


Figura 24 - Apresentação de sugestões para palavras incorrectas

Mostramos em seguida, a lista de dicionários e de alguns dos domínios por eles abrangidos. Em itálico figuram algumas palavras incluídas em cada um dos dicionários Essas palavras serão reconhecidas pelo corrector ortográfico sempre que esse dicionário estiver seleccionado. Entre parêntesis indicamos outros dicionários temáticos referentes a domínios específicos.

a. Artes

Arquitectura, Fotografia, Música, Teatro, Tauromaquia, Tipografia

*Ex***.:** *apogiatura, vermiculura***.**

b. Ciências da Saúde

Medicina (C. Naturais), Anatomia, Cirurgia, Farmácia, Fisiologia (C. Naturais), Veterinária, Psicologia (C. Sociais), Psicanálise (C. Sociais), Desporto.

Ex.: dacriocistite, patognomónica.

c. Ciências Exactas, Engenharias e Técnicas

Matemática, Aritmética, Geometria, Trigonometria, Estatística, Física, Química, Mecânica, Óptica, Informática, Materiais, Electricidade/Electrotecnia, Electrónica, Civil, Metalurgia, Aeronáutica, Náutica, Tipografia (Artes), Metrologia, Lógica (Filosofia).

Ex.: acetilsalicílico, benzenossulfónico, hodógrafo, FrontPage, iWork, ROM.

d. Ciências Históricas, Geográficas e afins

Geografia, Geometria (C. Exactas), História, Antropologia, Arqueologia, Heráldica.

Ex.: filelénico, nesografia.

e. Ciências Naturais

Astronomia, Topografia, Geologia, Petrologia/Petrografia, Meteorologia, Paleontologia, Biologia (C. Médicas), Mineralogia, Cristalografia, Fisiologia (C. Médicas), Botânica, Zoologia, Bioquímica, Ictiologia, Ornitologia, Histologia, Citologia.

Ex.: acotilédone, ciprinídeos, merostomáceos.

f. Ciências Sociais

Sociologia, Política, Direito, Comércio, Etnografia, Economia, Finanças, Antropologia (C. Históricas), Arqueologia (C. Históricas), Psicologia (C. Médicas), Psicanálise (C. Médicas), Filosofia (Filosofia), Heortonímia, Pedagogia.

Ex.: fidejussória, duopólio, pedonomia.

g. Filosofia, Linguística, Letras e Teologia

Linguística, Gramática, Filosofia (C. Sociais), Lógica (C. Exactas), Retórica, Poética, Religião, Bíblia, Mitologia.

Ex.: epanalepse, soteriologia.

h. Linguagem informal

Inclui palavras que ocorrem tipicamente em situações informais e que não são recomendadas para um registo formal ou corrente. Este dicionário engloba palavras pertencentes a registos de língua familiar ou popular, e também palavras consideradas grosseiras ou obscenas.

Ao contrário das palavras incluídas nos outros dicionários temáticos, as palavras deste dicionário nunca são apresentadas nas sugestões.

Ex: merda, puta

i. Nomes Próprios Estrangeiros

Inclui nomes próprios estrangeiros (de lugares e pessoas, por exemplo) usados com alguma frequência.

Ex.: Bruxelas, John.

Os dicionários temáticos, quando seleccionados, são utilizados pelo corrector ortográfico em qualquer aplicação com que este seja compatível.

Selecção dos Dicionários Temáticos

A selecção dos dicionários temáticos pode ser conseguida de dois modos distintos:

- através do configurador;
- a partir dos suplementos existentes para os processadores de texto.

Em qualquer dos casos, basta carregar com o botão esquerdo do rato sobre o quadrado que se encontra à esquerda para marcar ou desmarcar um dicionário, conforme se mostra na Figura 25.

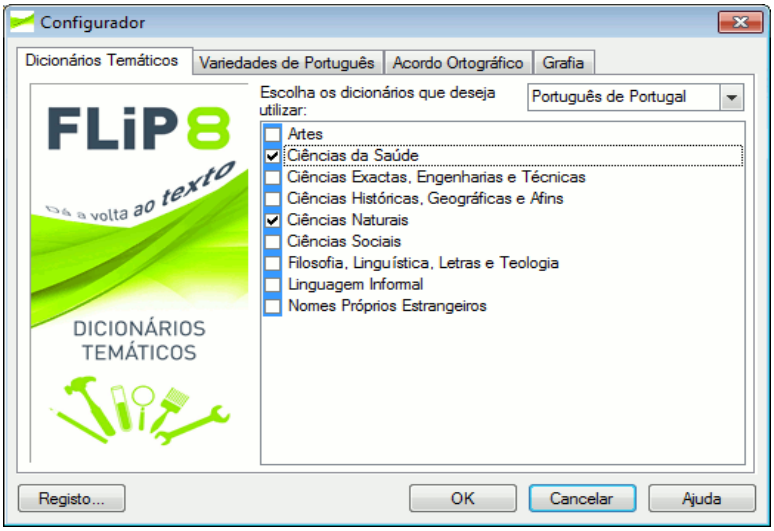


Figura 25 - Diferentes formas de seleccionar dicionários temáticos.

1 Editor

O FLiPEd é um editor de texto compatível com as ferramentas linguísticas descritas para o português. Este programa suporta não só os comandos básicos de edição e formatação (tipo de letra, estilos, listas, tabelas, imagens), mas também a verificação ortográfica e sintáctica automática, a hifenização, a consulta do dicionário de sinónimos (e de antónimos, para o português do Brasil), dos auxiliares de tradução e do conjugador.

O acesso às ferramentas linguísticas é feito através do menu "FLiP" ou do menu de contexto (associado aos cliques com o botão direito do rato sobre as palavras).

A activação da correcção ortográfica segundo o Acordo Ortográfico de 1945 ou de 1990 pode ser feita através do menu "Acordo Ortográfico" do FLiPEd, e aí se poderá aceder também às opções de personalização das grafias. O FLiPEd dispõe também da ferramenta de conversão automática de um texto ou de uma parte seleccionada de um texto para a

ortografia segundo o Acordo Ortográfico de 1990, obtendo-se o resultado em muito menos tempo do que no Word.

Uma característica bastante útil é a possibilidade de invocar os auxiliares de tradução para uma selecção de uma sequência de palavras, sendo apresentadas as diversas traduções em simultâneo. Além disso, o FLiPEd permite a utilização simultânea de um painel para os auxiliares de tradução e de outro para o conjugador.

É compatível com os formatos TXT, RTF, HTML, DOC e DOCX, incluindo ainda a opção de exportação para o formato PDF, satisfazendo assim as necessidades da maioria dosutilizadores em termos de compatibilidade com os formatos de documentos mais utilizados.

Um exemplo do Editor em funcionamento pode ser visto na Figura 26.

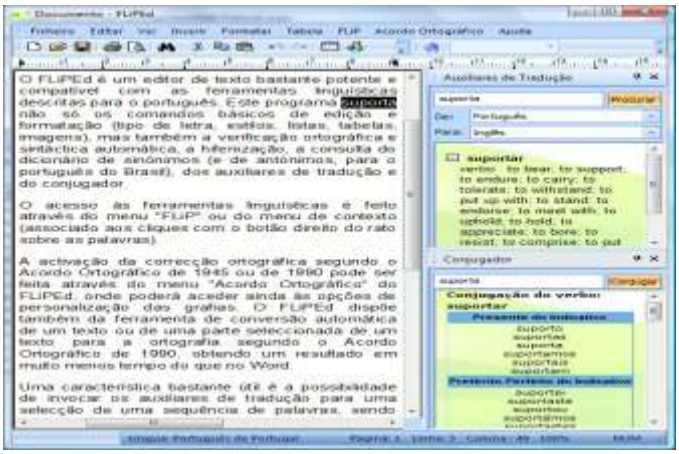


Figura 26 - Editor a funcionar dentro do FliP.

2 Etiquetas LegiX

As etiquetas inteligentes (smart tags) LegiX, incluídas no FLiP 8, permitem a consulta de documentos legais directamente a partir das aplicações do Microsoft Office (XP, 2003, 2007 e 2010).

As etiquetas LegiX detectam a presença de referências a informação jurídica e criam automaticamente ligações para essa informação no LegiX.

Se o LegiX 8 estiver instalado no computador, o documento é imediatamente visualizado. Se o documento não existir no LegiX 8 ou se o LegiX 8 não estiver instalado no computador

e uma ligação à internet estiver disponível, esse documento é também procurado no LegiX.pt.

Apresenta-se em seguida uma ilustração do funcionamento desta ferramenta nas Figs. 27 e 28.

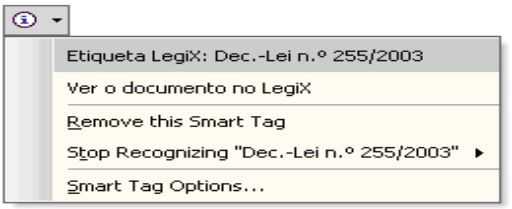


Figura 27 - Acesso à ferramenta LegiX do FliP.

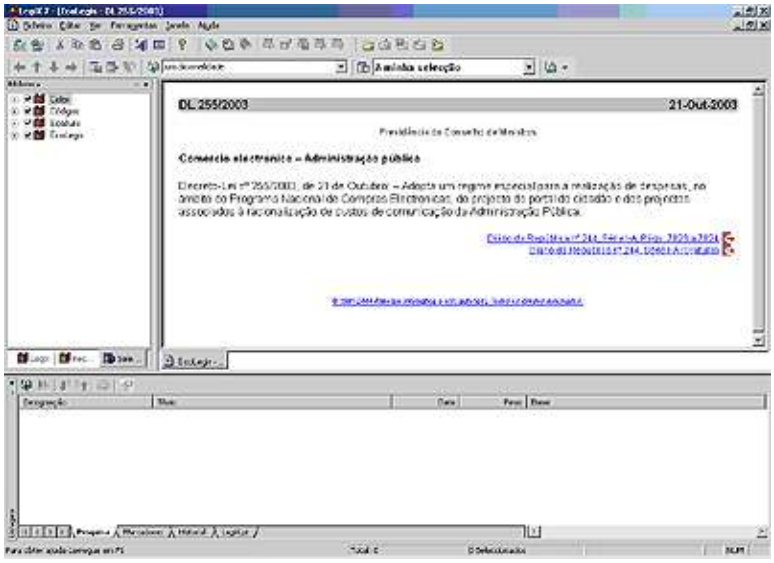


Figura 28 - Ferramenta LegiX em funcionamento.

3 Hifenizador

O hifenizador (ou translineador) executa a divisão automática das palavras no final de cada linha (translineação), de acordo com as regras ortográficas da língua portuguesa, como ilustra a Figura 29.

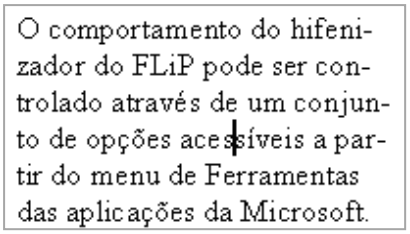


Figura 29 - Divisão automática de palavras no final de cada linha.

A hifenização deve ser sobretudo utilizada em documentos em que o texto se apresenta em colunas, para melhorar a sua distribuição.

Os hifenizadores para o português europeu e português do Brasil são diariamente utilizados na paginação de alguns jornais.

O suplemento para o Word XP, 2003, 2007 e 2010 permite o acesso ao conjugador, aos auxiliares de tradução, à selecção dos dicionários temáticos, ao Dicionário Priberam da Língua Portuguesa e às respostas das dúvidas linguísticas on-line. Também através do suplemento para o Word estão disponíveis a activação da ortografia segundo o Acordo Ortográfico de 1990, a personalização das duplas grafias que este acordo permite e a conversão de texto para a nova ortografia.

Figura 30 - Suplemento do FliP para o Word.

O suplemento para o Word permite ainda a localização de flexões e derivações de uma palavra em português.

Esta opção é bastante útil em algumas situações, nomeadamente de pesquisa, mas que apenas está disponível no Word para a língua inglesa. Se, por exemplo, pedir a localização de todas as formas do verbo ser, serão encontradas palavras como é, são ou fomos, que, obviamente, nunca seriam reconhecidas como formas daquele verbo se fosse utilizada a ferramenta de pesquisa do Word, porque não contêm a cadeia de caracteres "ser". Ver Figura 31.

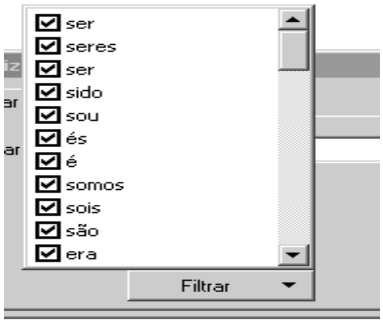


Figura 31 - Flexões do verbo *ser* fornecidas pelo FliP.

Opcionalmente, pode-se também seleccionar exactamente as formas que se quer procurar carregando no botão "Filtrar", como mostra a Figura 32.

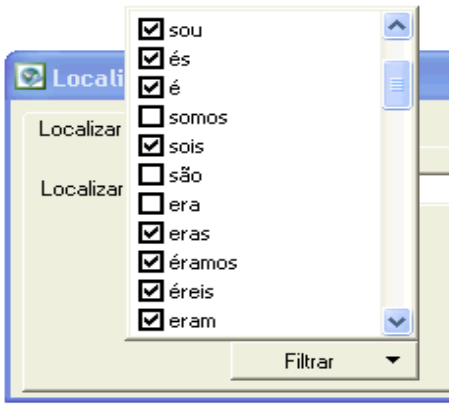


Figura 32 - Conjugação do verbo *ser* pelo FliP.

4 Variedades do Português

Neste módulo estão incluídos léxicos de variedades diferentes do português de Portugal e do português do Brasil, nomeadamente para o português de Angola, Cabo Verde, Galiza, Guiné-Bissau, Moçambique, São Tomé e Príncipe, Macau e Timor.

Os léxicos das variedades do português de Angola, Cabo Verde, Guiné-Bissau, Moçambique, São Tomé e Príncipe, Macau e Timor foram recolhidos pela Priberam a partir de dicionários, de vocabulários, de pequenos corpora e de pesquisas na Internet.

O léxico da variedade do português da Galiza baseia-se na terceira versão do Léxico da Galiza para ser integrado no Vocabulário Ortográfico Comum da Língua Portuguesa,

elaborado pela Comissão de Lexicologia e Lexicografia da Academia Galega da Língua Portuguesa.

Estes léxicos incluem vocabulário comum e nomes próprios, nomeadamente topónimos[20], antropónimos[21] e sociónimos[22].

Em qualquer dos casos, trata-se de léxicos não exaustivos que a Priberam pretende aumentar e melhorar futuramente.

5 Wordbreaker e Stemmer

Tanto o processo de indexação da informação como o processo de interrogação da mesma são baseados em três módulos que se apresentam em seguida:

- *Wordbreaker* – tem por função separar um texto em palavras e expressões, sendo este o primeiro passo no processo de indexação dos textos. É também este módulo que realiza a normalização das diferentes representações que os números, datas e horas podem ter para um formato único sobre o qual seja possível efectuar operações de pesquisa. Compete ainda a este módulo o tratamento da acentuação (ou falta dela), das abreviaturas, das siglas e acrónimos e das palavras compostas por hifenização;
- *Stemmer* – fornece todas as palavras que têm um mesmo radical. É este módulo que permite, por exemplo, que uma pesquisa por "expropriação" recupere documentos que incluam as palavras "expropriações" ou "expropriado". Particularmente no português, em que existem tantos mecanismos de derivação, a utilização deste componente tem um impacto bastante grande na eficácia das pesquisas;
- *Stop list* - Trata-se de uma lista das palavras mais frequentes do português que são ignoradas no processo de indexação por não acrescentarem qualquer informação útil para a localização de um assunto. São exemplos destas palavras "de" e "que".

[20] Nome próprio de um lugar, sítio ou povoação (ex.: *Lisboa, Brasil, Nampula*).

[21] Nome próprio de pessoa (ex.: *António, Francisco, Pedro, Fernandes, Mendes*).

[22] Nome próprio de empresa, entidade ou organização.

2.2.5 Gramática electrónica

Pacote de ferramentas constituído pelos seguintes componentes:

- Corrector ortográfico
- Análise gramatical
- Filtros de estilos
- Verificação mecânica
- Minigramática
- Dúvidas da língua portuguesa

2.2.6 Ispell

Ferramenta disponibilizada pelo Instituto de Matemática e Estatística da Universidade de São Paulo[23] e que tem como objectivo auxiliar a correcção ortográfica de um documento. Quando é encontrada uma palavra que não existe no dicionário, o Ispell tenta encontrar o termo mais próximo da palavra em causa.

2.2.7 Jspell

Analisador morfológico desenvolvido pela Universidade do Minho no âmbito do projecto Natura[24]. Dada uma palavra, este analisador é capaz de indicar as suas características morfológicas (género, número, categoria gramatical).

De um modo geral, um analisador morfológico é constituído por duas componentes:

- dicionário base (palavras e suas características morfológicas)*;*
- conjunto de regras de formação de novas palavras.

[23] http://www.ime.usp.br/~ueda/br.ispell/

[24] http://natura.di.uminho.pt/wiki/doku.php?id=ferramentas:jspell

Estrutura do dicionário

Cada entrada do dicionário do Jspell corresponde a um lema, uma descrição morfológica, um conjunto de regras de flexão e derivação conforme ilustrado no exemplo 2.1 que se encontra no Anexo I.

Estrutura das regras de afixos

Os identificadores das regras de derivação ou flexão são definidos num ficheiro em separado. As regras definidas neste ficheiro explicam como se formam novas palavras a partir do lema. Por exemplo, a formação de plurais segundo a regra 'p' efectua-se do seguinte modo:

flag *p[25]

[^Ã] [^LSMRNZX]	>	S;	"N=p"	# -> s	ex: pato, patos
Ã E	>	S;	"N=p"	# ãe -> ães	ex: mãe, mães
Ã O	>	-ÃO,ÕES;	"N=p"	# ão -> ões	ex: leão, leões
A L	>	-L,IS;	"N=p"	# al -> ais	ex:animal,animais
O L	>	-OL,OIS;	"N=p"	# ol -> óis	ex:anzol, anzóis
[^V] E L	>	-EL,EIS;	"N=p"	# el -> éis	ex: papel, papeis
	...				

Tabela 1 - Tabela de regras de formação de plurais.

Funcionalidade do Jspell

O Jspell pode ser utilizado tanto como corrector ortográfico, como biblioteca de linguagens de programação como C, Prolog ou Perl.

Pode ainda ser utilizado de forma interactiva com o utilizador como se pode ver nos exemplos 2.2, 2.3, 2.4 e 2.5, que podem ser consultados no Anexo I.

[25] http://alfarrabio.di.uminho.pt/~albie/publications/jspell.pm.pdf

2.2.8 Lince

Corrector ortográfico para o português europeu.

2.2.9 Redacção Língua Portuguesa

Ferramenta concebida para edição e correção ortográfica de textos escritos em português.

2.2.10 ReGra

Analisador gramatical para o Português do Brasil disponibilizado pelo NILC[26]. É composto por um léxico de 1,5 milhões de palavras incluíndo nomes próprios, acrónimos e abreviaturas.

O seu principal objectivo consiste em fornecer pontuação legível para textos escritos em Português do Brasil.

2.2.11 WebJspell

Ferramenta desenvolvida pela Linguateca em conjunto com a Universidade do Minho[27], tem como objectivo tornar o Jspell acessível ao máximo possível de utilizadores. Pode-se observar um exemplo de utilização no exemplo 2.6 que pode ser consultado no Anexo I.

[26] http://www.nilc.icmc.usp.br/nilc/projects/regra.htm

[27] http://natura.di.uminho.pt/webjspell/jsol.pl

2.3 ALINHADORES

2.3.1 Alinhador online CEPRIL

Ferramenta desenvolvida pelo LAEL [28] e disponível online em http://www2.lael.pucsp.br/corpora/alinhador/, permite identificar automaticamente correspondências entre excertos de textos originais e traduzidos.

O exemplo 2.7 (Anexo II) ilustra a utilização desta ferramenta .

2.3.2 MTTK

MTTK – Machine Translation ToolKit – é uma colecção de ferramentas de software desenvolvidas no Centro da Linguagem e da Fala da Universidade Johns Hopkins[29] que têm como objectivos:

- o alinhamento de documentos ao nível da frase, ou da sub-frase, tarefa conhecida como *chunking* (passo útil de pre-processamento com vista a preparar colecções de traduções para utilização na estimativa de parâmetros de modelos de alinhamento complexos.

  O alinhamento ao nível da sub-frase torna possível a segmentação de frases longas em segmentos mais pequenos, alinhados, que, de outra forma, seriam desprezados;

- treino de modelos estatísticos para o alinhamento de textos paralelos. Os seguintes modelos são suportados pelo MTTK:

  - IBM Model-1 e Model-2;
  - Modelos Ocultos de Markov entre palavras - Word-to-Word HMM's;
  - Modelos Ocultos de Markov entre palavras e frases - Word-to-Phrase HMM's, com probabilidades de tradução de bigramas.

[28] http://www4.pucsp.br/pos/lael/

[29] http://www.clsp.jhu.edu/

- paralelização dos procedimentos de treino de modelos – se dispusermos de máquinas com múltiplos processadores podemos dividir os textos de treino em subconjuntos mais pequenos tendo a possibilidade, deste modo, de acelerar os procedimentos de re-estimativa de parâmetros e reduzindo a quantidade de memória necessária ao treino. Isto é conseguido pelos procedimentos de estimativa de parâmetros baseados no algoritmo EM[30];
- geração de alinhamentos entre textos paralelos utilizando os processos palavra-palavra (Word-to-Word) e palavra-frase (Word-to-Phrase);
- extracção das tabelas de tradução palavra-palavra (word-to-word) a partir de bitext alinhados e de modelos estimados;
- extraccção das tabelas de tradução frase-a-frase (inventários phrase-pair) a partir de textos paralelos alinhados;
- utilização dos modelos de alinhamento dos modelos ocultos de Markov (HMM) para a indução de traduções frásicas a partir dos seus modelos estatísticos – a indução Phrase-pair pode gerar inventários mais ricos de traduções de frases do que aqueles que são extraídos a partir de alinhamentos *Viterbi*;
- edição do código C++ com vista à implementação dos próprios procedimentos de estimativa e alinhamento.

2.3.3 NATools

"*Package*" de ferramentas desenvolvido na Universidade do Minho[31] e que é utilizado no processamento de *corpora* paralelos, disponível sob licença GPL. Neste "*package*" estão incluídas as seguintes ferramentas:

2.3.3.1 *nat-shell*

Interface de shell utilizada para alinhamento de *corpora*;

[30] Algoritmo Expectation-Maximization

[31] http://linguateca.di.uminho.pt/natools/

2.3.3.2 *nat-sentence-align*

Interface para o alinhador Vanilla;

2.3.3.3 *nat-sentalign*

Alinhador de declarações escritas na linguagem C;

2.3.3.4 *nat-create*

Ferramenta que cria um corpus a partir de um par de ficheiros de texto ou de um ficheiro TMX;

2.3.3.5 *nat-pre*

Pre-processador para textos paralelos, contagem de palavras, verificação de números de frases;

2.3.3.6 *nat-initmat*

Ferramenta que inicializa uma matriz esparsa com co-ocorrências de palavras;

2.3.3.7 *nat-ipfp*

Implementação iterativa do algoritmo EM;

2.3.3.8 *nat-samplea*

Implementação iterativa do algoritmo EM;

2.3.3.9 *nat-sampleb*

Implementação iterativa do algoritmo EM;

2.3.3.10 *nat-mat2dic*

Ferramenta que executa a tradução de matrizes de co-ocorrências para um ficheiro com o dicionário;

2.3.3.11 *nat-postbin*

Ferramenta que executa a tradução do ficheiro com o dicionário para um formato legível pela linguagem de programação Perl.

Formatos de ficheiros

As NATools suportam dois tipos de ficheiros:

- TMX tem duas partes:
    - uma especificação do formato do contentor (os elementos de nível mais elevado fornecem informação acerca do ficheiro como um todo e acerca das entradas). Neste formato, uma entrada consistindo em segmentos de texto alinhados em duas ou mais linguas é chamada Unidade de Tradução (representada pelo símbolo <tu>).
    - uma especificação de um formato de meta-markup de baixo-nível para o conteúdo de um segmento de texto da memória de tradução. Neste formato, um segmento de texto da memória de tradução é denotado pela *tag* <seg>.

- formato específico – utiliza dois ficheiro (um para cada língua), cada unidade de tradução é separada das restantes por uma linha com o símbolo '$' e pode ocupar mais de uma linha como se pode ver no exemplo seguinte.

I saw a cat .	Eu vi um
$	gato .
The cat was	$
fat .	O gato era gordo .
$	$

Tabela 2 - Formato de um ficheiro para análise pelas NATools.

Ambos os textos devem ter o mesmo número de unidades de tradução e estar já 'tokenizados'.

Bootstrapping a partir de um par de ficheiros NATools

Quando se utiliza este método, é necessário ter um par de ficheiros alinhados ao nível da frase, no formato anteriormente especificado. Nos exemplos seguintes os ficheiros serão denominados 'lang1' e 'lang2' respectivamente.

Para proceder directamente ao alinhamento dos ficheiros utiliza-se o processo de identificação da linguagem como se segue:

```
[foo@bar]$  nat-create lang1 lang2
```

É igualmente possível especificar as línguas, no caso de se querer acelerar o processo, ou no caso de o processo de identificação da língua não identidicar correctamente as línguas dos ficheiros a analisar:

```
[foo@bar]$  nat-create -langs=PT..EN lang1 lang2
```

a opção '-langs' especifica as linguagens em causa na mesma ordem dos ficheiros especificados (neste caso, 'lang1' deveria ser 'Portuguese' e 'lang2' deveria ser 'English').

Ambos os métodos pedirão o nome de um *corpus.* O *script* cria em seguida seguida um directório com o nome especificado no qual serão armazenados os dados probabilísticos de tradução e colocados os ficheiros com o *corpus* codificado, léxico codificado e dicionários.

Criação de um ficheiro com o dicionário probabilístico de tradução

Em alguns casos poderá ser útil olhar para o dicionário probabilístico de tradução (PTD) extraído do corpus paralelo sem utilizar o servidor das NATools. Para alcançar este objectivo, podemos extraír o PTD para um ficheiro de texto (em formato Perl Data::Dumper que é legível tanto pelo homem como pelo computador).

Para levar a cabo esta tarefa deverá, em primeiro lugar, mudar-se o directório corrente para aquele que foi criado pelo processo de codificação do corpus, e executar o seguinte comando:

```
[foo@bar]$ nat-dumpDicts source.lex source-target.bin target.lex target-source.bin > dict.txt
```

O ficheiro dict.txt será criado com o PTD.

Utilização do nat-server

O servidor necessita de um ficheiro de configuração muito simples. As linhas que iniciam com o símbolo '#' são consideradas comentários, e desta forma, são ignoradas. As restantes deverão conter os caminhos absolutos para os directórios criados pelo comando nat-create (ou nat-shell). Por exemplo, se ao executar o comando 'nat-create' foi criado um *corpus* no directório /corpora/parallel com o nome EuroPar1, deverá adicionar-se a seguinte linha ao ficheiro de configuração:

```
/corpora/parallel/EuroParl
```

O servidor configurará cada *corpus* baseado no ficheiro de configuração 'nat.cnf', presente em cada uma das directorias daquele *corpus*.

Para iniciar o servidor, utiliza-se o seguinte comando:

```
[foo@bar]$  nat-server /path/to/the/config/file.cfg
```

2.3.4 VisualLIHLA

Ferramenta desenvolvida pelo NILC e disponível online em http://www.nilc.icmc.usp.br/nilc/projects/visuallihla.htm, baseada no alinhador lexical LIHLA. O LIHLA alinha *tokens*, palavras e unidades multipalavra baseadas em heurísticas independentes da linguagem, tais como a posição e constrói automaticamente recursos dependentes da língua (dicionários bilingues).

A precisão dos alinhamentos produzidos pelo LIHLA varia entre 84% e 92%.

No exemplo 2.8, que pode ser consultado no Anexo II, é mostrada uma execução do alinhamento efectuado pelo LIHLA entre dois textos.

2.3.5 VisualTCA

Ferramenta visual desenvolvida pelo NILC e disponível online em http://www.nilc.icmc.usp.br/nilc/tools/pagina-visualtca/visualtca.htm, que tem como objectivo efectuar o alinhamento de um par de textos paralelos ao nível da frase.

O exemplo 2.9. (Anexo II) ilustra o funcionamento desta ferramenta.

2.4 ANALISADORES

2.4.1 Concordanciador de um milhão de palavras

O concordanciador, desenvolvido pelo LAEL [32] e disponível online em http://www2.lael.pucsp.br/corpora/bp/conc é um *corpus* constituído por um milhão de palavras.

No Anexo III pode ser consultado um exemplo de utilização desta ferramenta (exemplo 2.10).

2.4.2 Curupira

Parser[33] autónomo relativamente à sua aplicação. O principal objectivo desta ferramenta, desenvolvida pelo NILC [34] e disponível online em http://www.nilc.icmc.usp.br/nilc/tools/curupira.html, consiste em fornecer, para uma dada frase introduzida como *input*, todas as suas possibilidades de análise sintáctica das palavras nela contidas, idenpendentemente do seu significado.

A arquitectura desta aplicação é mostrada na Figura 33 e explicada em seguida.

Figura 33 - Arquitectura do *parse*r Curupira.

[32] http://www2.lael.pucsp.br/

[33] Programa que subdivide uma entrada (*input*) para que um outro possa atuar sobre ela; analisador gramatical

[34] http://www.nilc.icmc.usp.br/nilc/

2.4.2.1 *Descrição do parser*

2.4.2.1.1 Dicionário

O CURUPIRA opera sobre uma base de dados lexical robusta, na qual estão representados apenas os termos simples e compostos da língua portuguesa. Esta base de dados não contém morfemas (lexicais ou gramaticais), nem expressões complexas que contenham espaços em branco.

O léxico contém cerca de um milhão e meio de entradas em formato texto plano, tendo sido obtido a partir do processamento automático do *corpus* do NILC, que contém cerca de 40 milhões de palavras, e é constituído por textos de género predominantemente jornalístico (embora contenha em menor grau, textos literários, textos técnicos e redacções escolares).

Após o processo de composição inicial, no qual as entradas foram classificadas manualmente, outras entradas foram acrescentadas, por flexão ou derivação automática (caso das formas verbais e dos advérbios de modo, por exemplo), além de outras palavras, obtidas nas várias sessões de testes a que o dicionário foi submetido. As entradas do dicionário possuem o seguinte formato:

```
cantar=<V.[BI.INT.TD.][FUT-SUBJ.ELE.FUT-SUBJ.EU.INF-PESS.ELE.INF
        PESS.EU]N.[a][cantar]0.>
```

Como se pode verificar pelo exemplo anterior, o léxico contém apenas informação de natureza morfossintática.

Em cada entrada estão representados:

- a classe das palavras (substantivo, adjectivo, numeral, nome próprio, abreviatura, sigla, prefixo, interjeição, conjunção, preposição, artigo, advérbio, verbo, pronome);
- as subclassificações pertinentes a cada classe (pronome pessoal, possessivo, demonstrativo, indefinido, interrogativo, de tratamento, em relação aos pronomes, por exemplo; ou numeral multiplicativo, cardinal, ordinal e fracionário relativamente aos numerais);
- o género, sempre que seja pertinente (masculino, feminino, ou uniforme);
- o número, sempre que se justifique (singular, plural ou invariável);

- o tempo e o modo, para as formas verbais (presente do indicativo, futuro do conjuntivo, etc.);
- o número e a pessoa, para as formas verbais e para os pronomes pessoais (primeira pessoa do singular, segunda pessoa do plural, etc.);
- o grau, sempre que pertinente (positivo, aumentativo, diminutivo);
- a transitividade, para os verbos (transitivo direto, intransitivo, etc.);
- a regência, para os substantivos, adjetivos, verbos e advérbios que exigem complemento preposicionado;
- o tipo, para os advérbios (de tempo, de modo, de intensidade, etc.);
- a forma canónica (ou a forma de citação da palavra, pela qual ela está relacionada às outras entradas do dicionário), para todas as formas.

Na medida em que nenhuma informação de natureza semântica está representada no dicionário, não são registadas diferenças entre diferentes significados de uma mesma entrada, nem mesmo quando correspondem a formas canónicas diferentes ou a diferentes classes gramaticais.

Nos casos de homonímia[35,] as diferentes classificações possíveis para cada verbete são ordenadas, na mesma entrada, segundo a conveniência de análise, que está normalmente amparada na frequência de ocorrência para o falante nativo da língua.

A desambiguação categorial é realizada em três modos distintos:

a) pela frequência de ocorrência indicada;
b) pelo próprio conjunto de possibilidades sintáticas previstas pela gramática;
c) por um conjunto muito restrito de regras de desambiguação incorporadas na ferramenta.

[35] Designação geral para os casos em que palavras de sentidos diferentes têm a mesma grafia (os homónimos homógrafos) ou a mesma pronúncia (os homónimos homófonos).

2.4.2.1.2 Módulo de compactação e acesso ao dicionário (KLS[36])

Sistema desenvolvido por Kowaltowski, Lucchesi e Stolfi que permite que a representação do léxico do NILC - cerca de um milhão e meio de entradas - e ainda os atributos seja codificado num autómato que ocupa menos de 1,3 Mb. Isto é devido, principalmente, ao elevado número de formas derivadas na língua portuguesa. Por exemplo, um verbo regular do português apresenta cinquenta e uma flexões distintas cujos sufixos podem ser compartilhados entre todos os outros verbos regulares.

O algoritmo utilizado parte de um estado inicial, percorre o autómato utilizando letras consecutivas da palavra para seleccionar as transições, até que um estado final seja alcançado ou não existam mais transições válidas.

2.4.2.1.3 Gramática

A gramática que serve de suporte ao CURUPIRA é definida como <FRASE, V, t, R, P> onde:

- FRASE é o símbolo inicial (qualquer intervalo entre dois delimitadores de frase);
- V corresponde ao vocabulário não-terminal (composto por um conjunto de etiquetas sintácticas);
- t corresponde ao vocabulário terminal, ou conjunto de traços categoriais presentes no dicionário;
- R é um conjunto de aproximadamente 600 regras de reescrita categorial, conforme sintaxe definida em seguida;
- P é a prioridade de aplicação das regras de reescrita, atribuída ora pela freqüência de ocorrência, ora pela conveniência de análise.

A gramática corresponde a um arquivo de texto plano que, antes de ser submetido ao CURUPIRA, passa pelo processo de compilação descrito em seguida. A sintaxe das regras de re-escrita categorial é a que se segue:

[36] Iniciais de Kowaltowski, Lucchesi e Stolfi

A#B#C onde:

- A = símbolo;
- B = prioridade de aplicação da regra: número inteiro igual ou superior a 1 (1 = prioridade máxima);
- C = regras de reescrita (conjunto de possibilidades sintácticas de realização do símbolo expresso por A);
- + = justaposição, com separação por espaço em branco (apenas);
- [X] = opcionalidade (x é opcional);
- X(Y) = Y deve ser atributo de X;
- (i) = indexado (os termos portadores de (i) devem concordar em número, pessoa e género);
- {X,Y} = exclusividade: ou X, ou Y;
- 'entre aspas simples' = entradas do dicionário;
- <X> = forma canónica.

Um exemplo de regra de reescrita categorial é apresentado a seguir:

SUJ_SIMPLES#5#AADNE + [APOSTO] + nucleo(SUJ2) + [AADND] + [APOSUJ].

2.4.2.1.4 Compilador

O compilador é um programa implementado em Visual C++.Net para a plataforma Windows que converte um ficheiro de texto plano contendo as regras de reescrita categorial em vários outros arquivos que englobam classes de objectos em linguagem de programação C++.

Uma classe de objectos é formada pela união de todas as regras de reescrita categorial que possuam o mesmo símbolo expresso por A. As regras de re-escrita expressas por C são convertidas em métodos da classe e controladas por um gestor de regras que aplica a prioridade dada por B. Este gestor de regras também é responsável por parte do controle

do mecanismo de *backtracking* e é responsável pela aplicação de alguns mecanismos de optimização.

Por exemplo, se depois de o gestor aplicar uma determinada regra em certa posição da frase e a regra não for válida para esta posição, ele irá desabilitar esta regra para esta posição da frase. Este mecanismo permite que regras não candidatas sejam disparadas apenas uma única vez por análise.

Uma das características aproveitadas pelo compilador é o mecanismo de herança. Assim, a classe-base inclui os métodos que todas as classes derivadas subsequentes devem possuir em comum (ou seja, quase todos os métodos com os mecanismos de gerenciamento das regras); e as classes derivadas incluem principalmente os métodos que correspondem às regras de reescrita categorial. A estrutura hierárquica referida é apresentada a seguir:

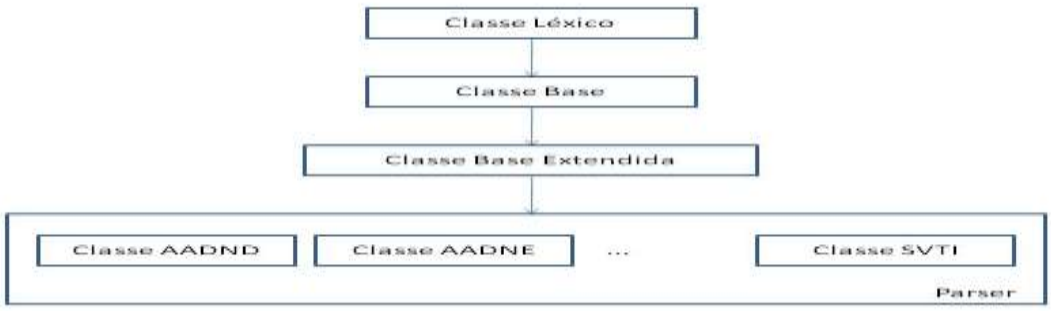


Figura 34 - Estrutura hierárquica do compilador do *parser* Curupira

A classe **Léxico** é uma interface responsável pela interacção entre o dicionário com informações lexicais e as regras de reescrita categorial. A classe **Base** contém os principais mecanismos de gestão das regras de reescrita categorial. Finalmente, a classe Base Estendida é criada pelo compilador e deve conter as peculiaridades do parser. Por exemplo, as regras com vocábulos terminais não são separadas em classes distintas de objectos, mas acrescentadas como métodos da classe **Base Estendida**.

Tanto o código fonte gerado pelo compilador, como o código fonte da arquitetura básica herdado do ReGra, são incorporados num projecto em Visual C++ que ao ser executado gera uma DLL.

Posteriormente, esta DLL e o dicionário compactado podem ser carregados por um programa executável que contém uma interface para chamar o *parser* e visualizar o resultado.

2.4.2.1.5 Interface

O CURUPIRA interpreta como "frase" qualquer intervalo de palavras entre um conjunto especificado de delimitadores sentenciais, a saber:

- sinais de pontuação (o ponto, o ponto-e-vírgula, os dois-pontos, as reticências, o ponto-de-interrogação, o ponto-de-exclamação e o travessão);
- caracteres de comando (marcadores de início-de-linha, de fim-de linha, de início-de-parágrafo, de fim-de-parágrafo, de início-de-coluna, de fim-de-coluna, de início-de-página e de fim-de-página).

Os textos de entrada devem estar convertidos no conjunto de caracteres alfanuméricos do padrão ANSI, sendo também admitida a utilização de alguns caracteres textuais mais comuns (vírgula, hífen, maior-que, menor-que, etc.). O formato da saída do CURUPIRA é caracterizado por uma estrutura arbórea, representada de forma identada baseada no formalismo da parentetização etiquetada.

As etiquetas sintáticas são assinaladas aos sintagmas, nos seus vários níveis, com a indicação da função sintática que as palavras e as expressões assumiriam no contexto da frase.

O conjunto das funções sintáticas, devido à aplicação ReGra - aplicativo do qual a ferramenta provém - acompanha o conjunto de rótulos previstos pela Nomenclatura Gramatical Brasileira, com algumas modificações.

2.4.3 GojolParser

Ferramenta desenvolvida por Vlad Gojol [37] e disponível online em http://www.linguateca.pt/Repositorio/GojolParser.txt. Esta ferramenta efectua a análise morfo-sintáctica em profundidade de um *corpus*. No exemplo 2.11, que pode ser consultado no Anexo III é mostrada a utilização deste *parser*.

2.4.4 Lexificador DeepDict

Ferramenta disponível online em http://gramtrans.com/deepdict/ que permite construir entradas de dicionário complexas e súmulas de contexto para uma dada palavra. As relações das palavras são baseadas na análise de dependências da gramática de restrições e funções gramaticais, e não apenas em co-ocorrências. Para cada relação são dados os valores das frequências absoluta e relativa.

Exemplo de utilização desta ferramenta introduzindo a palavra 'computador' e seleccionando a língua portuguesa, limite de frequência 'high'e atribuindo à ocorrência mínima o valor '2' obtém-se o resultado ilustrado na Figura 35.

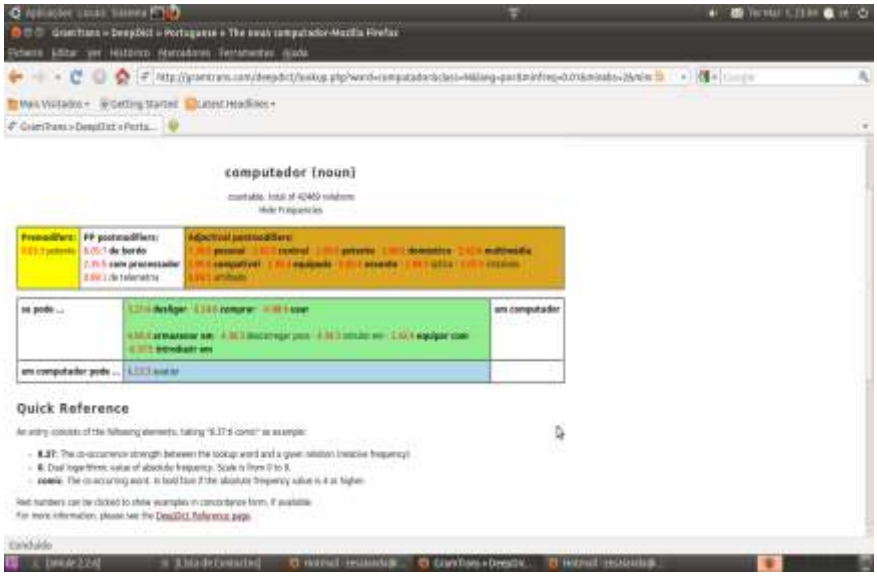


Figura 35 - Execução do lexificador DeepDict.

[37] http://www.nume-persoane.ro/gojol-vlad-714005.htm

2.4.5 Lingua::PT::PLNBase

Extensão escrita por Alberto Manuel Brandão Simões[38] na linguagem de programação Perl, disponível sob licença GPL, que inclui vários módulos úteis ao processamento computacional da língua portuguesa, tais como:

2.4.5.1 *atomizadores – neste módulo estão incluídas as funções:*

2.4.5.1.1 atomos

2.4.5.1.2 atomiza

2.4.5.1.3 tokenize

Função que utiliza um algorítmo desenvolvido no Projecto Natura e retorna o texto tokenizado (um token por linha) como se mostra no exemplo 2.12, que pode ser consultado no Anexo III;

2.4.5.1.4 tokeniza

Função que retorna um átomo por linha em contexto escalar, e uma lista de átomos em contexto de lista como se observa no exemplo 2.12, que pode ser consultado no Anexo III;

2.4.5.1.5 cqptokens

Função que retorna um átomo por linha de acordo com a notação CWB. O separador de frases (ou de registo) pode ser modificado utilizando a opção 'irs' como se mostra nos exemplos 2.13 e 2.14 (Anexo III).

[38] http://search.cpan.org/~ambs/Lingua-PT-PLNbase-0.24/

2.4.5.2 *Segmentadores*

2.4.5.2.1 frases;

A utilização desta função é ilustrada no exemplo 2.15 do Anexo III.

2.4.5.2.2 sentences;

2.4.5.2.3 separa_frases

Função que retorna um texto com uma frase por linha como se mostra no exemplo 2.16, Anexo III;

2.4.5.2.4 xmlsentences

Ferramenta que utiliza o método frases e aplica uma etiqueta XML a cada frase. Por omissão, as frases são ladeadas por '<s>' e '</s>'. O nome da etiqueta pode ser substituído usando o parâmetro opcional st. A utilização desta função pode ser observada no exemplo 2.17, Anexo III.

2.4.5.3 *segmentadores a vários níveis*

2.4.5.3.1 fsentences

Função que permite segmentar um conjunto de ficheiros a vários níveis:

- por ficheiro;
- por parágrafo;
- por frase.

O *output* pode ser conseguido em vários formatos obtendo-se ou não numeração de segmentos.

A invocação desta função é efectuda utilizando uma tabela de hash de configuração e uma lista de ficheiros a processar.

Se a lista estiver vazia a função recebe os dados a partir do STDIN[39].

O resultado do processamento é enviado para o STDOUT[40] a não ser que a chave output do hash de configuração esteja definida. Nesse caso, o seu valor será usado como ficheiro de resultado.

A chave input_p_sep permite definir o separador de parágrafos. Por omissão, é usada uma linha em branco.

A chave o_format define os modos de etiquetação do resultado. De momento, a única política disponível é a XML.

As chaves s_tag, p_tag e t_tag definem as etiquetas a usar, na política XML, para etiquetar frases, parágrafos e textos (ficheiros), respectivamente. Por omissão, as etiquetas usadas são s, p e text.

As etiquetas podem ser numeradas, definindo as chaves s_num, p_num ou t_num do seguinte modo:

- '0' - Nenhuma numeração;
- 'f' - Só pode ser utilizado com o t_tag, e define que as etiquetas que delimitam ficheiros usarão o nome do ficheiro como identificador;
- '1'- Numeração a um nível. Cada etiqueta terá um contador diferente;
- '2' - Só pode ser usado com o p_tag ou o s_tag e obriga à numeração a dois níveis (N.N);
- '3' - Só pode ser usado com o s_tag e obriga à numeração a três níveis (N.N.N).

No exemplo 2.18 - Anexo III - é mostrada a utilização de etiquetas neste módulo.

[39] Entrada padrão que por omissão corresponde ao teclado.

[40] Saída padrão que por omissão corresponde ao monitor.

2.4.5.4 *funções de acentuação*

2.4.5.4.1 **remove_accents**

Função que remove a acentuação do texto passado como parâmetro;

2.4.5.4.2 has_accents

Função que verifica se o texto passado como parâmetro tem caracteres acentuados.

2.4.5.5 *funções auxiliares*

2.4.5.5.1 recupera_ortografia_certa;

2.4.5.5.2 tratar_pontuacao_interna

2.4.6 LX-Suite

Pacote de ferramentas, desenvolvido pela Universidade de Lisboa[41], disponível online em http://lxsuite.di.fc.ul.pt/, constituído pelos seguintes módulos:

2.4.6.1 *Lematizador Verbal*

Ferramenta que, dada uma forma verbal como *input*, mostra o lema correspondente, como se pode observar no exemplo 2.19 - Anexo III;

[41] http://www.ul.pt/portal/page?_pageid=173,1&_dad=portal&_schema=PORTAL

2.4.6.2 *Conjugador Verbal*

Ferramenta que aceita como *input* um verbo no modo infinitivo e devolve como *output* todas as suas flexões. O exemplo 2.20 ilustra a utilização desta ferramenta e pode ser consultado no Anexo III.

2.4.6.3 *Flexionador Nominal*

Ferramenta que, dados uma forma (atestada ou hipotética) e os traços flexionais pretendidos para o resultado, devolve uma forma flexionada.

Um exemplo de utilização desta ferramenta pode ser encontrado no exemplo 2.21 do Anexo III.

2.4.6.4 *Anotador Categorial*

Ferramenta que, dado um texto como input, classifica cada uma das suas palavras na respectiva categoria gramatical. No exemplo 2.22, que pode ser consultado no Anexo III, é ilustrada a utilização desta ferramenta.

2.4.6.5 *Reconhecedor de Nomes Próprios*

Ferramenta que, dado um texto como input, devolve os nomes próprios nele existentes, como se mostra no exemplo 2.23 que consta no Anexo III.

2.4.6.6 *Navegador Corpus*

Ferramenta que, dada uma palavra ou expressão, mostra os vários contextos em que aquela aparece, como se pode observar no exemplo 2.24 - Anexo III.

2.4.6.7 *Buscador TreeBank*

Ferramenta que procura no corpus do TreeBank a expressão introduzida, como se mostra no exemplo 2.25, Anexo III.

2.4.7 PoS FreeLing

Biblioteca open source de analisadores de lingua natural desenvolvida pelo Grupo ProLNat da Universidade de Santiago de Compostela e disponibilizada sob licença GPL.

Esta biblioteca foi concebida essencialmente como um conjunto de módulos necessários ao desenvolvimento de aplicações de processamento de língua natural, embora contenha um programa útil para pessoas que apenas pretendem ter um analisador textual.

Os módulos incluídos nesta biblioteca são descritos em seguida:

2.4.7.1 *Tokenizer*

Este módulo converte um texto plano[42] num vector de objectos *word*, de acordo com um conjunto de regras de tokenização.

As regras de tokenização são expressões regulares que são comparadas com o início da linha de texto a ser processada. A primeira regra que corresponde ao emparelhamento é utilizada para extrair o *token*. A substring correspondente é eliminada da linha em causa e o processo é repetido até ao momento em que a linha se encontra vazia. A API deste módulo pode ser consultada no Anexo III.

Como se pode observar pela API, no momento da criação do objecto da classe *tokenizer*, ter-se-á de fornecer uma string que representa um ficheiro sem quaisquer formatações.

[42] Texto sem quaisquer recursos de formatação.

Em seguida, o objecto criado permite obter uma lista de objectos do tipo word que correspondem aos *tokens*[43] criados.

Um ficheiro de regras de tokenização é mostrado no exemplo 2.26, Anexo III.

Este ficheiro é dividido em três secções:

- <Macros> ... </Macros> - permite ao utilizador definir macros[44] regexp que serão mais tarde utilizadas pelas regras. Um exemplo de macro pode ver-se no exemplo 2.27 do Anexo III.
- <RegExps> ... </RegExps> - nesta secção são definidas as regras de tokenização. As macros previamente definidas podem ser referenciadas pelos seus nomes colocados entre parêntesis curvos, como se mostra em seguida:

  *ABREVIATIONS1 0 ((\{ALPHA\}+\.)+)(?!\.\.)

  As regras são expressões regulares e são aplicadas pela ordem da sua definição.

  A primeira regra a emparelhar o início de cada linha é aplicada, construindo-se um token, e sendo ignoradas as restantes regras.

  O processo repete-se até que cada linha tenha sido completamente processada.

  O formato de cada regra é definido como se segue:

  - o primeiro campo da regra corresponde ao seu nome. Se este se inicia com um '*', produzir-se-á apenas um token se o emparelhamento for encontrado na lista de abreviações;
  - o segundo campo corresponde à substring com a qual se vão formar os tokens.

    Este valor pode ser igual a zero, caso em que a toda a expressão corresponde ao emparelhamento, ou qualquer número compreendido entre um e nove. Neste último caso será criado um token, para cada substring com comprimento variável entre um e o valor especificado.

[43] Conjunto de caracteres (de um alfabeto, por exemplo) com um significado colectivo.

[44] Abstrações que definem como um padrão de entrada deve ser substituído por um padrão de saída, de acordo com um conjunto de regras.

- <Abbreviations> ... </ Abbreviations> - nesta secção são definidas abreviaturas comuns (uma por linha) que não deverão ser separadas do ponto final que se lhe segue – etc. por exemplo.

  Estas deverão ainda ser colocadas com letra minúscula, ainda que possam aparecer com letra maiúscula no texto a analisar.

2.4.7.2 *Splitter*

Este módulo recebe listas de objectos *word* (produzidos pelo tokenizador ou por qualquer outro meio) e armazena-os até que um limite de frase seja detectado, momento a partir do qual é retornada uma lista de objectos *sentence*.

O *buffer*[45] do *splitter* pode reter apenas parte dos tokens se a lista fornecida não terminar com uma fronteira de frase clara.

A aplicação chamadora pode enviar posteriormente listas de *tokens* a adicionar, ou efectuar o pedido ao *splitter* de limpeza do *buffer*.

A API deste módulo figura no Anexo III.

Pela observação da API, pode concluir-se que no momento da criação do objecto splitter, há a necessidade de lhe fornecer um ficheiro de opções cujo exemplo é apresentado no Anexo III.

Como aí se pode pode observar o ficheiro de opções do *splitter* é constituído por quatro secções:

- <General> - contém as opções gerais para o *splitter*, tais como *AllowBetweenMarkers* e *MaxLines*.

  No exemplo 2.29 - Anexo III - pode ver-se um exemplo de utilização desta secção;

[45] Uma região de memória temporária utilizada para escrita e leitura de dados

- <Markers> - lista os pares de caracteres que têm de ser considerados como marcas 'abrir-fechar'. O exemplo 2.30 que se encontra no Anexo III, ilustra a declaração desta secção;
- <SentenceEnd> - lista dos caracteres considerados como finais de frase.

  Cada um é seguido por um valor binário que indica se se trata de um final de frase ambíguo ou não.

  No exemplo 2.31 – Anexo III - '?' é uma marca inequívoca de frase, sendo incondicionalmente introduzido um split de frase após cada '?'.

  Os outros dois caracteres são ambíguos, sendo que, só será introduzido um *split* de frase se forem seguidos por uma palavra escrita em maiúscula ou por um caracter que indique o início de uma frase. como se verifica no exemplo apresentado no Anexo III.

- <SentenceStart> - nesta secção são declarados os caracteres que se sabe aparecerem apenas no início de uma frase.

  O exemplo 2.32 - Anexo III - mostra a declaração desta secção para o idioma espanhol.

2.4.7.3 *Analisador morfológico*

Meta-módulo que por si só não efectua qualquer tipo de processamento.

Trata-se de um módulo utilizado para simplificar a instanciação e chamada dos sub-módulos descritos em seguida.

No momento da instanciação este módulo recebe um objecto de opções *maco*, que contém informação acerca de quais sub-módulos terão de ser criados e dos ficheiros ficheiros a utilizar para proceder a tal criação.

Contudo, qualquer aplicação pode ignorar este módulo e prosseguir directamente à chamada dos sub-módulos.

A API deste módulo pode ser consultada no Anexo III.

Para instanciar um objecto do analisador morfológico, a aplicação de chamada necessita instanciar um objecto da classe *maco_options*, inicializar os seus campos com os valores desejados, e utilizá-lo no momento da chamada do constructor da classe *maco*.

O objecto criado criará os sub-módulos requeridos, e quando pedido para analisar frases, passará o controlo para cada sub-módulo e retornará o resultado final.

2.4.7.4 *Detector de números*

Este módulo reconhece expressões numéricas, tais como '1.220.54' ou 'dois mil e sessenta e cinco', e associa-os a um valor normalizado como lema.

Basicamente o módulo consiste num autómato de estados finito que reconhece expressões numéricas válidas.

Para línguas que não possuam uma implementação de um autómato específico, é utilizado um módulo genérico que reconhece expressões numéricas com dígitos numéricos.

A API deste módulo pode ser consultada no Anexo III.

Os parâmetros necessários à criação de um objecto desta classe são:

- código da língua - utilizado para decidir se é usado o reconhecedor genérico ou um módulo específico para a língua;
- símbolo do ponto decimal;
- símbolo do ponto dos milhares.

Os últimos dois parâmetros são necessários, porque em algumas línguas latinas a vírgula é utilizada como separador de ponto decimal, e o ponto como marca dos milhares, enquanto que em línguas como o Inglês se procede de outro modo.

Estes parâmetros tornam possível especificar qual o caracter a ser utilizado em cada uma das posições.

2.4.7.5 *Detector de marcas de pontuação*

Este módulo associa *tag*'s do discurso oral a símbolos de pontuação.

A API para este módulo pode ser consultada no Anexo III.

O constructor recebe como parâmetro o nome de um ficheiro que contém a lista de *tag's PoS* a associar a cada símbolo de pontuação.

Este módulo deverá ser utilizado após a utilização do tokenizador uma vez que só anotará símbolos que tenham sido separados após aquele passo. Por exemplo, se incluirmos as reticências (...) como um símbolo de pontuação. Isso não causará nenhum efeito a não ser que o tokenizador possua uma regra que obrigue esta *substring* a ser tokenizada como uma peça.

O formato do ficheiro de *tag*'s de pontuação, ilustrado no exemplo 2.33 - Anexo III - consiste num símbolo de pontuação por linha, no qual cada linha tem o seguinte formato:

símbolo de pontuação tag

Uma linha especial pode ser incluída para símbolos de pontuação indefinidos (qualquer palavra sem caracteres alfanuméricos é considerada um símbolo de pontuação).

Esta linha especial tem o formato <*Other*> *tag*, como se exemplifica em 2.34 que pode ser consultado no Anexo III.

2.4.7.6 *Detector de datas*

Este módulo, à semelhança do módulo de detecção de números, é uma colecção de autómatos de estado finitos específicos da língua em causa, não necessitando, deste modo, que seja fornecido algum ficheiro de dados em tempo de instanciação.

Para línguas que não possuam um autómato específico, é utilizado, por defeito, um analisador que detecta padrões simples de data (isto é, DD-MM-AAAA, MM/DD/AAAA, etc.).

A API para este módulo figura no Anexo III.

O único parâmetro esperado pelo constructor é a língua em que o texto a analisar se encontra escrito, de modo a poder escolher o autómato específico apropriado, ou seleccionar o autómato por defeito, se não houver nenhum disponível.

2.4.7.7 *Pesquisa de dicionário*

Este módulo tem duas funções:

- pesquisar as formas de uma palavra no dicionário de modo a encontrar os seus lemas e *tag's PoS*;
- aplicar as regras de afixos para encontrar a mesma informação em casos em que a forma é uma derivação não incluída no dicionário.

A decisão acerca do que é incluído no dicionário e do que é excluído através das regras de afixos é da responsabilidade do desenvolvedor de dados linguísticos.

A API para este módulo encontra-se no Anexo III.

Os parâmetros a passar ao constructor são os seguintes:

- a língua correspondente ao texto processado. Esta é requerida pelo sub-módulo de afixação, de modo a reconhecer acentos gráficos em línguas latinas;
- o nome do ficheiro com o dicionário que pode ser um ficheiro indexado em BerkeleyDB (com extensão .db) ou um ficheiro em formato de texto plano (com extensão .src);
- um valor booleano (*true* ou *false*) que indica se a análise de afixos deve ou não ser aplicada;
- o nome do ficheiro das regras de afixos (pode ser uma *string* vazia se o valor booleano anterior tiver o valor *false*).

O módulo de dicionário baseia-se na extensão do ficheiro fornecido para decidir qual o formato a utilizar:

- ficheiro de texto plano (sem quaisquer formatações) - cada linha deste ficheiro possui o seguinte formato:

  form lemma1 PoS1 lemma2 PoS2 ...., como se pode ver no exemplo 2.35, - Anexo III;

  As linhas correspondentes a palavras que são contracções podem ter um formato alternativo se a contracção estiver a ser dividida.

  Neste caso o formato adoptado é o seguinte:

  form form1+form2+... PoS1+PoS2+...., como se pode observar no exemplo 2.36, Anexo III.

  O espaço em branco actua como separador, não devendo, deste modo, existir espaços em branco entre os campos ou no final de cada linha;

- ficheiro indexado em BerkeleyDB – pode ser criado com o programa *indexdict* que é fornecido com a biblioteca FreeLing e é invocado do seguinte modo:

  indexdict indexed-dict-name.db <source-dict.src

  no qual *source-dict.src* é um dicionário em texto plano e *indexed-dict-name.db* é o ficheiro indexado resultante da execução do comando anterior, e que pode ser passado directamente ao constructor do módulo de pesquisa de dicionário.

  No exemplo 2.37 - Anexo III – é possível observar um ficheiro deste tipo.

Ainda relativamente a este módulo há a considerar o ficheiro de regras de afixação, necessário ao manipulador do dicionário.

O ficheiro é composto por duas secções opcionais:

- <Suffixes> - contém regras de sufixação;
- <Prefixes> - contém regras de prefixação.

Tanto as regras de prefixação como as de sufixação podem aparecer em qualquer ordem e têm o mesmo formato, diferindo apenas no modo como são aplicadas (as regras de prefixação são aplicadas ao início da palavra, enquanto que as regras de sufixação são aplicadas ao final da palavra).

Cada regra tem de ser escrita numa linha diferente, e tem nove campos:

- afixo para eliminar da palavra, como se pode ver no exemplo 2.38, Anexo III;
- condição da *tag* condicional da entrada de dicionário encontrada, como se pode observar no exemplo 2.39, Anexo III.

A condição é uma expressão regular escrita em Perl;

- *tag* condicional para palavra sufixada(* = manter tag na entrada de dicionário);
- verificação de adição de acentos nos lemas;
- sufixo enclítico (comportamento especial de acentuação em Espanhol);
- impedimento a módulos posteriores de afectar *tag's* adicionais à palavra;
- afectação de lema:

  qualquer combinação de F,R,L,A, ou um literal de string separados pelo sinal '+', como se mostra no exemplo 2.40, Anexo III, em que:

  - F - forma original da palavra (antes da remoção do afixo, como por exemplo, *crucecitas*);
  - R - raíz encontrada no dicionário (após remoção de afixos e reconstrução da raíz como por exemplo, *cruces*);
  - L - lema na entrada de dicionário que emparelha (por exemplo, *cruz*);
  - A - sufixo removido pela regra.

No exemplo 2.41 - Anexo III – é possível observar o modo de afectação de lemas.

- tentativa de afixação, não apenas para palavras desconhecidas;
- informação de retokenização.

No exemplo 2.42 - Anexo III - é possível observar a definição de regra de prefixo.

No exemplo 2.43 - Anexo III - encontra-se uma definição de regra de sufixo.

2.4.7.8 *Identificador multipalavra*

Este módulo agrega *tokens* dados como *input* num objecto palavra se forem encontrados numa lista de multipalavras fornecida.

A API para este módulo pode ser consultada no Anexo III.

A classe *automat* implementa um autómato de estados finito genérico.

A classe *locutions* é uma classe derivada que implementa um autómato de estados finito para reconhecer os padrões de palavras que se encontram no ficheiro fornecido ao constructor.

O ficheiro fornecido contém uma lista de multipalavras a reconhecer.

O formato do ficheiro consiste numa multipalavra por linha com três campos:

- forma multipalavra;
- lema multipalavra;
- *tag PoS* multipalavra.

No exemplo 2.44 - Anexo III - é ilustrado um ficheiro deste tipo.

2.4.7.9 *Identificador de entidades nomeadas*

Existem dois módulos que executam o reconhecimento de entidades nomeadas, sendo que, a escolha de um deles depende da aplicação.

O primeiro módulo consiste na classe *np*, que implementa um autómato de estados finito, que reconhece sequências de palavras iniciadas com maiúscula , levando em conta algumas lexias complexas[46] (por exemplo, Banco de Portugal) e frases que se iniciam com letra maiúscula.

[46] Sequência de vocábulos que tem o valor de uma única palavra.

O segundo módulo, nomeado *bioner*, consiste em algoritmos de machine learnig baseados nas bibliotecas Omlet&Fries, e tem de ser treinado a partir de um *corpus* previamente etiquetado.

O módulo *np* é simples, rápido e fácil de adaptar para utilização em novas línguas, uma vez que a utilização de maiúscula é o mecanismo básico utilizado no reconhecimento de entidades nomeadas.

A performance estimada deste módulo é de cerca de 85% de entidades nomeadas reconhecidas correctamente.

A API para este módulo pode ser consultada no Anexo III.

O ficheiro que controla o funcionamento deste módulo deverá ser constituído pelas seguintes secções:

- <FunctionWords> - lista os sintemas que podem ser encaixados num nome próprio tal como 'Banco de Portugal'.

  No exemplo 2.45 - Anexo III- pode observar-se uma declaração desta secção;

- <SpecialPunct> - lista as *tag's PoS*, após as quais pode estar uma palavra iniciada com maiúscula e que poderá indicar o início de uma frase ou cláusula e não necessariamente uma entidade nomeada.

  Exemplos típicos destes casos são:

  - o símbolo de pontuação ':';
  - parêntesis aberto '(';
  - hífen '-'.

  O exemplo 2.46 - Anexo III - mostra uma declaração desta secção.

- <NE_Tag> - secção constituída apenas por uma linha que contém a *tag PoS* que será associada às entidades reconhecidas. Se o classificador de entidades nomeadas for

utilizado mais tarde, terá de ser informado acerca desta tag no momento da sua criação.

O exemplo 2.47 - Anexo III - ilustra a utilização desta secção.

- <Ignore> - este bloco contém uma lista de formas escritas em minúsculas ou *tag's PoS* escritas em maiúsculas que não é suposto serem consideradas como entidades nomeadas, mesmo quando aparecem escritas com letra inicial maiúscula no meio de uma frase.

  Por exemplo, a palavra Português, na frase 'Ele começou a estudar Português há dois anos', não é uma entidade nomeada. Se as palavras aparecerem na lista com outras que se iniciam com maiúscula, aquelas são consideradas formarem uma entidade nomeada, como é o caso de 'Um anúncio do Banco Espanhol de Comércio foi emitido ontém'.

  Cada palavra ou *tag* é seguida por '0' ou '1' indicando se a condição ignore é estrita ou não (0: *non-strict*, 1: *strict*).

As entidades marcadas como *non-strict* terão o comportamento anteriormente descrito.

As entidades marcadas como *strict* nunca serão consideradas como entidades nomeadas ou partes de entidades nomeadas.

No exemplo seguinte a secção <Ignore> declara que a palavra 'I' não é um nome próprio, a não ser que alguma das suas palavras vizinhas o seja.

O mesmo exemplo também declara que qualquer palavra que possua a *tag* 'RB', e qualquer dos nomes listados não deverão nunca ser considerados como entidades nomeadas.

No exemplo 2.48 - Anexo III - é ilustrada a definição desta secção.

- <Names> - esta secção contém uma lista de lemas que podem ser nomes, mesmo que entrem em conflito com algum critério utilizado pelo reconhecedor de entidades mencionadas. Isto é particularmente útil quando os nomes aparecem com letra maiúscula no início da frase.

A inclusão de uma forma na secção <Names> obriga a que a escolha levada a cabo pelo reconhecedor de entidades nomeadas seja incluída na lista de possíveis *tag's* da forma em causa, dando ao *tagger* a possibilidade de decidir se se trata de um nome próprio ou de um verbo.

No exemplo 2.49 – Anexo III – encontra-se um exemplo de declaração desta secção.

- <RE_NounAdj>, <RE_Closed> e <RE_DateNumPunct> - estas três secções permitem modificar as expressões regulares definidas para as *tag's PoS* PAROLE.

  Estas expressões regulares são utilizadas pelo reconhecedor de entidades mencionadas para determinar se uma palavra de início de frase tem alguma *tag* que seja Noun ou Adj, ou qualquer outra *tag* que corresponda a uma data, sinal de pontuação ou número.

Uma declaração destas secções é mostrada no exemplo 2.50 - Anexo III.

- <TitleLimit> - esta secção contém apenas uma linha com um valor inteiro que indica o comprimento acima do qual uma frase escrita inteiramente em maiúsculas será considerada um título e não um nome próprio.

A declaração desta secção é ilustrada no exemplo 2.51-Anexo III.

Se TitleLimit=0 (valor por defeito) a detecção de título é desactivada, isto é, todas as frases escritas apenas em maiúsculas são sempre consideradas como entidades nomeadas.

O objectivo deste método é considerar que os títulos de jornal são habitualmente escritos em maiúsculas e tendem a ter pelo menos duas ou três palavras, enquanto que entidades nomeadas escritas desta forma tendem a ser siglas (por exemplo API, PDA, ...) e usualmente têm no máximo uma ou duas sequências de caracteres.

Por exemplo, a sigla 'IBM INC..', tendo menos de três palavras será considerada um nome próprio.

Convém, no entanto realçar que este método não é 100% exacto, mas em alguns casos (por exemplo, em análise de jornais) pode ser preferível ao comportamento definido por defeito, que não é 100% exacto.

- <SplitMultiwords> - esta secção contém apenas uma linha com um único valor (yes ou no).

  Se a secção *SplitMultiwords* estiver activada, as entidades nomeadas ainda são reconhecidas, mas não são tratadas como uma unidade, apenas com uma *tag PoS* para a combinação toda.

  A cada palavra será atribuída a sua própria *tag PoS*.

  As *tag's PoS* de palavras escritas em minúsculas dentro de uma entidade nomeada (tais como preposições e artigos) são mantidas inalteradas.

A utilização desta secção é ilustrada no exemplo 2.52, Anexo III.

O módulo *bioner*, apresenta uma precisão superior à do módulo anterior (mais de 90%), embora seja mais lento e a sua adaptação a novas línguas exija um *corpus* de treino e alguns recursos de engenharia.

A API para este módulo pode ser consultada no Anexo III.

O ficheiro de configuração deste módulo é constituído pelas seguintes secções:

- <RGF> - contém uma linha com o caminho do ficheiro RGF do modelo. Este ficheiro consiste na definição das características processadas pela biblioteca *libfries* que o reconhecedor de entidades mencionadas terá de considerar.

O exemplo 2.53 - Anexo III - mostra uma definição desta secção;

- <AdaBoostModel> - inclui uma linha com o caminho do ficheiro que contém o modelo adquirido com *AdaBoost*.

Estes modelos são construídos e utilizados pela biblioteca *libomlet*.

A definição desta secção é ilustrada no exemplo 2.54 - Anexo III;

- <Lexicon> - contém uma linha com o caminho do ficheiro com o léxico do modelo aprendido.

O léxico é utilizado para traduzir características codificadas em *string* para valores inteiros codificados que são necessários à biblioteca *libomlet*.

O ficheiro com o léxico é gerado pela biblioteca *libfries* em tempo de treino.

A declaração desta secção é mostrada no exemplo 2.55, Anexo III;

- <Classes> - contém apenas uma linha com as classes do modelo e respectiva tradução para as *tag's* B,I,O.

A declaração desta secção é ilustrada no exemplo 2.56 - Anexo III;

- <InitialProb> - contém as probabilidades de encontrar cada classe no início de uma frase.

Estas probabilidades são necessárias para utilização do algoritmo de *Viterbi* utilizado para anotar entidades nomeadas numa frase.

O exemplo 2.57 ilustra a declaração desta secção e pode ser consultado no Anexo III;

- <TransitionProb> - contém as probabilidades de transição de uma dada classe para outra classe, utilizadas pelo algoritmo de *Viterbi*.

A declaração desta secção é ilustrada no exemplo 2.58 - Anexo III;

- <TitleLimit> - secção descrita anteriormente em 2.4.7.9;
- <SplitMultiwords> - contém apenas uma linha com um de dois valores (*yes* ou *no*).

  Se SplitMultiwords tiver o valor *yes*, as entidades nomeadas ainda serão reconhecidas, mas não serão tratadas como uma unidade possuindo uma única *tag PoS.*

  Cada palavra terá a sua *tag PoS*.

  As palavras inteiramente escritas em maiúsculas ficam com a sua tag especificada na secção NE_Tag.

  As *tag's PoS* de palavras escritas em minúsculas dentro de uma entidade nomeada, tais como preposições e artigos, serão mantidas inalteradas.

2.4.7.10 *Identificador de quantidades*

A classe *quantities* implementa um autómato de estados finito que reconhece *ratios*, percentagens e grandezas físicas ou monetárias, como por exemplo, vinte por cento, 20%, um quinto, 1/5.

Este módulo depende do de detecção de números, anteriormente descrito.

Efectivamente, se os números não forem previamente detectados e anotados na frase, as quantidades não serão reconhecidas.

À semelhança do módulo de detecção de números, depende da língua em causa, isto é, terá de construir um autómato de estados finito que emparelhe os padrões de expressões de *ratio* na língua pretendida.

As grandezas monetárias e físicas podem ser reconhecidas em qualquer língua, desde que seja fornecido o ficheiro de dados apropriado.

A API responsável pela execução desta tarefa pode ser consultada no Anexo III.

O ficheiro de configuração necessário ao funcionamento deste módulo é constituído pelas seguintes secções:

- <Currency> - contém uma única linha constituída por um dos códigos especificados na secção <Measure>, que significa 'valor monetário'.

  O código anteriormente referido é utilizado para associar a quantidades monetárias uma *tag* distinta daquela que é utilizada para as grandezas físicas.

A declaração desta secção é exemplificada no exemplo 2.60 - Anexo III;

- <Measure> - indica o tipo de medida correspondente a cada unidade possível.

  Cada linha contém dois campos:

  - o código da medida;
  - o código da unidade.

Os códigos podem ser escolhidos pelo utilizador, e serão utilizados para a construção do lema da multipalavra da quantidade reconhecida.

O exemplo 2.61 ilustra a declaração desta secção e pode ser consultado no Anexo III.

- <MeasureNames> - esta secção descreve quais as multipalavras que têm de ser interpretadas como uma medida, e as unidades elas representam.

  A unidade deve aparecer na secção <Measure> com o respectivo código associado.

  Cada linha tem o formato *multiword_description code tag* onde:

  - *multiword_description* corresponde a um padrão multipalavra;
  - *code* é o tipo de grandeza que a unidade descreve;
  - *tag é* uma restrição dos componentes lematizados da multipalavra.

O exemplo 2.62 ilustra a declaração desta secção e pode consultar-se no Anexo III.

As multipalavras de quantidade serão reconhecidas apenas quando seguidas por um número.

É importante notar que as expressões multipalavra lematizadas, isto é, aquelas que se encontram colocadas entre parentesis rectos, só serão reconhecidas se o lema estiver presente no dicionário com as formas flexionadas correspondentes.

2.4.7.11 *Afectador de probabilidades e reconhecedor de palavras desconhecidas*

Esta classe apresenta-se como o final da análise morfológica e tem duas funções:

- associar uma probabilidade inicial a cada análise de cada palavra - estas probabilidades serão necessárias mais tarde à *tag PoS*;
- encontrar as possíveis tag's de uma palavra com base na palavra final, se essa palavra não possuir ainda nenhuma análise realizada com sucesso.

A API para este módulo pode ser consultada no Anexo III.

Como é possível observar pela declaração da classe responsável por este módulo, o

constructor recebe três parâmetros:

- o código da língua – utilizado apenas para decidir se um etiquetador EAGLES é ou não utilizado, e limitar as *tag's* se for necessário;
- o nome do ficheiro de probabilidades – este ficheiro contém toda a informação estatística necessária, pode ser gerado a partir de um *corpus* de treino etiquetado contido em *src/utilities* e é constituído pelas seguintes secções:

  – <FormTagFreq> - contém dados propabilísticos de algumas formas que possuem frequência elevada.

    Se a palavra for encontrada nesta lista, as probabilidades lexicais são calculadas utilizando os dados da secção *<FormTagFreq>*.

A lista anteriormente referida, consiste numa forma por linha com o seguinte formato:

form ambiguity-class, tag1 #observ1 tag2 #observ2 ...

As probabilidades das formas são aperfeiçoadas de modo evitar probabilidades *zero* - exemplo 2.63, Anexo III.

– <ClassTagFreq> - são declarados dados probabilísticos de classes ambíguas.

Se a palavra não for encontrada na secção <FormTagFreq>, são utilizadas as frequências da sua classe ambígua.

A lista consiste numa classe por linha, tendo cada linha o seguinte formato:

class tag1 #observ1 tag2 #observ2 ...

As probabilidades das formas são aperfeiçoadas de forma a evitar probabilidades *zero*, exemplo 2.64, Anexo III.

– <SingleTagFreq> - esta secção declara probabilidades de unigramas[47].

Se a classe de ambiguidade não for encontrada na secção *<ClassTagFreq>*, são utilizadas as frequências individuais para as suas possíveis *tag's*.

[47] Um elemento de texto, normalmente uma palavra

Nesta secção é definida uma tag por linha com o seguinte formato:

tag #observ.

As probabilidades das formas são aperfeiçoadas de forma a evitar probabilidades *zero* – exemplo 2.65, Anexo III.

- <Theeta> - define o valor para o parâmetro *theeta*, utilizado na 'suavização' das probabilidades de *tag* baseadas em sufixos de palavras.

  Se a palavra não for encontrada no dicionário e se a sua lista das possíveis *tag's* for desconhecida, a distribuição é calculada utilizando os dados das secções *<Theeta>*, *<Suffixes>* e *<UnknownTags>*.

Esta secção contém apenas uma linha com um número real, como se pode ver no exemplo 2.66, Anexo III.

- <Suffixes> - contém os sufixos obtidos a partir de um corpus de treino, com informação acerca das tag's que foram associadas à palavra com o sufixo em causa.

A lista em causa contém um sufixo por linha, tendo cada linha o seguinte formato:

suffix #observ tag1 #observ1 tag2 #observ2 ...

A declaração desta secção é ilustrada no exemplo 2.67, Anexo III.

- <UnknownTags> - lista as *tag*'s de categoria aberta a considerar como possíveis candidatas para qualquer palavra desconhecida.

  Cada linha dentro desta secção possui o seguinte formato:

  tag #observ onde:

  - *tag* é o rótulo PAROLE completo;
  - *observ* corresponde ao número de ocorrências existentes num *corpus* de treino.
  - o limite mínimo – este limite é utilizado para palavras desconhecidas, quando

a probabilidade de cada *tag* possível tiver sido calculada pelo estimador de acordo com as terminações das palavras.

As *tag's* que possuam um valor inferior ao limite especificado serão ignoradas.

2.4.7.12 *Corrector ortográfico*

Este módulo considera as semelhanças fonéticas de palavras, e tenta fornecer análises para palavras que de outra forma seriam tratadas como desconhecidas.

Por exemplo, se uma palavra incorrecta tal como '*cavallo*' for encontrada, este módulo reconhece que tem o mesmo som da palavra correcta existente no dicionário ('cavalo'), fornecendo assim uma análise para aquilo que, de outra forma, seria encarado como uma palavra desconhecida.

2.4.7.13 *Etiquetador de sentidos*

Este módulo pesquisa o lema de cada análise num dicionário de sentidos e enriquece-o com a lista de sentidos lá encontrados.

A API para este módulo pode ser encontrada no Anexo III.

Como se pode observar, o constructor desta classe recebe dois parâmetros:

- o nome do ficheiro com o dicionário de sentidos;
- um valor booleano que indica se a análise com mais de um sentido deve ou não ser duplicada.

Por exemplo, a palavra crane possui as seguintes análises:

crane

crane NN 0.833

crane VB 0.083

crane VBP 0.083

Se a lista de sentidos for simplesmente adicionada, isto é, a variável booleana *duplicate*, contém o valor *false*, obteremos o seguinte resultado:

crane

crane NN 0.833 02516101:01524724

crane VB 0.083 00019686

crane VBP 0.083 00019686.

Mas se à variável booleana for atribuído o valor *true*, obteremos valores duplicados como se mostra em seguida:

crane

crane NN 0.416 02516101

crane NN 0.416 01524724

crane VB 0.083 00019686

crane VBP 0.083 00019686.

2.4.7.14 *Desambiguador de sentidos de palavras*

Este módulo é responsável pela clarificação do sentido das palavras em frases dadas. Trata-se de um *wrapper*[48] para o algoritmo UKB que se baseia numa rede de relações

[48] Padrão que permite que determinadas classes trabalhem juntas, o que de outro modo não seria possível

semânticas para desambiguação dos sentidos mais prováveis para as palavras de um texto utilizando o algoritmo PageRank.

A API para este módulo pode ser consultada no Anexo III.

Como é possível observar, o constructor desta classe recebe quarto parâmetros:

- o ficheiro com o grafo de relações semânticas a carregar;
- o dicionário de sentidos, com todos os sentidos possíveis para cada palavra;
- os últimos dois parâmetros, são parâmetros necessários à execução do algoritmo UKB (sendo o terceiro parâmetro um valor que indica a precisão com que o algoritmo PageRank é executado e o quarto parâmetro, o número máximo de iterações do algoritmo anteriormente referido).

2.4.7.15 *Etiquetador de discurso oral*

Este módulo está dividido em dois sub-módulos:

2.4.7.15.1 hmm_tagger

Etiquetador Markoviano de trigramas[49].

A API para este módulo pode ser consultada no Anexo III.

Como se pode observar através da API, o constructor deste módulo recebe quatro parâmetros que se descrevem em seguida:

- o código da língua pretendida, utilizado para determinar se a língua utiliza um etiquetador EAGLES, encurtando neste caso as *tag's PoS*;

- o ficheiro que contém os parâmetros a utilizar pelo modelo oculto de Markov implementado pela classe constituída pelas seguintes secções:

  – <Tag> - contém a lista das probabilidades de tag's de unigramas.

[49] Sequências de três elementos de texto, normalmente palavras

Cada linha desta secção possui o seguinte formato:

Tag Probability

Tanto as *tag's* que têm probabilidade *zero*, como as que não são observadas devem ser incluídas nesta secção.

A declaração desta secção é ilustrada no exemplo 2.68, Anexo III.

- <Bigram> - contém as probabilidades de transição de bigramas.

  Cada linha desta secção tem o seguinte formato:

  Tag1.Tag2 Probability

A declaração desta secção é ilustrada nos exemplos 2.69 e 2.70, descritos no Anexo III.

- <Trigram> - contém a lista de probabilidades de transição de trigramas.

  Cada linha desta secção tem o seguinte formato:

  Tag1.Tag2.Tag3 Probability

A declaração desta secção é ilustrada nos exemplos 2.71 e 2.72, Anexo III.

- <Initial> - contém a lista de probabilidades de estados iniciais.

  Cada linha deverá ter o seguinte formato:

  InitialState LogProbability

  onde *InitialState* é um código de bigramas do tipo *PoS* com o formato 0.*tag*, exemplos 2.73 e 2.74, Anexo III.

- <Word> - declara uma lista de probabilidades de palavras e é utilizada em conjunto com as probabilidades de tag, anteriomente explicadas, para o cálculo das probabilidades de emissão necessárias ao modelo oculto de Markov implementado pela classe *hmm_tagger*.

  Cada linha desta secção possui o seguinte formato:

word LogProbability.

Esta secção deve ainda conter uma linha especial:

<UNOBSERVED_WORD>.

A declaração desta secção é ilustrada no exemplo 2.75, descrito no Anexo III.

- <Smoothing> - contém três linhas correspondentes aos coeficientes utilizados na interpolação linear das probabilidades de unigramas (c1), bigramas (c2) e trigramas (c3).

No exemplo 2.76, no Anexo III, é possível observar a declaração desta secção.

- <Forbidden> - esta secção tem como objectivo impedir a suavização de algumas combinações com probabilidade zero.

  As linhas desta secção são trigramas colocados no seguinte formato:

  Tag1.Tag2.Tag3

A declaração desta secção é ilustrada no exemplo 2.77, Anexo III.

- um valor booleano que indica se as palavras que contêm informação de retokenização devem ou não ser retokenizadas;

- um valor inteiro que indica se e quando o etiquetador deve utilizar apenas uma análise em caso de ambiguidade.

  Os possíveis valores que este parâmetro pode tomar são:

  - FORCE_NONE (ou 0) – nenhuma selecção é forçada, palavras que são ambíguas, após o processo de etiquetação de palavras, permanecem ambíguas;
  - FORCE_TAGGER (ou 1) – força o processo de selecção imediatamente a seguir

ao processo de etiquetação, e antes do processo de retokenização;

- FORCE_RETOK (ou 2) – força o processo de selecção após o processo de retokenização.

2.4.7.15.2 relax_tagger

Sistema híbrido que integra tanto conhecimento estatístico, como conhecimento codificado manualmente.

A API para este módulo pode ser consultada no Anexo III.

Como se pode observar através da API, o constructor deste módulo recebe seis parâmetros que se descrevem em seguida:

- um ficheiro de restrições composto por duas secções:
  - SETS – cada linha desta secção possui o seguinte formato:

    Set-name = element1 element2 ... elementN onde*:*

    - *Set-name* é qualquer *string* que se inicia com letra maiúscula;
    - *elements* podem ser:
      - formas colocadas entre parêntesis, tais como (computador);
      - lemas colocados entre colchetes, tais como <comer>;
      - *tag's PoS* que se iniciam com letra maiúscula, tais como NCMS000;
      - *sentidos colocados entre parêntesis rectos, tais como [00794578].*

No exemplo 2.78, descrito no Anexo III, é mostrada uma declaração desta secção.

- CONSTRAINTS - consiste numa série de definições de restrições de contexto, cada uma com o seguinte formato:

  weight core context no qual:

- *weight* é um valor real que estabelece o grau de compatibilidade da etiqueta com o contexto;
- *core* especifica a análise ou as análises (interpretação da forma) a que uma palavra ficará sujeita pela restrição.

  Este parâmetro poderá ter um dos seguintes formatos:

  - *tag* plana tal como VMIP3S0;
  - *tag curinga* tal como VMI* ou VMIP*;
  - um lema colocado entre colchetes, tal como <comer>, opcionalmente precedido por uma *tag*, como em VMIP3S0<comer>, ou por uma *tag* com *coringas* como em VMI*<comer>;
  - uma forma colocada entre parêntesis, precedida por uma *tag PoS* (ou uma *tag* com *coringas*), tal como VMIP3S0(comió) ou VMI*(comió);
  - um sentido colocado entre parêntesis rectos, opcionalmente precedido por uma *tag*, ou uma *tag* com coringas, tais como [00862617], NCMS000[00862617] ou NC*[00862617].

- *context* corresponde a uma lista de condições a que o contexto da palavra deve obdecer para que a restrição possa ser aplicada.

  Cada condição é colocada entre parêntesis e a lista é finalizada com ';'.

  Cada condição é especificada num dos seguintes formatos:

  (position terms) ou *(position terms barrier terms)*

  onde:

  - *position* é a posição relativa onde a condição deve ser satisfeita:

    - -1 indica a palavra anterior;
    - +1 indica a próxima palavra;
    - uma posição com uma estrela, como por exemplo, -2*, indica que qualquer palavra é permitida no emparelhamento desde a posição indicada até ao início/final da frase.

– *terms* é uma lista de um ou mais termos separados pelo *token or*.

Cada termo pode ser:

- uma *tag* plana *PoS* completa, tal como VMIP3S0;
- um prefixo *tag PoS* com *coringas*, tal como VMI* ou VMIP*;
- um lema colocado entre colchetes, tal como <rir>, um lema precedido por uma *tag*, tal como VMIP3S0<rir>, ou uma *tag* com *coringas*, tal como VMI*<rir>;
- uma forma colocada entre parêntesis, tal como (riu), ou uma forma precedida por uma *tag*, tal como VMIP3S0(riu), ou uma *tag* seguida por *coringas*, tal como MI*(riu);
- um código de sentido colocado entre parêntesis rectos, tal como [00862617], precedido por uma *tag*, como NCMS000[00862617], ou por uma *tag* com *coringas*, como NC*[00862617];
- uma referência de conjunto, isto é, o nome de um conjunto previamente definido, colocado entre chavetas, tal como {DetMasc};

– *barrier* obriga a que um emparelhamento do primeiro termo da lista seja aceite apenas se entre a palavra em causa e a palavra emparelhada não houver nenhum emparelhamento para o segundo termo da lista.

O exemplo 2.79, que ilustra a declaração de restrições, pode ser consultado no Anexo III.

- um valor inteiro que especifica o número máximo de iterações à espera de convergência antes de parar o algoritmo de desambiguação de palavras;
- um valor real que indica o factor de escala para os pesos das restrições;

- um valor real que representa o limite abaixo do qual quaisquer mudanças são consideradas muito pequenas. Este valor é utilizado para detectar convergência;

- um valor booleano que indica se as palavras que têm informação de retokenização deverão ou não ser retokenizadas após o processo de etiquetação;

- um valor inteiro que indica se e quando o etiquetador deve utilizar apenas uma análise em caso de ambiguidade.

Os possíveis valores que este parâmetro pode tomar são:

- FORCE_NONE (ou 0) – nenhuma selecção é forçada. Palavras que são ambíguas, após o processo de etiquetação de palavras, permanecem ambíguas;
- FORCE_TAGGER (ou 1) – força o processo de selecção imediatamente a seguir ao processo de etiquetação, e antes do processo de retokenização;
- FORCE_RETOK (ou 2) – força o processo de selecção após o processo de retokenização.

2.4.7.16 *Classificador de entidades nomeadas*

Este módulo tem a responsabilidade de associar uma classe a entidades nomeadas num texto.

Sendo a implementação de um algoritmo de Machine Learning, as classes podem ser qualquer coisa reconhecida pelo modelo treinado.

Após o processo de classificação, a *tag PoS* de uma palavra é alterada para a etiqueta definida pelo modelo treinado.

A API para este módulo pode ser consultada no Anexo III.

Como é possível observar, o constructor desta classe recebe dois parâmetros:

- a *tag* que o módulo reconhecedor de entidades nomeadas associou às entidades nomeadas, podendo deste modo, saber quais as palavras a classificar;
- prefixo dos ficheiros de configuração para o modelo.

Este módulo requer três ficheiros de configuração, com o mesmo caminho e nome, e com as seguintes extensões:

- .abm – contém um modelo AdaBoost baseado em árvores de decisão superficiais;
- .lex – contém um dicionário que associa um número a cada símbolo utilizado no modelo AdaBoost;
- .rgf – contém a informação das características de contexto que devem ser extraídas para cada entidade nomeada.

2.4.7.17 *Parser gráfico*

O parser gráfico enriquece cada instância da classe *sentence* com um objecto da classe *parse_tree*, cujas folhas têm uma ligação para as palavras da frase.

A API para este módulo encontra-se descrita no Anexo III.

O constructor da classe *chart_parser* recebe um ficheiro com a gramática CFG que se descreve em seguida.

As regras da gramática têm o formato *x* → *y, A, B*, ou seja, a cabeça da regra é um símbolo não-terminal especificado do lado esquerdo da seta (→).

O corpo da regra é constituído pela sequência de símbolos terminais e não-terminais separados por vírgulas (,) e finalizado com um ponto-final (.).

Regras vazias não são permitidas, uma vez que tendem a tornar o parser muito lento.

Regras que possuam a mesma cabeça, podem ser agrupadas numa única que tem a cabeça comum a todas as regras em causa, e os respectivos corpos separados pelo caracter '|' como se pode ver no seguinte exemplo:

x ==> A, y | B, C.

O componente que constitui a cabeça da regra pode ser prefixado com o símbolo '+', como mostra o seguinte exemplo:

nounphrase ==> DT, ADJ, +N, prepphrase.

Se a cabeça da regra não for especificada, o primeiro símbolo do corpo é assumido como a cabeça daquela.

Os símbolos terminais (*tag's PoS*) devem ser especificados, exactamente da mesma forma com que são gerados pelo *tagger*.

Os símbolos não-terminais devem aparecer com letra maiúscula.

Embora os símbolos terminais correspondam a *tag's PoS* são permitidas algumas variações para flexibilidade:

- *tag* plana: um terminal pode ser uma *tag PoS*, tal como VMIP3S0;
- *coringas*: um terminal pode ser um prefixo constituído por uma *tag PoS* seguido de *, como VMI* ou VMIP*;

O ficheiro da gramática pode ainda incluir algumas directivas que ajudam o *parser* a decidir quais as arestas do grafo deverão ser seleccionadas para a construção da árvore.

Essas directivas são:

- @NOTOP – símbolos não-terminais listados nesta directiva não são considerados como raízes válidas da árvore, mesmo que cubram a frase completa;
- @START – especifica qual o símbolo inicial da gramática;
- @FLAT – as sub-árvores correspondentes a símbolos não-terminais que sejam "aplanados", são aplanadas quando o símbolo é recursivo.

Apenas a ocorrência mais elevada aparece na árvore de *parse* final;

- @HIDDEN – os símbolos não-terminais especificados sob esta directiva não aparecerão na árvore final;
- @PRIOR – esta directiva coloca os símbolos não-terminais por ordem de prioridade descendente (ao último símbolo não-terminal da lista corresponde a prioridade mais baixa).

O método *get_start_symbol* retorna o símbolo inicial da gramática e é necessário ao *parser* de dependências.

2.4.7.18 *Parser de dependências*

Este módulo recebe um conjunto de frases já analisadas, isto é, objectos da classe sentence, que foram já enriquecidos com uma parse_tree pelo parser gráfico.

A API deste módulo pode ser consultada no Anexo III.

Como é possível observar, pela definição da *classe dep_txala*, o constructor recebe duas *strings*:

- o nome do ficheiro que contém as regras de dependências a ser utilizado, constituído pelas seguintes secções:

  - <GRPAR> - contém as regras que permitem completar a análise produzida pelo parser gráfico.

    As regras da gramática são aplicadas da prioridade mais alta para a mais baixa e da esquerda para a direita e podem ser activadas ou desactivadas através de *flags* globais de activação.

    Cada linha contém uma regra com o seguinte formato:

    priority flags context (ancestor,descendant) operation op-params flag-ops onde:

    - *priority* é um valor numérico que indica a prioridade da regra;
    - *flags* corresponde a uma lista de strings separadas pelo símbolo '|'. Cada *string* corresponde ao nome de uma *flag* que activará a regra. Se *enabling flags* for igual a '-', a regra estará sempre activa;
    - *context* é uma limitação de contexto à aplicação da regra apenas aos pares de pedaços que estão limitados pelos contextos apropriados ('-' significa que não existem limitações, e que a regra é aplicada a qualquer par de pedaços que emparelhem com a frase);
    - *(ancestor,descendant)* corresponde às etiquetas de pares adjacentes de pedaços da regra que será aplicada. As etiquetas, ou são associadas pelo *parser*, ou por alguma operação de RELABEL em alguma outra regra.

O par anteriormente referido tem de ser colocado entre parêntesis, ser separado por uma vírgula e não poderá conter espaços em branco.

As etiquetas podem ser sufixadas com uma condição extra da forma *(form), <lemma>*, *[class]*, ou *{PoS_regex}*;

A seguinte tabela ilustra a utilização de labels e o seu significado.

A label:	**corresponderia:**
np	qualquer pedaço de frase etiquetado com *np*
np(cats)	qualquer pedaço de frase etiquetado com *np* através de uma regra que tem à cabeça a forma *cats*
np<cat>	qualquer pedaço de frase etiquetado com *np* através de uma regra que tem à cabeça o lema *cat*
np[animal]	qualquer pedaço de frase etiquetado com *np* através de uma regra que tem à cabeça um lema da categoria *animal*
np{^N.M}	qualquer pedaço de frase etiquetado com *np* através de uma regra que tem à cabeça uma palavra com uma *tag* PoS que emparelha a expressão regular *^N.M*

Tabela 3 - *Labels* utilizadas pelo *parser* de dependências e seu significado.

- *operation* é o modo no qual os nós ancestor e descendant são combinados;
  - o componente *op-params* tem dois significados, dependendo do campo *operation* : as operações *top_left* e *top_right* devem ser seguidas pelo literal RELABEL juntamente com as novas etiquetas a associar a cada pedaço;
  - As outras operações devem ser seguidas pelo literal MATCHING seguido da etiqueta a ser emparelhada;

– <GRLAB> - contém dois tipos de linhas:

  - UNIQUE label1 label2 label3 ... – As etiquetas declaradas nesta lista serão afectadas uma única vez por cabeça, isto é, se uma cabeça tiver uma filha com uma dependência já etiquetada com label1, as regras de

afectação dessa etiqueta serão ignoradas para todas as outras regras filhas com a mesma cabeça.

Por exemplo, se um verbo tiver obtido uma etiqueta subject para uma das suas dependências, nenhuma outra dependência obterá aquela etiqueta, mesmo que as condições o permitam.

- ancestor-label dependence-label condition1 condition2 ... – onde:
  - ancestor-label é a etiqueta do nó que é cabeça da dependência;
  - dependence-label é a etiqueta a ser associada à dependência;
  - condition é uma lista de condições a que a dependência tem de obedecer para satisfazer a regra.

Cada condição tem uma das seguintes formas:

- node.attribute = value;
- node.attribute != value.

onde:

- node é uma string que descreve um nó no qual o atributo attribute tem de ser verificado.

  node especifica um caminho para encontrar o nó a ser verificado.

O caminho deve iniciar com p (nó pai) ou d (nó descendente), e pode ser seguido por uma lista de etiquetas separadas por ':'.

Por exemplo, p:sn:n, refere-se ao primeiro nó etiquetado n encontrado abaixo de um nó etiquetado sn, que por sua vez depende do nó pai p.

node pode também tomar o valor As ou o valor Es que verificarão a lista de todos os filhos do antecessor p excluíndo o filho em foco, d.

As e Es também podem ser seguidos por um caminho, tal como p e d.

Por exemplo, Es:sn:n selecionará um irmão com aquele caminho, e As:sn:n verificará que todos os irmãos têm aquele caminho;

- value é uma string a ser emparelhada, ou um conjunto de strings separadas por '|'.

As strings podem ter coringas colocados à direita, isto é, np* é uma string permitida, mas n*p é uma string proibida.

Para o atributo pos, value pode ser qualquer expressão regular válida;

- attribute pode ser:

  - label - chunk label ou tag PoS do nó;
  - side - posição direita ou esquerda do nó especificado relativamente a outro.

Esta propriedade é válida apenas para os nós p e d;

- lemma - lema do nó da palavra da cabeça do nó;
- pos - tag PoS da palavra da cabeça do nó;
- class - classe de palavras do lema da palavra da cabeça do nó;
- tonto - propriedades ontológicas EWN da palavra da cabeça do nó;
- semfile - ficheiro semântico WN da palavra da cabeça do nó;
- synon - lemas sinónimos da palavra da cabeça do nó, de acordo com a WN;
- asynon - lemas sinónimos dos predecessores da palavra da cabeça do nó, de acordo com a WN.

Desde que não seja exigida nenhuma desambiguação, os atributos referentes a propriedades semânticas serão satisfeitos se qualquer dos sentidos satisfizer a condição.

Os exemplos 2.80, 2.81, 2.82 e 2.83 -Anexo III - ilustram a definição de regras.

- <SEMDB> - esta secção só é necessária se as regras de etiquetação de dependências da secção <GRLAB> utilizarem condições em valores semânticos, isto é, *tonto*, *semfile*, *synon*, ou *asynon*.

Sempre que necessária, esta secção terá de ser definida antes da secção GRLAB.

A secção tem de conter duas linhas que especificam os dois ficheiros com a informação semântica.

A definição desta secção terá de obedecer ao seguinte formato:

<SEMDB>

SenseFile ../senses16.db

WNFile ../../common/wn16.db

</SEMDB>

O ficheiro definido em SenseFile, terá de ser um ficheiro indexado BerkeleyDB.

O ficheiro definido em WNFile, terá, tal como o anterior, ser do tipo BerkeleyDB indexado, contendo a informação ontológica.

- <CLASS> - esta secção contém as definições de classes que podem ser utilizadas como atributos nas regras de etiquetação de dependências.

  Cada linha desta secção possui um dos seguintes formatos:

  - *class-name lemma comments;*
  - *class-name "filename" comments.*

Por exemplo, as seguintes linhas afectam à classe *mov* os quatro verbos indicados, e à classe *animal* todos os lemas encontrados no ficheiro *animals.dat*.

mov go prep= to,towards

mov come prep= from

mov walk prep= through

mov run prep= to,towards D.O.

animal "animals.dat"

Qualquer coisa que apareça à direita do segundo campo é tratado como um comentário.

- *o símbolo inicial da gramática utilizado pelo chart_parser para analisar frases.*

O *parser* de dependências funciona em três etapas:

- na primeira etapa, as regras da secção <GRPAR> do ficheiro de dependências são utilizadas para completar a análise produzida com uma árvore de *parsing* completa.

  As regras são aplicadas a um par de pedaços adjacentes.

  Em cada passo, o par seleccionado é fundido num único pedaço.

  O processo pára quando resta apenas um único pedaço;

- a segunda etapa consiste na conversão da árvore de parsing completa numa árvore de dependências.

  O processo de conversão é simples, uma vez que a gramática de *parsing* codifica a informação acerca da cabeça de cada regra;

- a terceira etapa consiste no processo de etiquetação.

  Cada aresta da árvore de dependências é etiquetada com uma função sintáctica utilizando as regras definidas na secção <GRLAB>.

2.4.7.19 *Resolução de co-referência*

Este módulo implementa um algoritmo de Machine-Learning baseado num *solver* de co-referência.

O funcionamento deste módulo requer um documento, analisado pelo *parser* para detectar sintagmas nominais, e decidir quais os sintagmas nominais que são co-referenciados.

A API deste módulo pode ser consultada no Anexo III.

Os parâmetros que o constructor recebe são um ficheiro e uma máscara de bits inteira que especifica quais os atributos que têm de ser utilizados pelo classificador.

O ficheiro de configuração que é passado ao constructor possui as seguintes secções:

- <ABModel> - declara o caminho do ficheiro que contém o modelo de treino *AdaBoost* baseado em árvores de decisão superficiais.

  A declaração desta secção é ilustrada pelo exemplo 2.84, Anexo III;

- <SemDB> - especifica os dois ficheiros (*SenseFile* e *WNFile*) que contêm uma base de dados semântica, informação que é necessária ao *solver* no cálculo de atributos WN-based, exemplo 2.85 no Anexo III;
- <MaxDistance> - declara a distância máxima em palavras à qual possíveis co-referências serão consideradas.

Valores pequenos levam o *solver* a perder co-referentes distantes. Por outro lado, valores muito elevados introduzem uma grande quantidade de possíveis pares candidatos co-referentes, tornam o sistema muito lento e produzem uma grande quantidade de falsos positivos[50].

O significado dos atributos pode ser encontrado no ficheiro source que se encontra em include/freeling/coref_fex.h.

Para utilizar este módulo, ter-se-á de colocar o valor deste parâmetro igual a 0xFFFFFF para seleccionar todos os atributos.

[50] Resultado que ocorre quando a hipótese nula é satisfeita, mas rejeitada pelo teste.

2.4.7.20 *Base de dados semântica*

Este não é um módulo autónomo, pois pode ser utilizado para enriquecer os resultados das análises efectuadas por outros.

É utilizado por outros módulos que necessitam aceder à base de dados semântica, tais como:

- o anotador de sentidos, *senses*;
- o *parser* de dependências, *dep_txala*;
- o resolverdor de coreferências, *coref*.

A API deste módulo pode ser consultada no Anexo III.

Como é possível observar, o constructor desta classe recebe duas *strings* como parâmetros que correspondem ao dicionário de sentidos e ao ficheiro WordNet.

O ficheiro que corresponde ao dicionário de sentidos contém várias linhas com o seguinte formato:

type:lemma:PoS synset1 synset2 ....

O campo *type* pode ser *W (Word)* ou *S (Sense)* e indica se o resto da linha contém uma palavra e todos os seus códigos de sentidos, ou um código de sentido e todas as suas palavras sinónimas.

Para entradas do tipo '*W*' assume-se que a lista de códigos de sentidos é ordenada por ordem decrescente de frequência para aquele lema PoS, pelo módulo de anotação de sentidos.

As entradas do tipo '*S*' são utilizadas pelas regras do *parser* de dependências.

O exemplo 2.86 - Anexo III - ilustra a definição deste tipo de ficheiro.

O ficheiro WordNet é um ficheiro indexado Berkeley DB, que pode ser criado com o programa *indexdict* fornecido com o FreeLing, invocando aquele da seguinte forma:

indexdict indexed-wn-name <source-wn

O ficheiro *source* deve conter em cada linha a informação relativa a um dado sentido com o seguinte formato:

synset:PoS hypern:hypern:...:hypern semfile TopOnto:TopOnto:...:TopOnto

onde:

- o primeiro campo é o código *synset* juntamente com a sua *tag* PoS separados por ':';
- o segundo campo é uma lista de hiperonímias[51] separadas por ':';
- o terceiro campo corresponde ao ficheiro semântico WN a que o sentido pertence;
- o último campo é uma lista de códigos EuroWN TopOntology válidos para o sentido em causa.

2.4.8 PoS Tree-Tagger

Ferramenta desenvolvida por Helmut Schmid no Instituto de Linguística Computacional da Universidade de Estugarda[52], sob licença Treetagger, procede à anotação de um determinado texto, inserido como *input*, com *tag's PoS* e devolve informação acerca dos lemas existentes, como se pode observar no exemplo 2.87 descrito no Anexo III.

2.4.9 PtStemmer

Ferramenta desenvolvida por Pedro Oliveira[53] cujo objectivo é a lematização de um texto dado em formato plano.

O exemplo 2.88 ilustra o funcionamento desta ferramenta, e pode ser consultado no Anexo III.

[51] Relação semântica de inclusão entre uma unidade lexical mais genérica (hiperónimo) e outra mais específica (hipónimo), em que esta é dependente semanticamente da primeira

[52] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

[53] http://www.cpdomina.net

2.4.10 Rembrandt

Esta ferramenta, desenvolvida no Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa[54], é um sistema de reconhecimento de entidades mencionadas e de detecção de relações entre entidades e foi concebido para efectuar o reconhecimento de todo o tipo de entidades mencionadas em textos escritos em português.

O funcionamento desta ferramenta é divido em três etapas:

- reconhecimento de expressões numéricas e geração de candidatas a entidades mencionadas - durante esta fase os textos são previamente divididos em frases e unidades, por intermédio do atomizador disponibilizado pela Linguateca, através do módulo Perl Lingua::PT::PLN, disponível em http://search.cpan.org/~ambs/Lingua-PT-PLN/.

  Um conjunto inicial de regras procede à identificação de expressões numéricas no texto, tais como unidades compostas só por algarismos, ou números escritos por extenso, ordinais e cardinais.

  Seguidamente, são aplicadas regras que procedem ao reconhecimento de expressões temporais e valores, tirando proveito dos números anteriormente reconhecidos.

  A geração de candidatas a entidades nomeadas realiza-se pela identificação de sequências de unidades que contenham pelo menos uma letra maiúscula ou um algarismo, podendo existir uma das seguintes unidades, desde que não inicie ou termine a entidade nomeada:

  - ✓ *de;*
  - ✓ *da;*
  - ✓ *do;*
  - ✓ *das;*
  - ✓ *dos;*
  - ✓ *e.*

[54] http://xldb.di.fc.ul.pt/Rembrandt/

- classificação de entidades mencionadas - nesta etapa, cada uma das entidades nomeadas é previamente classificada pela SASKIA, sendo posteriormente classificada através de regras gramaticais.

  Desta dupla classificação resultam duas vantagens:

  1. a SASKIA efectua uma classificação de acordo com os vários significados que a entidade mencionada pode ter, criando-se deste modo um ponto inicial a partir do qual o processo de desambiguação consegue melhor encontrar o correcto significado da entidade mencionada;
  2. as regras gramaticais englobam sinais externos e internos das entidades mencionadas, o que permite efectuar uma supervisão das classificações da SASKIA segundo o contexto das entidades mencionadas.

O exemplo 2.89, que pode ser consultado no Anexo III, ilustra o funcionamento destas regras.

A terminar a etapa de classificação, é aplicada uma segunda ronda de regras gramaticais, que aproveita as classificações existentes para detectar entidades mencionadas com uma morfologia mais elaborada.

Durante esta fase, ou é aplicada uma etiqueta <ALT> às entidades nomeadas, ou, se necessário, estas são novamente separadas em entidades mais pequenas, que, por sua vez, serão sujeitas ao processo de classificação da SASKIA e das regras gramaticais;

- repescagem de entidades mencionadas sem classificação - nesta etapa, é efectuada a detecção de relações entre entidades mencionadas utilizando um conjunto de regras específicas para a tarefa em causa.

  As relações detectadas são utilizadas para recuperar entidades nomeadas sem classificação, mas que se relacionam com entidades nomeadas correctamente classificadas.

  Posteriormente, é efectuada uma nova e última recuperação de entidades nomeadas com nomes de pessoas por comparação com uma lista de nomes comuns.

Finalmente, as entidades nomeadas que continuam sem classificação são eliminadas, assim como números por extenso sem uma letra maiúscula e aquelas que pertencem à categoria NUMERO são convertidas em VALOR/QUANTIDADE.

2.5 CONJUGADORES VERBAIS

2.5.1 Conjugue

Script desenvolvido pelo Instituto de Matemática e Estatística da Universidade de São Paulo[55], escrito em awk[56], disponível sob licença GPL, capaz de conjugar verbos da língua portuguesa, a partir de um banco de paradigmas.

O exemplo 2.90 ilustra o funcionamento desta ferramenta e pode ser consultado no Anexo IV.

2.5.2 Gconjugue

Interface gráfica disponível em http://gconjugue.codigolivre.org.br/, desenvolvida sob licença GPL, baseada no conjugue descrito no ponto anterior.

A Figura 36 ilustra o funcionamento desta ferramenta, na conjugação do verbo *olhar*.

Figura 36 - Utilização do Gconjugue para conjugação do verbo *olhar*.

[55] http://www.ime.usp.br/webadmin/

[56] Programa utilizado para seleccionar registos de um ficheiro e executar operações sobre eles.

2.5.3 LX-Conj

Ferramenta que executa a conjugação de um determinado verbo dado no infinitivo – disponível online em http://lxcenter.di.fc.ul.pt/pt/LXServicesConjugatorPT.html.

O exemplo 2.91 - Anexo IV - mostra a execução desta ferramenta na conjugação do verbo *ser*.

2.6 EXTRACTORES DE N-GRAMAS

2.6.1 NSP

O NSP – Ngram Statistic Package - é um "*package*" de ferramentas desenvolvido pela equipa liderada por Ted Pedersen[57], disponível sob licença GPL, que tem como objectivo proceder à extracção e classificação de n-gramas a partir de um *corpus*.

O modelo de funcionamento deste "package" é ilustrado pela Figura 2.37.

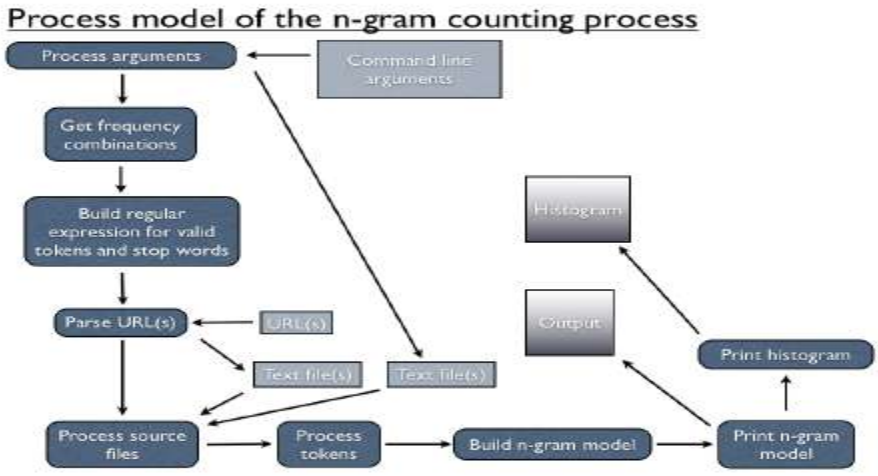


Figura 37 - Arquitectura do "*package*" NSP.

Este package é constituído por dois programas principais e alguns utilitários descritos em seguida.

[57] http://www.d.umn.edu/~tpederse/

2.6.1.1 *Programas principais*

2.6.1.1.1 count.pl

Este programa recebe como *input* um ou mais ficheiros em formato de texto plano, parte os textos introduzidos em *tokens* e gera uma lista com os n-gramas existentes naqueles mostrando-os em seguida por ordem decrescente de frequência.

Os exemplos 2.92, 2.93 e 2.94 - Anexo V - ilustram o funcionamento deste programa.

2.6.1.1.2 statistics.pl

Este programa recebe como input uma lista de n-gramas juntamente com as suas frequências, gerada pelo programa *count.pl* e executa uma medida estatística de associação, definida pelo utilizador, de modo a efectuar o cálculo de uma contagem para cada um dos n-gramas.

Os n-gramas e as respectivas contagens são mostrados por ordem decrescente.

O modo habitual de invocação deste programa é a seguinte:

statistic.pl dice test.dice test.cnt

onde:

- *dice* é o nome da biblioteca estatística a ser carregada;
- *teste.dice* é o nome do ficheiro de output no qual os resultados da aplicação da biblioteca especificada em dice serão armazenados;
- *test.cnt* é o nome o ficheiro de *input* que contém os n-gramas e os valores das suas frequências.

As medidas de associação definidas neste "*package"* que podem ser utilizadas estão divididas da seguinte forma:

– medidas de bigramas

- Dice Coefficient (Text::NSP::Measures::2D::Dice::dice);
- Fishers exact test - left sided (Text::NSP::Measures::2D::Fisher::left);
- Fishers exact test - right sided (Text::NSP::Measures::2D::Fisher::right);
- Fishers twotailed test - right sided (Text::NSP::Measures::2D::Fisher::twotailed);
- Jaccard Coefficient (Text::NSP::Measures::2D::Dice::jaccard);
- Log-likelihood ratio (Text::NSP::Measures::2D::MI::ll);
- Mutual Information (Text::NSP::Measures::2D::MI::tmi);
- Odds Ratio (Text::NSP::Measures::2D::odds);
- Pointwise Mutual Information (Text::NSP::Measures::2D::MI::pmi);
- Phi Coefficient (Text::NSP::Measures::2D::CHI::phi);
- Pearson's Chi Squared Test (Text::NSP::Measures::2D::CHI::x2);
- Poisson Stirling Measure (Text::NSP::Measures::2D::MI::ps);
- T-score (Text::NSP::Measures::2D::CHI::tscore).

– medidas de trigramas

- Log-likelihood ratio (Text::NSP::Measures::3D::MI::ll);
- Mutual Information (Text::NSP::Measures::3D::MI::tmi);
- Pointwise Mutual Information (Text::NSP::Measures::3D::MI::pmi);
- Poisson Stirling Measure (Text::NSP::Measures::3D::MI::ps).

– medidas de 4-gramas

- Log-likelihood ratio (Text::NSP::Measures::4D::MI::ll).

O exemplo 2.95 - Anexo V - ilustra o funcionamento deste programa.

2.6.1.2 *Utilitários*

2.6.1.2.1 combig.pl

Esta ferramenta combina as contagens de frequências de bigramas calculadas a partir do mesmo par de palavras em qualquer ordem possível.

O exemplo 2.96 - Anexo V - ilustra a execução deste programa a partir do ficheiro de bigramas gerado pelo programa 'count.pl' a partir do exemplo 2.92 - Anexo V.

2.6.1.2.2 count2huge.pl

Este programa recebe como *input* o ficheiro gerado pelo programa '*count.pl*' e coloca os bigramas por ordem alfabética.

O exemplo 2.97 - Anexo V - mostra o funcionamento deste programa com base no ficheiro de bigramas gerado pelo programa 'count.pl' utilizando grandes quantidades de dados repartindo-os por vários ficheiros separados, efectuando as contagens em cada um dos ficheiros e juntando-as todas no final, com o objectivo de obter os resultados globais.

No exemplo 2.98 - Anexo V - é possível observar o funcionamento deste programa.

2.6.1.2.3 huge-delete.pl

Este programa recebe uma lista de bigramas e elimina aqueles que possuem uma frequência acima ou abaixo do valor especificado, como se pode ver no exemplo 2.99 - Anexo V.

2.6.1.2.4 huge-merge.pl

Esta ferramenta tem a função de proceder à junção de ficheiros ordenados de bigramas.

O exemplo 2.100, que mostra o funcionamento desta ferramenta, pode ser consultado no Anexo V.

2.6.1.2.5 huge-sort.pl

Esta ferramenta recebe como input um ficheiro de bigramas gerado pelo programa count.pl com a opção --tokenlist, efectua a contagem das frequências dos bigramas e ordena-os alfabeticamente.

O exemplo 2.101 - Anexo V - ilustra o funcionamento desta ferramenta.

2.6.1.2.6 huge-split.pl

Esta ferramenta é responsável por partir bigramas em peças mais pequenas.

O exemplo 2.102 - Anexo V - ilustra o funcionamento desta ferramenta.

2.6.1.2.7 kocos.pl

Este programa encontra k-ésima ordem de co-ocorrências[58] de uma palavra – exemplo 2.103, Anexo V.

2.6.1.2.8 rank.pl

Programa que, dado o mesmo conjunto de n-gramas, gerado pelo programa '*count.pl*', calcula o coeficiente de correlação entre duas medidas estatísticas a que aquele conjunto tenha sido submentido – exemplo 2.104, Anexo V.

[58] Palavras que ocorrem juntas no mesmo contexto

2.6.2 SENTA

Ferramenta desenvolvida pelo Departamento de Informática da Universidade da Beira Interior que permite efectuar associações textuais n-árias de um corpus[59].

2.7 FERRAMENTAS ESPECIALIZADAS

2.7.1 DepPattern

"Package" de ferramentas linguísticas desenvolvido pelo Grupo ProLNat da Universidade de Santiago de Compostela[60], disponível sob licença GPL, que compreende um compilador de gramáticas, etiquetadores *PoS* e *parsers* baseados em dependências.

Os exemplos 2.105 e 2.106 que ilustram o funcionamento desta ferramenta podem ser consultados no Anexo VI.

2.7.2 DiZer 2.0

Interface Web desenvolvida por uma equipa constituída pelo NILC[61], pelo LIA[62] e pelo IULA[63] baseada na ferramenta DiZer, utilizada para efectuar análises de discurso.

O sistema produz a estrutura do discurso de um texto introduzido como *input*, seguindo a teoria retórica da estrutura.

O exemplo 2.107 - Anexo VI - mostra o funcionamento desta ferramenta.

[59] http://senta.di.ubi.pt/

[60] http://gramatica.usc.es/pln/tools/deppattern.html

[61] http://www.nilc.icmc.usp.br/nilc/index.html

[62] http://lia.univ-avignon.fr/

[63] http://www.iula.upf.edu/

2.7.3 EELO

Ferramenta disponível online em http://label.ist.utl.pt/pt/eelo_online_pt.php que efectua a etiquetação de um *corpus* introduzido como *input*.

O exemplo 2.108 ilustra o funcionamento desta ferramenta e pode ser consultado no Anexo VI.

2.7.4 e-Termos

Ferramenta desenvolvida em parceria entre a CNPTIA, a USP e a UFSCar representadas pelos laboratórios de pesquisa LabInfo[64], NILC[65] e GETerm[66].

Esta ferramenta encontra-se disponível em ambiente colaborativo web que implementa as seguintes funcionalidades:

- gestão colaborativa de projectos terminológicos, lexicográficos e de tradução;
- controle integrado da estrutura de projectos, de equipas e etapas de trabalho;
- equipa multidisciplinar com perfis profissionais específicos;
- ferramentas de comunicação síncronas e assíncronas;
- compilação automática e semi-automática de *corpora*:
  - contadores de frequência de palavras;
  - contadores de frequência de uma única palavra ou expressão;
  - concordanciadores;
- identificação e recuperação de lexias.

[64] http://www.cnptia.embrapa.br/

[65] http://www.nilc.icmc.usp.br/nilc/

[66] http://www.geterm.ufscar.br/

2.7.5 Etiquet(H)AREM

Ferramenta desenvolvida pelo Pólo de Coimbra da Linguateca[67] que tem como objectivo a anotação de corpora com vista à etiquetação de entidades mencionadas e de relações entre entidades mencionadas.

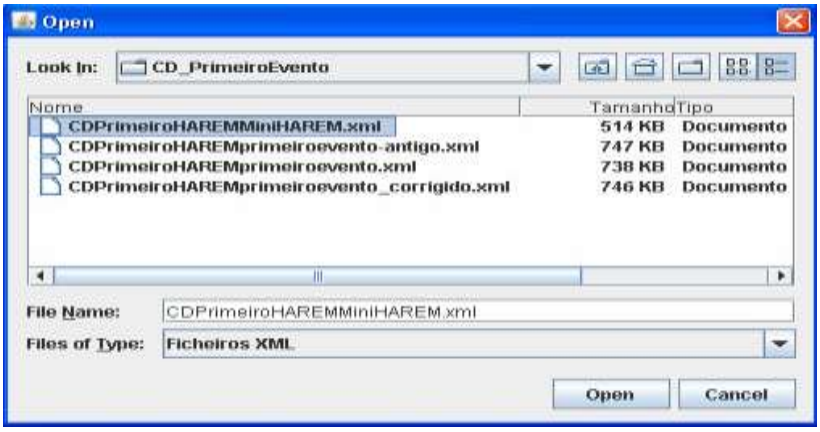


Figura 38 - Selecção do ficheiro a analisar.

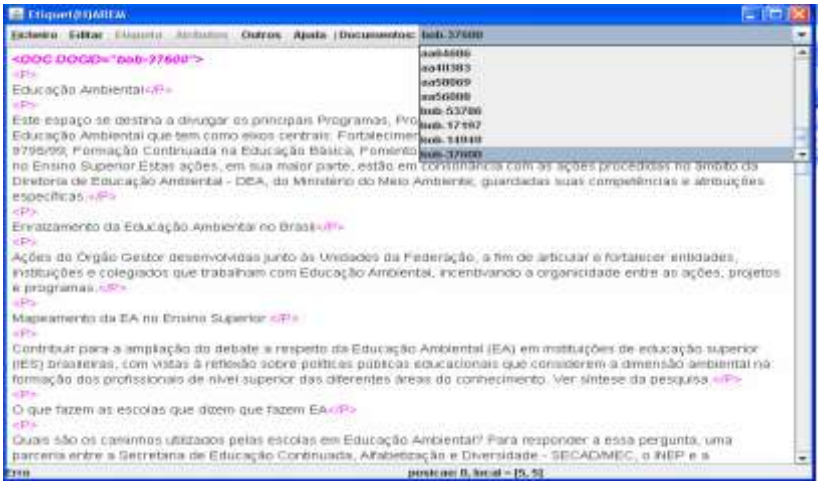


Figura 39 - Selecção do identificador do documento a analisar.

Figura 40 - Selecção da categoria da entidade pretendida.

[67] http://linguateca.dei.uc.pt/index.php?sep=recursos

2.7.6 HPC

O HPC- Historical Portuguese Corpora - é um sistema desenvolvido pelo NILC[68] e que tem o objectivo de processar documentos históricos escritos em português.

Este sistema é constituído pelas seguintes ferramentas:

2.7.6.1 *Procorph*

Sistema de gestão de dicionários históricos;

2.7.6.2 *Siaconf*

Sistema para extracção automática de variantes do discurso baseada em regras de transformação;

2.7.6.3 *Renahb*

Sistema que devolve entidades nomeadas abreviadas;

2.7.6.4 *Protej*

Sistema que converte texto plano com anotação simples em anotação XML ou em texto plano sem anotação;

2.7.6.5 *Protew*

Sistema utilizado para converter documentos Word em texto plano com anotação simples.

[68] http://www.nilc.icmc.usp.br/nilc/projects/hpc/

2.7.7 Indexador estatístico

Ferramenta desenvolvida por Rodrigo Panchiniak Fernandes[69] disponível sob licença GPL que executa análise estatística de *corpora*.

O exemplo 2.109, que ilustra o funcionamento desta ferramenta, pode ser consultado no Anexo VI

2.7.8 Lácio-Web

Package disponível online em http://www.nilc.icmc.usp.br/lacioweb/ e que tem como objectivo a divulgação de:

- vários *corpora* do português brasileiro (Lácio-Ref, Lácio-Dev, Par-C, Comp-C, Mac-Morpho e Lácio-Sint) escrito contemporâneo, representando bancos de textos adequadamente compilados, catalogados e codificados num padrão que possibilite fácil intercâmbio, navegação e análise;
- ferramentas linguístico-computacionais, tais como contadores de frequência, concordanciadores e etiquetadores morfossintáticos.

As ferramentas disponíveis neste *package* são apresentadas em seguida de acordo com o *corpus* a que estão associadas:

2.7.8.1 *corpus Lácio-Ref*

- contador de frequência padrão;
- contador de frequência por palavra;
- concordanciador para corpus sem anotação;
- etiquetadores morfossintácticos.

[69] http://search.cpan.org/~fernandes/

2.7.8.2 *corpus MAC-Morpho*

- concordanciador para *corpus* anotado morfossintaticamente.

2.7.8.3 *corpus em português do utilizador*

- o utilizador pode fazer "upload" de um *corpus* escrito em Português do Brasil para que o mesmo seja processado pelos contadores de frequência, concordanciadores para corpus sem anotação e etiquetadores.

2.7.9 Lingua Toolkit

Package de ferramentas desenvolvido por Pablo Gamallo Otero[70], disponível sob licença GPL e que tem como objectivo realizar o cálculo de medidas de similaridade entre as palavras de um *corpus* dado como *input*.

As ferramentas contidas neste *package* são:

2.7.9.1 *MultiLingua*

Analisador sintáctico;

2.7.9.2 *AutoThesaurus*

Gerador de *thesaurus*[71].

Os exemplos 2.110 e 2.111 ilustram o funcionamento das ferramentas contidas neste *package* e podem ser consultadas no Anexo VI.

[70] http://gramatica.usc.es/~gamallo/

[71] Dicionário que contém uma lista de palavras agrupadas de acordo com a semelhança de significado (contendo sinónimos e por vezes antónimos)

2.7.10 Multilingual Dependency *Parser*

Ferramenta desenvolvida por Pablo Gamallo Otero[72], disponível sob licença GPL, que tem como objectivo proceder à análise sintáctica de um corpus dado como input no formato de texto plano, isto é, sem quaisquer formatações.

O output gerado consiste em várias linhas de texto contendo cada uma:

- os lemas de cada frase etiquetados com tag's PoS;
- um terno do tipo (relation;head_lemma;dependent_lemma).

2.7.11 Multilingual Term Extractor

Ferramenta desenvolvida por Pablo Gamallo Otero[73], sob licença GPL, que recebe como *input* um ficheiro contendo um texto sem quaisquer formatações, e gera uma lista de termos em quatro fases:

- etiquetação morfossintáctica - nesta fase procede-se à etiquetação morfossintáctica do texto de *input* mediante a utilização do *TreeTagger* ou do *FreeLing*;
- geração de padrões de etiquetas - após o processo de etiquetação anteriormente realizado, procede-se à selecção de expressões que aparecem em cinco padrões de etiquetas(N=nome, A=adjectivo, P=preposição, V=verbo, PCLE=partícula):

  - N-A;
  - A-N;
  - N-N;
  - N-P-N;
  - V- PCLE;

- filtragem - o sistema selecciona as expressões que ocorrem no corpus com uma frequência superior a um determinado valor (freq = 1, por defeito).

[72] http://gramatica.usc.es/~gamallo/

[73] http://gramatica.usc.es/~gamallo/

O formulário permite escolher a frequência como um valor inteiro compreendido entre um e cinco inclusivé.

- ordenação - posteriormente à fase de filtragem procede-se à ordenação decrescente da lista obtida, utilizando para o efeito uma das seguintes medidas estatísticas:

  - co-ocorrências;
  - logarítmo;
  - qui-quadrado;
  - informação mútua;
  - scp.

2.7.12 Navegador MultiWordnet

Ferramenta disponível online em http://lxcenter.di.fc.ul.pt/pt/LXServicesWordnetPT.html. Apresenta um dicionário visual gráfico dos possíveis contextos em que uma dada palavra ou expressão poderá aparecer.

A Figura 41 ilustra o funcionamento desta ferramenta.

Figura 41 - Funcionamento do Navegador MultiWordnet.

2.7.13 NILC's Taggers

Projecto desenvolvido pelo NILC[74] que tem como objectivo o estudo e a avaliação do estado da arte de vários *taggers* existentes na Web.

2.7.14 O Constructor

Ferramenta computacional interactiva, disponível online em http://www.leonel.profusehost.net/indexc.htm, que tem como objectivo a construção de comandos destinados à pesquisa linguística nos corpora do português disponíveis no âmbito da Linguateca.

O exemplo 2.113 - Anexo VI - ilustra o funcionamento desta ferramenta.

2.7.15 SciPo

Conjunto de ferramentas integradas desenvolvido pelo NILC[75] e disponível online em http://www.nilc.icmc.usp.br/~scipo/, que tem como objectivo proporcionar auxílio a estudantes na escrita de resumos e introduções de textos científicos.

O exemplo 2.114, que pode ser consultado no Anexo VI, mostra o funcionamento desta ferramenta.

2.7.16 SciPo-Farmácia

Esta ferramenta desenvolvida pelo NILC e disponível online em http://www.nilc.icmc.usp.br/scipo-farmacia/, é idêntica à anterior com a diferença que se aplica às ciências farmacêuticas.

[74] http://nilc.icmc.usp.br/nilc/tools/nilctaggers.html

[75] http://www.nilc.icmc.usp.br/

2.7.17 Sílabas-PT

Ferramenta, escrita em Java por Hugo Gonçalo Oliveira[76], que tem como objectivo a separação das sílabas de uma palavra dada como *input*.

2.7.18 Smell

Ferramenta desenvolvida pelo LABEL [77] disponível online em http://label.ist.utl.pt/pt/smell_intr_pt.php utiliza recursos tais como léxicos e gramáticas formalizadas na identificação de entidades nomeadas.

A anotação das entidades nomeadas é feita recorrendo ao sistema Unitex.

Este serviço ainda se encontra em fase de teste.

2.7.19 TeP 2.0 Beta

Ferramenta disponível online em http://www.nilc.icmc.usp.br/tep2/index.htm que pretende disponibilizar um thesaurus para o português do Brasil.

As Figs. 42, 43 e 44 ilustram o funcionamento desta ferramenta.

Figura 42 - Ecrã inicial da interface do TeP 2.0 Beta.

[76] http://eden.dei.uc.pt/~hroliv/

[77] http://label.ist.utl.pt/

Figura 43 - Selecção da categoria grammatical.

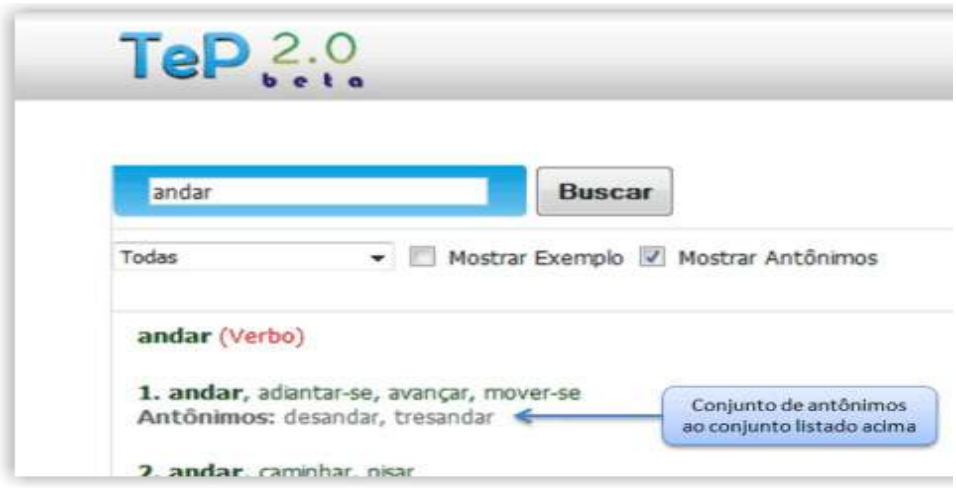


Figura 44 - Sinónimos do verbo *andar*.

2.7.20 Textcat

Ferramenta desenvolvida na Universidade de Groningen78, disponível sob licença LGPL, que identifica a língua na qual um dado documento, fornecido como input, se encontra escrito.

O exemplo 2.115 - Anexo VI - ilustra o funcionamento desta ferramenta.

[78] http://odur.let.rug.nl/~vannoord/

2.7.21 TextQuim

Package desenvolvido na UFRGS[79] e disponível online em http://www6.ufrgs.br/textquim/, sendo constituído pelas seguintes ferramentas:

2.7.21.1 *Concordanciador*

Esta ferramenta permite efectuar a pesquisa de contextos para palavras presentes no *corpus*.

O exemplo 2.116 - Anexo VI - ilustra o funcionamento desta ferramenta;

2.7.21.2 *lista de palavras*

Esta ferramenta permite efectuar a listagem das palavras existentes no corpus introduzido, organizada por ordem decrescente de frequência.

O funcionamento desta ferramenta pode ser verificado no exemplo 2.117, que se encontra no Anexo VI;

2.7.21.3 *n-gramas*

Esta ferramenta permite obter uma listagem de grupos de palavras repetidos ao longo do *corpus* introduzido como *input*.

O funcionamento desta ferramenta pode ser verificado no exemplo 2.118 - Anexo VI.

2.7.21.4 *concordanciador alinhado*

Esta ferramenta efectua a busca de contexto em corpora alinhados em inglês / português.

O funcionamento desta ferramenta é ilustrado no exemplo 2.119, que consta no Anexo VI.

[79] http://www6.ufrgs.br/

2.7.22 Unitex 2.0

Package de ferramentas e recursos, disponível para download sob as licenças GPL, LGPL e LGPLLR, desenvolvido pelo IGM[80], com o objectivo de se proceder à análise de textos em língua natural.

Os recursos disponíveis neste *package* consistem em dicionários electrónicos, gramáticas e tabelas lexico-gramaticais.

As ferramentas que se podem encontrar neste *package* são descritas em seguida de forma reduzida:

2.7.22.1 *BuildKrMwuDic*

Ferramenta que gera um grafo de dicionário de unidades multipalavra a partir de um dicionário;

2.7.22.2 *Cassys*

Ferramenta que aplica uma lista ordenada de gramáticas a um texto e constrói um índice de ocorrências encontradas;

2.7.22.3 *CheckDic*

Ferramenta responsável pela verificação do formato do dicionário que pode ser DELAS[81] ou DELAF[82];

2.7.22.4 *Compress*

Ferramenta responsável por efectuar a compressão de um dicionário do tipo DELAF;

[80] http://www-igm.univ-mlv.fr/

[81] DELA de formes Fléchies, DELA of inflected forms

[82] DELA de formes simples, simple forms DELA

2.7.22.5 *Concord*

Ferramenta responsável por produzir uma concordância a partir de um ficheiro gerado pelo programa Locate descrito mais à frente;

2.7.22.6 *ConcorDiff*

Ferramenta que recebe dois ficheiros de concordâncias e produz uma página HTML mostrando as diferenças;

2.7.22.7 *Convert*

Ferramenta que efectua a codificação de fecheiros de texto;

2.7.22.8 *Dico*

Ferramenta que aplica dicionários a um texto previamente partido em unidades lexicais pelo programa *Tokenize*;

2.7.22.9 *Elag*

Ferramenta que recebe um autómato de texto aplicando-o na remoção de regras de ambiguidade;

2.7.22.10 *ElagComp*

Ferramenta responsável pela compilação de gramáticas que serão posteriormente utilizadas pelo programa *Elag*;

2.7.22.11 *Evamb*

Ferramenta responsável pelo cálculo da taxa média de ambiguidade do autómato de texto especificado ou numa dada frase;

2.7.22.12 *Extract*

Ferramenta que extrai de um texto todas as frases que contêm pelo menos uma ocorrência de concordância;

2.7.22.13 *Flatten*

Ferramenta que transforma uma gramática num tradutor de estados finais;

2.7.22.14 *Fst2Check*

Ferramenta responsável por verificar se um ficheiro possui ou não erros para que possa ser utilizado pelo programa *Locate*;

2.7.22.15 *Fst2List*

Ferramenta que apresenta as sequências de caracteres reconhecidas pela gramática dada como *input*;

2.7.22.16 *Fst2Txt*

Ferramenta responsável por aplicar um tradutor a um texto na fase de pré-processamento, antes de o texto ser partido em unidades lexicais;

2.7.22.17 *Grf2Fst2*

Ferramenta responsável pela compilação da gramática dada como *input*;

2.7.22.18 *ImplodeFst2*

Ferramenta responsável pela elaboração compacta do autómato de texto dado como *input;*

2.7.22.19 *Locate*

Ferramenta que aplica uma gramática a um texto e constrói um ficheiro indexado das ocorrências encontradas;

2.7.22.20 *LocateTfst*

Ferramenta que aplica uma gramática a um autómato de texto, e guarda o índice da sequência emparelhada pelo autómato num ficheiro com o nome concord.ind;

2.7.22.21 *MultiFlex*

Ferramenta responsável pela inflexão automática de um dicionário do tipo DELA que contém lemas de palavras simples ou compostas;

2.7.22.22 *Normalize*

Ferramenta que tem por objectivo executar a normalização dos separadores de texto;

2.7.22.23 *PolyLex*

Ferramenta que recebe um ficheiro que contém uma lista de palavras desconhecidas e tenta analisar cada uma como uma palavra composta obtida pela junção de palavras simples;

2.7.22.24 *RebuildTfst*

Ferramenta que re-constrói o autómato de texto levando em consideração as alterações manuais;

2.7.22.25 *Reconstrucao*

Ferramenta que gera uma gramática de normalização para ser aplicada antes da construção de um autómato para um texto escrito em Português;

2.7.22.26 *Reg2Grf*

Ferramenta que constrói um ficheiro correspondente à expressão regular escrita no ficheiro de *input*;

2.7.22.27 *SortTxt*

Ferramenta que procede à ordenação lexicográfica das linhas do ficheiro dado como *input*;

2.7.22.28 *Stats*

Ferramenta que calcula algumas estatísticas a partir do ficheiro indexado de concordâncias dado como *input*;

2.7.22.29 *Table2Grf*

Ferramenta que gera grafos automaticamente a partir de uma gramática lexical e de um grafo modelo;

2.7.22.30 *Tagger*

Ferramenta que recebe como input um autómato de texto e procede à sua linearização;

2.7.22.31 *TagsetNormTfst*

Ferramenta que executa a normalização do autómato fornecido como input de acordo com o ficheiro de tag's especificado, ignorando códigos de dicionário não declarados e entradas lexicais incoerentes;

2.7.22.32 *TEI2Txt*

Ferramenta responsável pela geração de um ficheiro de texto a partir de um ficheiro XML dado como *input*;

2.7.22.33 *Tfst2Grf*

Ferramenta que produz um autómato de frases a partir de um autómato de texto dado como *input*;

2.7.22.34 *Tfst2Unambig*

Ferramenta que recebe um autómato de texto como *input* e produz um ficheiro de texto equivalente se o autómato for linear;

2.7.22.35 *Tokenize*

Ferramenta responsável por separar em unidades lexicais um ficheiro de texto dado como *input*;

2.7.22.36 *TrainingTagger*

Ferramenta que gera dois ficheiros de dados de etiquetação a partir de um corpus previamente etiquetado dado como *input*;

2.7.22.37 *Txt2Tfst*

Ferramenta que constrói um autómato de texto a partir do ficheiro de texto dado como *input*;

2.7.22.38 *Uncompress*

Ferramenta que recebe como *input* um dicionário com a extensão .bin e converte-o num ficheiro de texto com a extensão .dic;

2.7.22.39 *Untokenize*

Ferramenta responsável pela reconstrução do texto original a partir da lista de tokens gerada pelo programa *Tokenize*;

2.7.22.40 *UnitexTool*

Ferramenta que permite a chamada de todas as outras ferramentas descritas nesta secção;

2.7.22.41 *UnitexToolLogger*

Ferramenta responsável pela manutenção do histórico de operações executadas;

2.7.22.42 *XMLizer*

Ferramenta que recebe como *input* um ficheiro de texto e cria o ficheiro correspondente do tipo TEI ou do tipo XML.

2.7.23 UNL

O UNL - Universal Networking Language - é uma metalinguagem desenvolvida por uma equipa do NILC liderada por Maria das Graças Volpe Nunes[83], disponível online em http://www.nilc.icmc.usp.br/nilc/projects/unl.htm, com o objectivo de descrever os aspectos informais do significado de frases.

Esta metalinguagem é constituída por:

- um dicionário de palavras universais (UWs);
- um conjunto de etiquetas de relações binárias (RLs) existentes entre pares de frases ou pares de UWs;
- um conjunto de etiquetas de atributos (Als) que especificam valores particulares de características gramaticais de UWs;
- fórmulas do tipo RL(UW1,UW2);

A Figura 45 ilustra a arquitectura desta ferramenta.

[83] http://www.icmc.usp.br/~gracan/

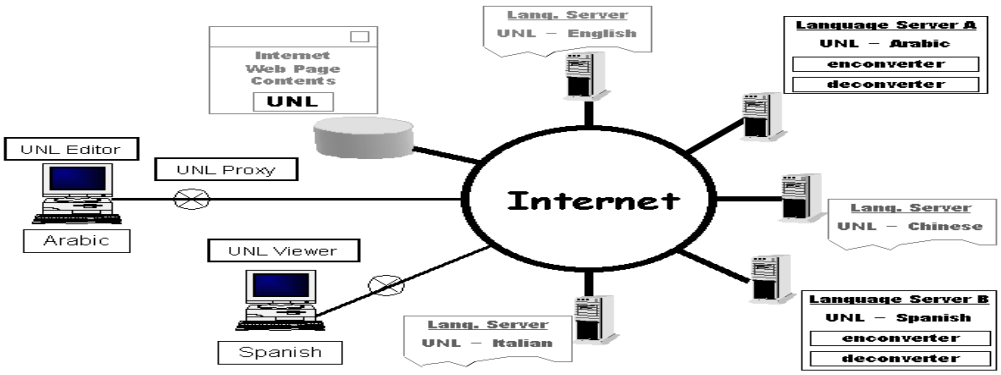


Figura 45 - Arquitectura da ferramenta UNL.

2.7.24 WordNetBr

Ferramenta desenvolvida pelo NILC [84] disponível online em http://www.nilc.icmc.usp.br/~carol/wn.html que tem como objectivo a criação de uma base de dados de palavras para o Português do Brasil.

Esta base de dados é constituída por:

- um *corpus* baseado em frases para cada palavra ou forma contida no dicionário de sinónimos;
- um glossário de sinónimos;
- hierarquias independentes da linguagem, semânticas e relações relevantes de hiperonímia[85], hiponímia[86] e meronímia[87];

[84] http://www.nilc.icmc.usp.br/

[85] Relação semântica de inclusão entre uma unidade lexical mais genérica (hiperónimo) e outra mais específica (hipónimo), em que esta é dependente semanticamente da primeira

[86] Relação semântica que se estabelece entre significados de itens lexicais, em que um deles, mais específico (hipónimo), está incluída noutro, mais geral (hiperónimo)

[87] Relação semântica entre duas unidades lexicais, em que uma indica uma parte (merónimo) relativamente à outra, que indica o todo (holónimo)

2.8 PROCESSAMENTO DE FALA

2.8.1 Dixi

Sintetizador de fala a partir de texto escrito em Português disponível online em http://www.speech.inesc.pt/~lco/dixi/dixi_pt.cgi desenvolvido pelo Grupo de Processamento de Fala do INESC[88] em colaboração com o Grupo de Fonética e Fonologia do CLUL[89].

2.8.2 Info-Maker

Ferramenta desenvolvida pelo Instituto de Telecomunicações da Universidade de Coimbra[90] disponível online em http://www.it.uc.pt/signal/sait/InfoMaker.htm e que permite efectuar o reconhecimento de voz e dígitos DTMF.

A Figura 46 ilustra a definição de um serviço de informações.

Figura 46 - Exemplo de um serviço de informações.

[88] https://www.l2f.inesc-id.pt/wiki/index.php/Main_Page

[89] http://www.clul.ul.pt/index.php

[90] http://www.it.uc.pt/

2.8.3 Lingua-PT-Speaker

Ferramenta escrita em Perl por Alberto Simões[91] que tem como objectivo a produção do discurso oral do Português.

2.8.4 Páginas Falantes

Um serviço telefónico, disponível online em http://www.it.uc.pt/signal/sait/falantes.htm, desenvolvido pelo Instituto de Telecomunicações da Universidade de Coimbra[92,] que contém grandes quantidades de informação. O seu funcionamento é semelhante ao tele-texto, ou seja, será pedido ao interlocutor o número da página de informação desejada, e este poderá fornecer esse número por digitação de DTMF's ou por voz.

A Figura 47 ilustra o funcionamento desta ferramenta.

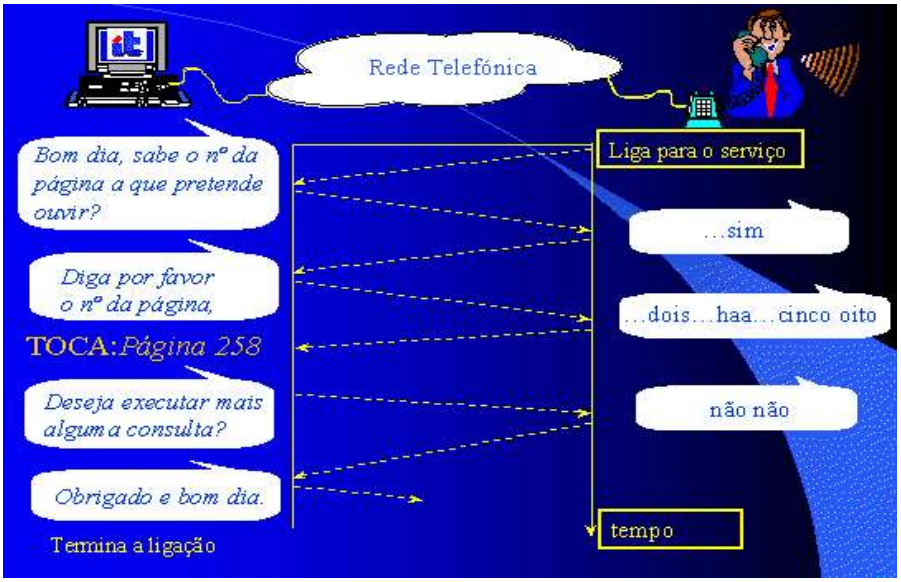


Figura 47 - Arquitetura do sistema Páginas Falantes.

[91] http://search.cpan.org/~ambs/

[92] http://www.it.uc.pt/

2.8.5 SVITD

Sintetizador de números de telefone em Português, desenvolvido pelo Laboratório de Sistemas de Linguagem Falada do INESC [93] e disponível online em http://www.speech.inesc.pt/~lco/svit/svitd_pt.cgi.

A Figura 48 ilustra o funcionamento desta ferramenta.

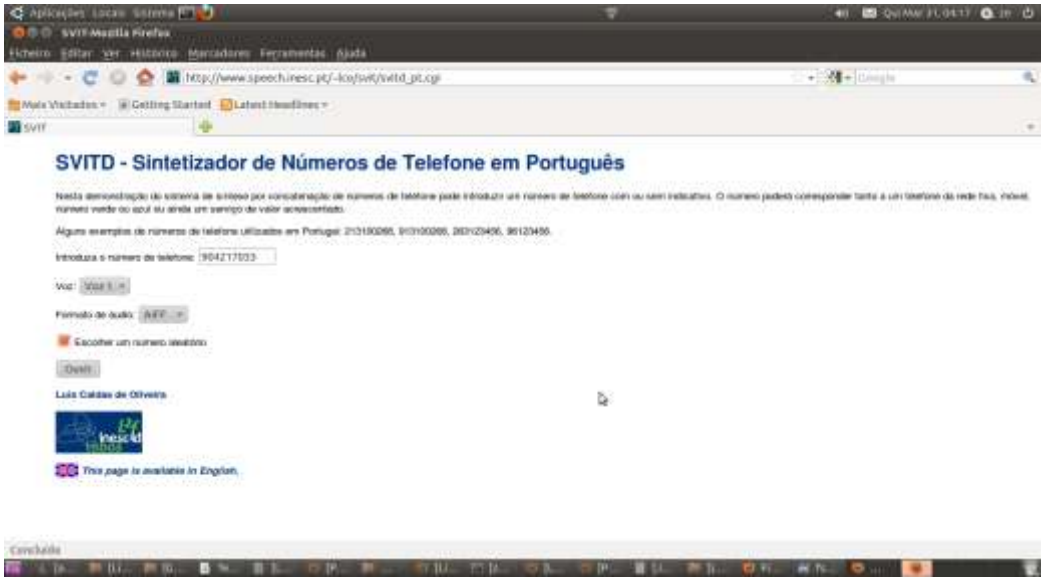


Figura 48 - Ilustração do funcionamento do Sintetizador de Números de Telefone em Português.

2.8.6 Tele-Balcão

Sistema de pesquisa desenvolvido pelo Instituto de Telecomunicações da Universidade de Coimbra[94] e disponível online em http://www.it.uc.pt/signal/sait/telebal.htm, destinado à venda de produtos.

Permite navegação por digitação de DTMF's ou através de reconhecimento de voz.

A Figura 49 ilustra a estrutura deste sistema.

[93] https://www.l2f.inesc-id.pt/wiki/index.php/Main_Page

[94] http://www.it.uc.pt/

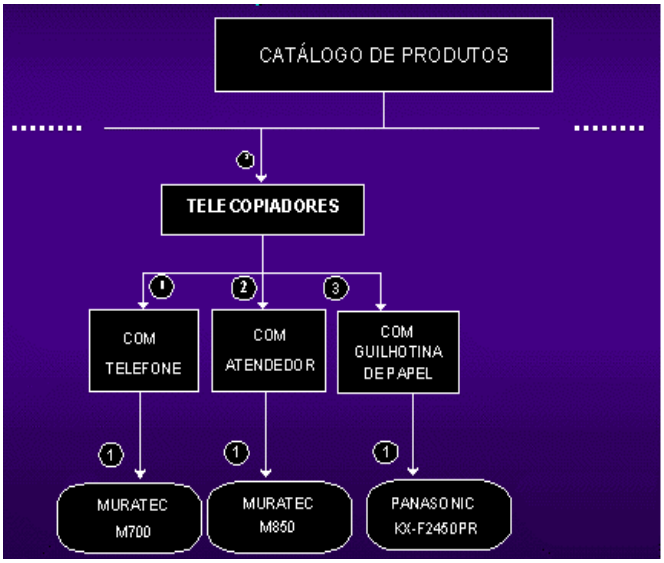


Figura 49 - Arquitectura do sistema Tele-Balcão.

2.8.7 Voice Mail

Sistema de reconhecimento de voz e dígitos DTMF desenvolvido pelo Instituto de Telecomunicações da Universidade de Coimbra [95], disponível online em http://www.it.uc.pt/signal/sait/voicemail.htm.

A Figura 50 mostra o funcionamento desta ferramenta.

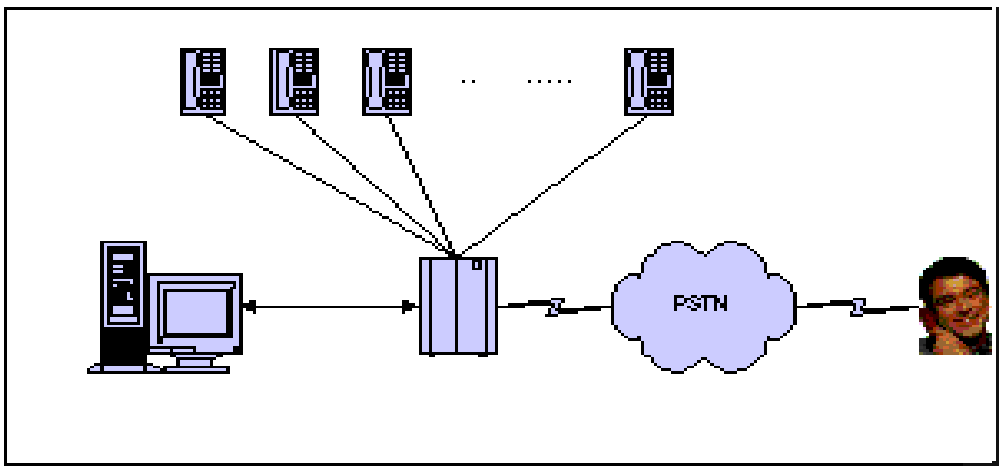


Figura 50 - Arquitectura do sistema de Voice Mail.

[95] http://www.it.uc.pt/

2.8.8 Web Wake Up

Serviço de despertar activado pela Internet, desenvolvido pelo Instituto de Telecomunicações da Universidade de Coimbra [96] e disponível online em http://www.it.uc.pt/signal/sait/Despertar.htm.

2.9 SUMARIZADORES

2.9.1 Explosa

Ferramenta desenvolvida na UFSCar pela equipa liderada por Lucia Helena Machado Rino, docente do Departamento de Computação da Universidade Federal de São Carlos[97], e disponível online em http://www2.dc.ufscar.br/~lucia/PROJECTS/EXPLOSA.htm, que utiliza métodos de sumarização automática para textos escritos em Português do Brasil.

2.9.2 GistSumm

Ferramenta desenvolvida pelo NILC[98] que tem como objectivo a produção de um resumo do texto dado como input.

A produção do resumo acima referida baseia-se em dois pressupostos:

- todo o texto veicula uma ideia principal;
- é possível identificar num texto uma frase que melhor exprime a sua ideia principal.

[96] http://www.it.uc.pt/

[97] http://www2.dc.ufscar.br/~lucia/

[98] http://nilc.icmc.usp.br/nilc/

2.10 TRADUÇÃO AUTOMÁTICA

2.10.1 Apertium machine translation engine and tools

Ferramenta desenvolvida pelo grupo de investigação Transducens[99] do Departamento de Linguagens e Sistemas Informáticos da Universidade de Alicante[100] em colabotação com Prompsit Language Engineering[101].

Esta ferramenta encontra-se disponível online em http://xixona.dlsi.ua.es/apertium-www/ e para download sob licença GPL, e tem como objectivo a tradução entre diversos idiomas de textos fornecidos como input.

O exemplo 2.120 que ilustra a utilização desta ferramenta pode ser consultado no Anexo VII.

2.10.2 EPT-Web

Ferramenta desenvolvida pelo NILC [102] e disponível online em http://www.nilc.icmc.usp.br/nilc/projects/ept-web.htm que tem como objectivo realizar a tradução automática de páginas Web utilizando a UNL - Universal Networking Language.

2.10.3 Galician-Portuguese translator

Ferramenta desenvolvida por Pablo Gamallo Otero[103] que permite translinear e traduzir texto entre Português e Galego.

O exemplo 2.121, que pode ser consultado no Anexo VII, ilustra o funcionamento desta ferramenta.

[99] http://transducens.dlsi.ua.es/
[100] http://www.dlsi.ua.es/
[101] http://www.prompsit.com/pt/
[102] http://www.nilc.icmc.usp.br/
[103] http://gramatica.usc.es/~gamallo/

2.10.4 PUL0

Ferramenta desenvolvida pelo NILC [104] e disponível online em http://www.nilc.icmc.usp.br/nilc/projects/LIBRAS2.htm, que pretende realizar a tradução entre uma língua oral tal como o Português e a linguagem gestual.

[104] http://www.nilc.icmc.usp.br/nilc/

3 DESCRIÇÃO DO PORTAL DE FERRAMENTAS PARA PROCESSAMENTO DE LÍNGUA PORTUGUESA

Para concretizar a construção do portal de ferramentas de processamento da língua portuguesa foi escolhido o CMS Joomla! pelas seguintes razões:

- possuir um sistema de administração de publicidade fácil de gerir;
- disponibilizar um sistema de ajuda incorporado;
- facilitar a pesquisa através do site através de um motor de busca built-in;
- permitir a distribuição de conteúdo através do mecanismo de RSS[105];
- permitir a criação, publicação, reordenação e edição de artigos num clique;
- facilitar a criação de um calendário de publicação;
- disponibilizar um sistema de fácil programação de exibição de conteúdos;
- possuir um conjunto de motores geridos por uma base de dados banco de dados;
- disponibilizar as funcionalidades de envio de emails e impressão de artigos em formato pdf;
- possuir uma interface gráfica administrativa de fácil utilização;
- possuir um editor de texto semelhante ao WordPad;
- permitir a facilidade de gestão e edição de notícias, produtos ou secções de serviços;
- disponibilizar um Trash Manager("Lixeira");
- facilitar a gestão e upload de mídia (imagens e documentos);
- possuir um motor de busca de URL's amigável através do mecanismo de SEF;
- permitir a edição de conteúdos utilizando editores WYSIWYG;
- facilitar a realização de pesquisas simples.

3.1 DESCRIÇÃO DO CMS JOOMLA!

O Joomla! é um CMS *open-source,* disponível sob licença GPL, que funciona sob qualquer plataforma na qual estejam instalados o servidor de páginas Web (Apache[106] ou MS-IIS), o

[105] RSS é um recurso desenvolvido em XML que permite aos responsáveis por sites e blogs divulgarem notícias ou novidades acerca destes.

[106] http://www.apache.org/

servidor de bases de dados MySql[107] e a linguagem de programação PHP[108], e permite não só a rápida elaboração de um site, bem como a gestão do respectivo conteúdo.

A arquitectura deste CMS é ilustrada na figura abaixo.

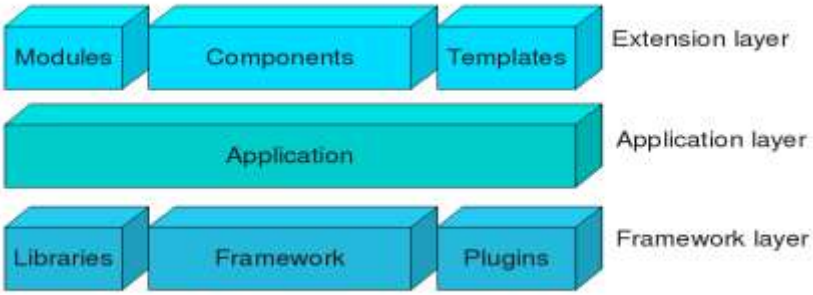


Figura 51 - Arquitectura do CMS Joomla!

Como se pode observar pela Figura 51 o CMS Joomla! é constituído por três camadas:

- extensão - camada constiuída por três tipos de extensões:
  - componentes - aplicações do Joomla! Responsáveis por mostrar conteúdos;
  - módulos - pequenos blocos de códigos que podem ser colocados em qualquer lugar do template e não dependem de nenhuma acção do utilizador para serem mostrados;
  - *templates* - modelos ou *layouts* de um site – sem conteúdo. Um template Joomla! é composto por arquivos PHP, HTML, XML, CSS e imagens que, combinadas, definem o visual do *site* e a configuração da exibição das colunas, cores, fontes e parágrafos.

    Quando necessário, o *layout* pode ser modificado sem que se altere o conteúdo que fica armazenado na base de dados.

    Quando se carrega num *link*, o respectivo conteúdo é pesquisado na base de dados e exibido no *template* que faz a sua formatação.

- aplicação - camada constituída pelas aplicações que derivam da classe **Japplication**

[107] http://www.mysql.com/

[108] http://www.php.net/

- **JInstallation** - classe responsável pela instalação do CMS Joomla! num servidor Web e que deverá ser eliminada assim que o processo de instalação é concluído.

  O processo de instalação do CMS Joomla! pode ser consultado no Anexo VIII.

- **JAdministrator** - classe responsável pelo administrador de *backend*[109].

A Figura 52 ilustra o *backend* do Joomla!

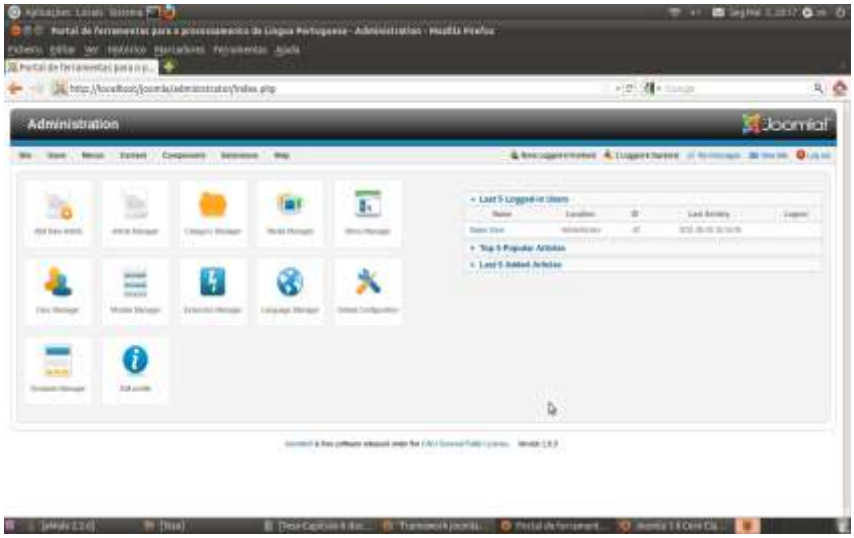


Figura 52 - Vista de administrador do Joomla! – *backend*-

Como se pode observar pela Figura 52 o *backend* está dividido em duas partes.

Na metade do lado esquerdo encontram-se os menus que permitem ao administrador realizar a administração do *site*.

Na metade do lado esquerdo encontram-se os menus que permitem ao administrador realizar a administração do site:

- adição de novos artigos (Figura 53);

[109] Visão do administrador. É no *backend* que se efectuam as operações de configuração, manutenção, limpeza, criação de estatísticas e de conteúdos.

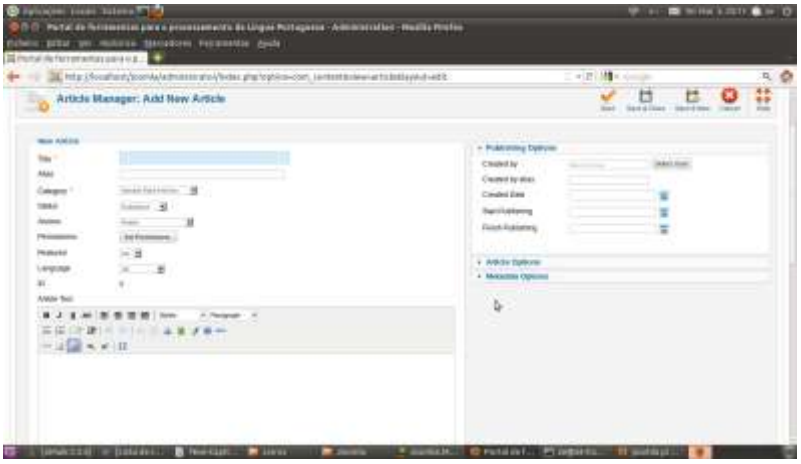


Figura 53 - Adição de um novo artigo.

– gestor de artigos (Figura 54);

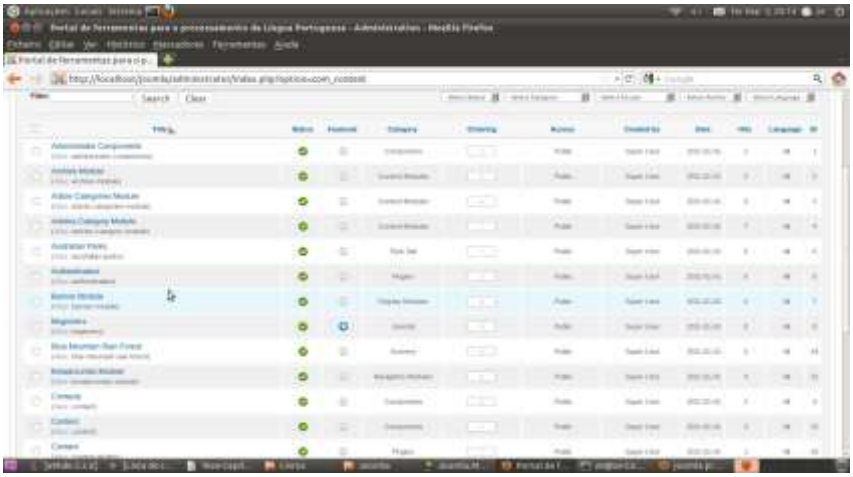


Figura 54 – Gestor de artigos.

– gestor de categorias (Figura 55);

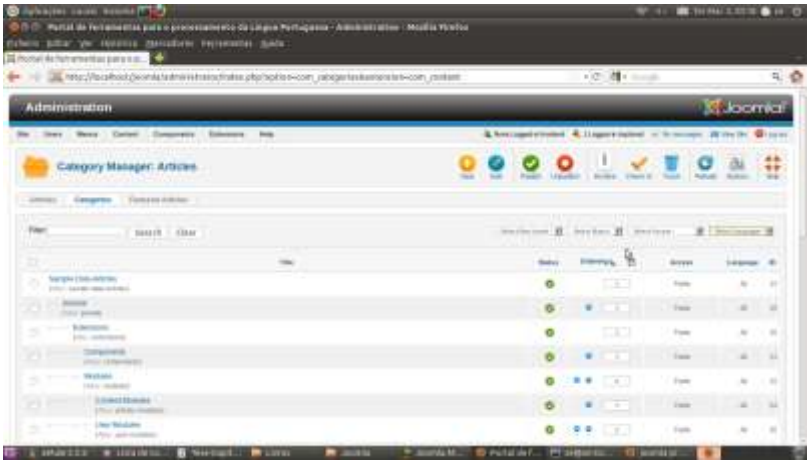


Figura 55 - Gestor de categorias.

– gestor de media (Figura 56);

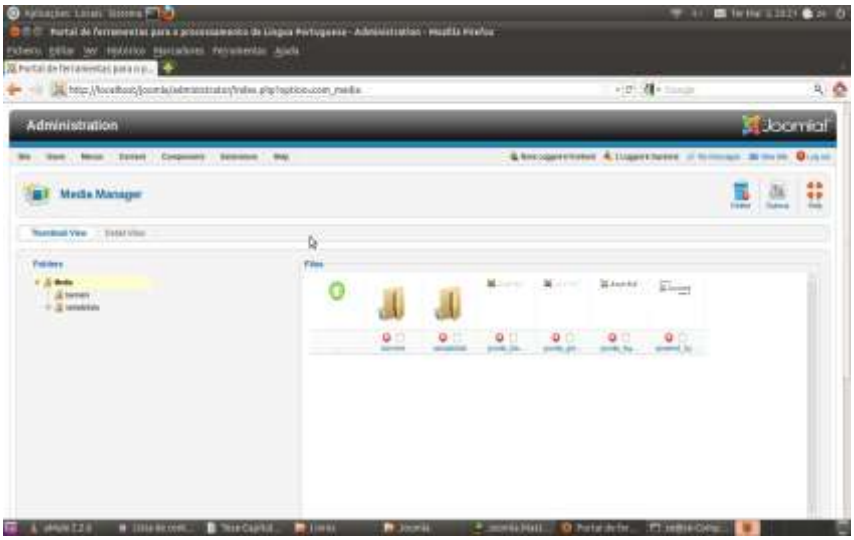


Figura 56 - Gestor de media.

– gestor de menus (Figura 57);

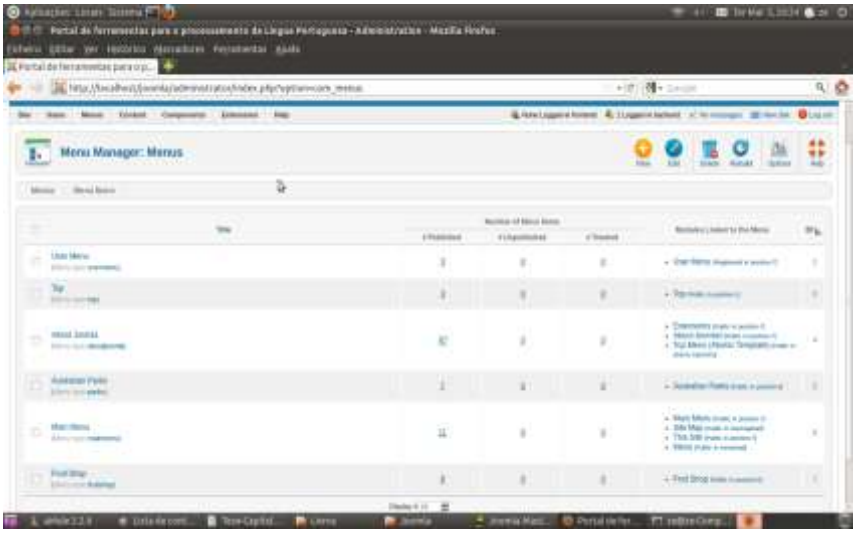


Figura 57 - Gestor de menus.

– gestor de utilizadores (Figura 58);

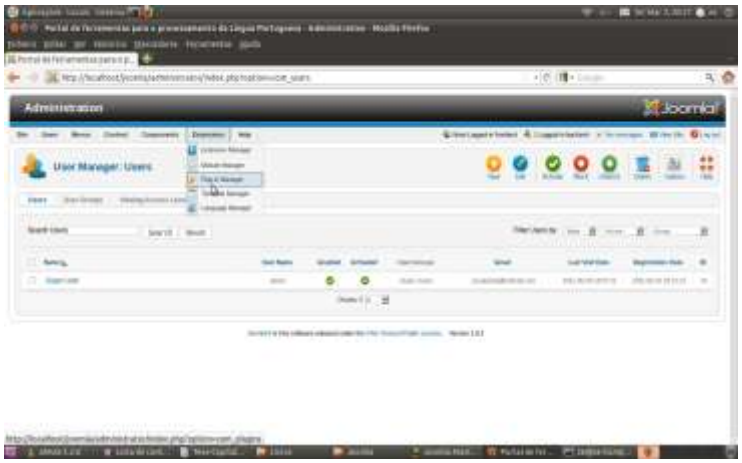


Figura 58 - Gestor de utilizadores.

– gestor de módulos (Figura 59);

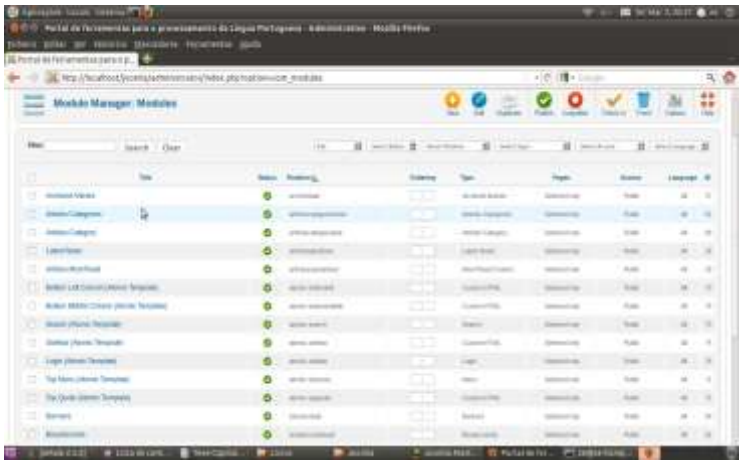


Figura 59 - Gestor de módulos.

– gestor de extensões (Figura 60);

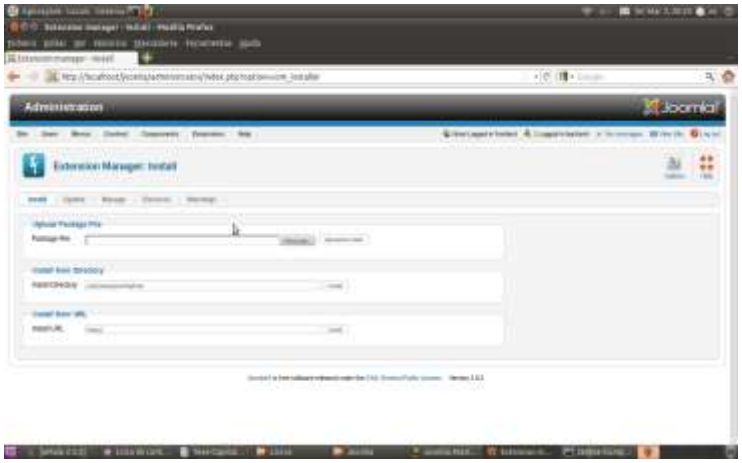


Figura 60 - Gestor de extensões

– gestor de idiomas (Figura 61);

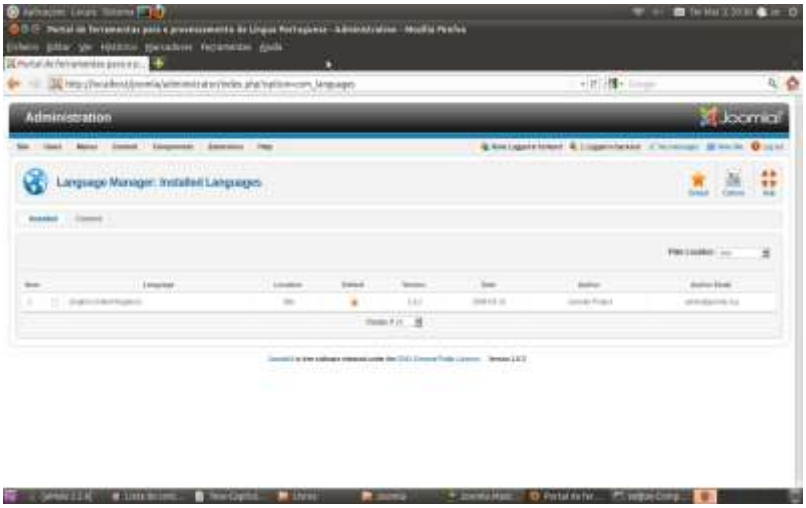


Figura 61 – Gestor de idiomas.

– menu de configuração geral (Figura 62);

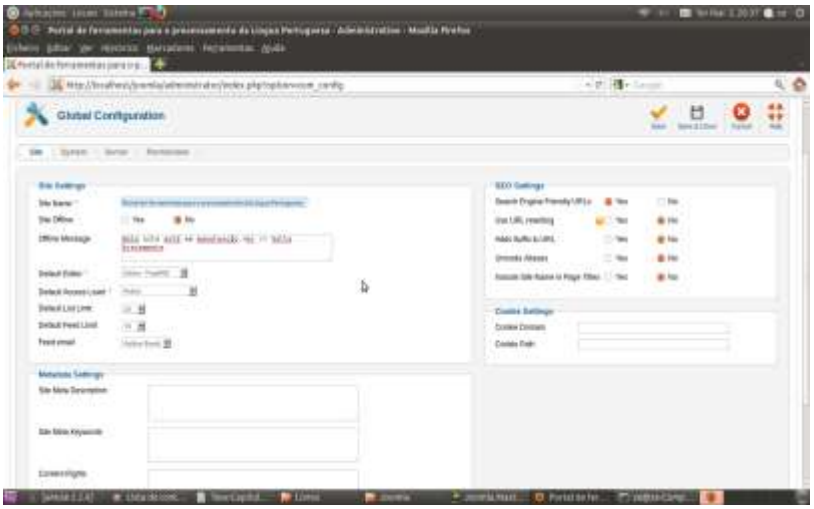


Figura 62 - Menu de configuração geral.

– gestor de *templates* (Figura 63);

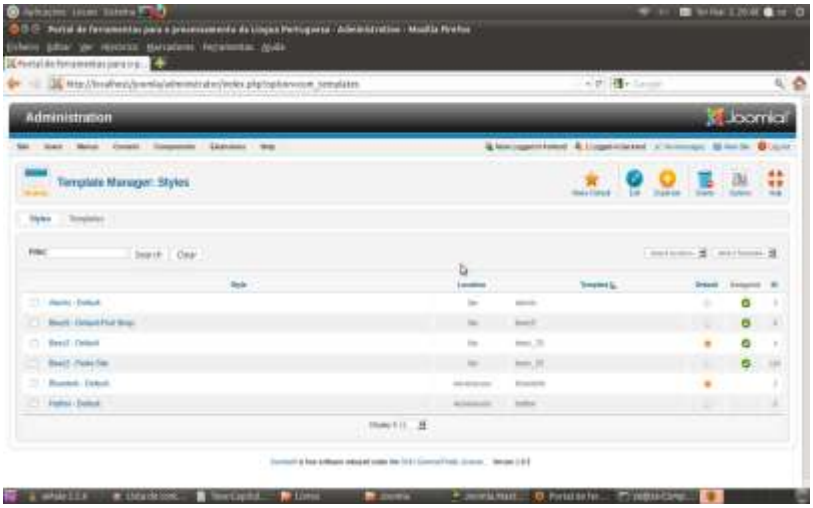


Figura 63 - Gestor de templates.

– editor de profile (Figura 64).

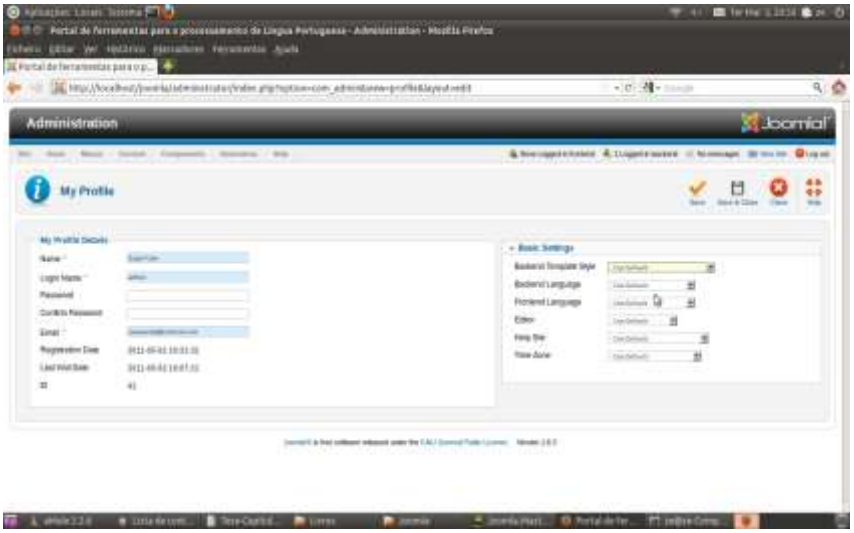


Figura 64 - Editor de *profile*.

Na metade do lado direito podemos ver a informação acerca de quem está autenticado no site, quais os artigos mais populares e os que foram adicionados recentemente.

- **JSite** - classe responsável pelo front-end[110] do site.

  A Figura65 ilustra o *frontend* do Joomla!.

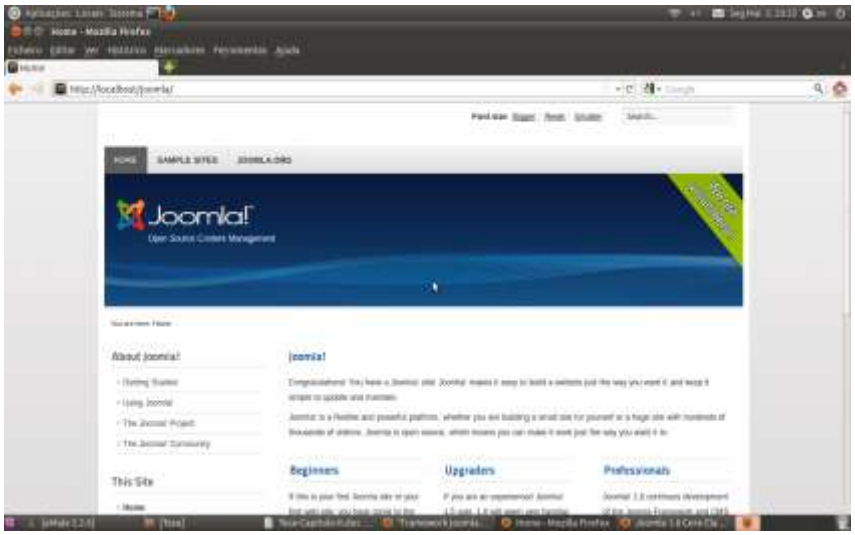


Figura 65 - Visão de utilizador do Joomla! – *frontend*.

[110] Visão do utilizador, ou seja aquilo que os visitantes e os utilizadores autenticados vêem.

- **XML-RPC** - classe responsável pela administração remota do website criado com o CMS Joomla!.

– *framework* - camada constituída por:

- *libraries* - "*packages*" de código que contêm funções de auxílio à *framework* do Joomla! ou às suas extensões;
- *framework*[111] do Joomla! - é a parte mais importante de toda a arquitectura do Joomla!, uma vez que é da sua responsabilidade a manutenção e extensão deste CMS.

Os pacotes que integram a *framework* do Joomla! são os seguintes:

- pacote aplicativo que gere a cama de aplicação e contém a classe *JApplication*;
- pacote com as bibliotecas de cache[112];
- pacote denominado de "comum", que contém as classes básicas e as bibliotecas de compatibilidade com versões anteriores;
- pacote conector com bibliotecas para clientes como FTP e LDAP;
- pacote para a base de dados que contém a classe JDatabase e as bibliotecas necessárias;
- pacote documento com as bibliotecas necessárias para criar e apresentar páginas;
- pacote do sistema de arquivos com bibliotecas para interagir com o *filesystem*;
- pacote **i18n** com bibliotecas de internacionalização (idiomas);
- pacote de instalação com bibliotecas que auxiliam a instalação de extensões (componentes, módulos, plugins, templates, etc);
- pacote de e-mail com bibliotecas;
- pacote modelo com bibliotecas objecto de acesso a dados;
- pacote de parâmetros com bibliotecas para a manipulação de parâmetros;
- pacote de registo (*registry*) com bibliotecas de armazenamento de configurações;

[111] Abstracção que une códigos comuns entre vários projectos de software fornecendo uma funcionalidade genérica.

[112] Dispositivo de acesso rápido, interno a um sistema, que serve de intermediário entre um operador de um processo e o dispositivo de armazenamento ao qual esse operador acede.

- pacote de modelos com bibliotecas de *templates*;
- pacote de utilidades com diversas bibliotecas de uso geral;
- classe **JFactory** que permite instanciar os objectos da *framework*;
- classe **JVersion** que permite obter a versão do Joomla!.

– *plugins* - secções de código que é executado quando ocorre um evento pré-definido no Joomla!.

3.2 DESCRIÇÃO DO SITE

Após o levantamento das ferramentas existentes para o processamento da língua portuguesa, procedeu-se ao desenvolvimento do site que se encontra alojado em http://saianda.xdi.uevora.pt/joomla.

O desenvolvimento deste site está dividido em duas partes:

– *BackEnd;*
– *FrontEnd.*

3.2.1 Descrição do *BackEnd*

A elaboração deste *site* encontra-se está dividida em duas partes:

3.2.1.1 *Análise de documentos*

Esta funcionalidade permite ao utilizador, através de um formulário criado com o componente Chronoforms[113], inserir um documento em formato text/plain[114] e em seguida seleccionar a ferramenta pretendida para efectuar a análise que se quer fazer.

[113] http://www.chronoengine.com/

[114] Conteúdo de um arquivo comum sequencial, legível como material textual, sem muito processamento.

As Figs. 66 e 67 ilustram o processo de selecção de um documento e de uma ferramenta para o seu processamento.

Após a selecção do documento e da ferramenta pretendidos é chamado o ficheir .php correspondente à ferramenta em causa para efectuar o processamento sobre o documento seleccionado.

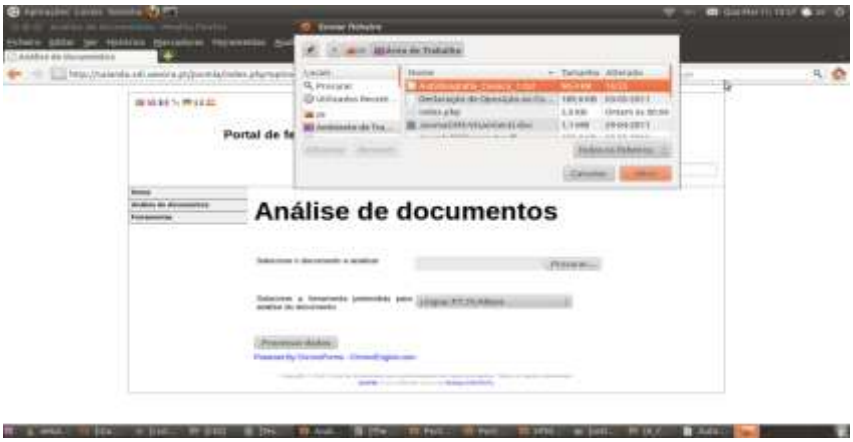


Figura 66 - Selecção de um documento para análise.

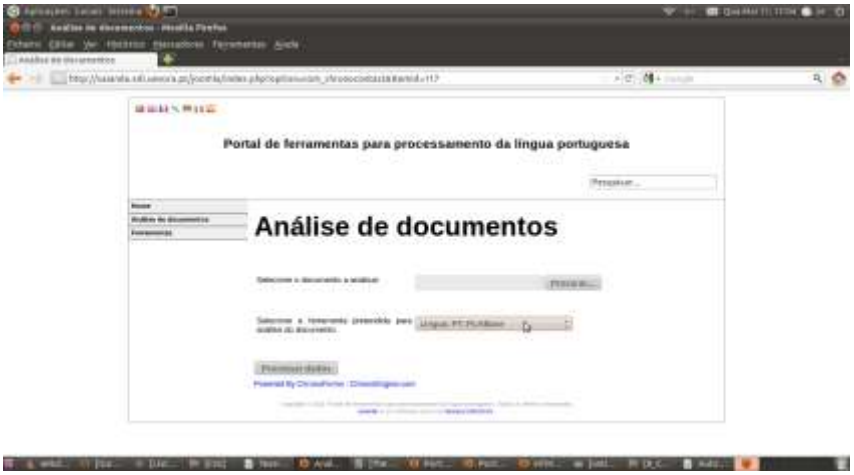


Figura 67 - Selecção da ferramenta pretendida para análise do documento introduzido.

3.2.1.2 *Ferramentas*

Esta opção permite o acesso às várias ferramentas existentes que se encontram agrupadas em diversas categorias (Ajuda ao ensino, Ajuda à redacção, Alinhadores, Analisadores, Conjugadores verbais, Extractores de N-Gramas, Ferramentas especializadas, Processamento de fala e Tradução automática) conforme a função que desempenham.

Este agrupamento das diversas ferramentas é ilustrado pela Figura 68.

Figura 68 - Menu de ferramentas existentes no *site.*

Para cada ferramenta foi criado um artigo no CMS Joomla!, como se mostra nas Figs.69 e 70.

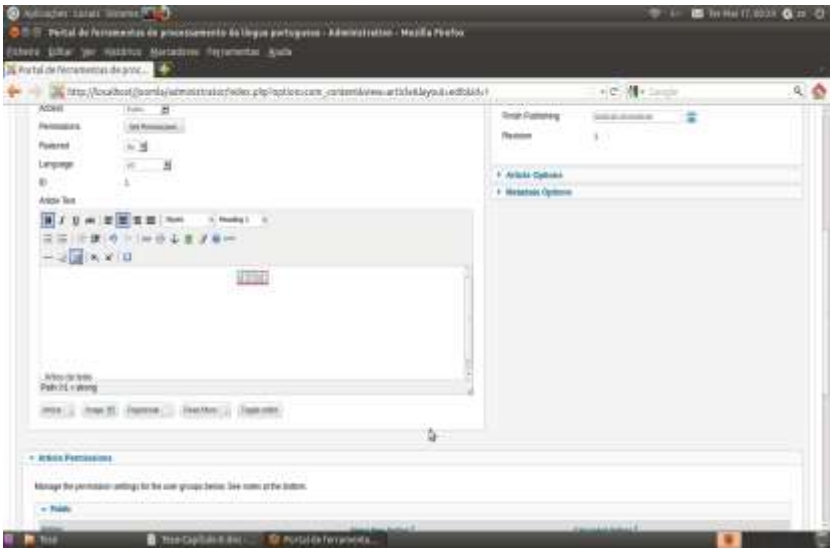


Figura 69 - Acesso ao gestor de artigos do CMS Joomla!

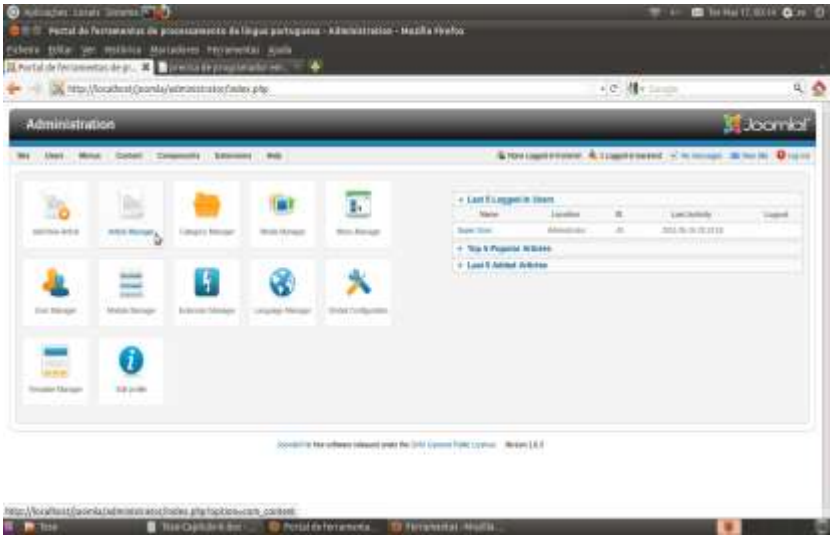


Figura 70 - Criação de um artigo no gestor de artigos do CMS Joomla!

Em cada artigo é apresentada a descrição de cada ferramenta, o tipo de licença a que está sujeita e ainda a sua autoria.

Nos casos em que se encontra acessível apenas *online* é apresentada no artigo uma hiperligação para o *site* que lhe dá acesso.

Quando se trata de uma ferramenta que se encontra disponível para *download* apresenta-se a hiperligação e, sempre que possível uma segunda hiperligação para o *link* onde poderá ser consultada *online*.

De modo a que o utilizador possa navegar através do site foi criado um menu inicial de navegação através do gestor de menus que se apresenta nas Figs. 71 e 72.

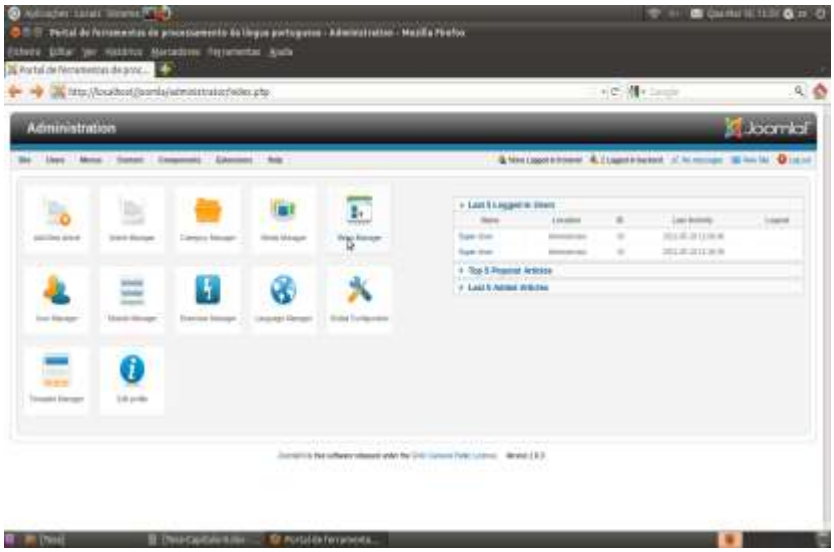


Figura 71 - Acesso ao gestor de menus.

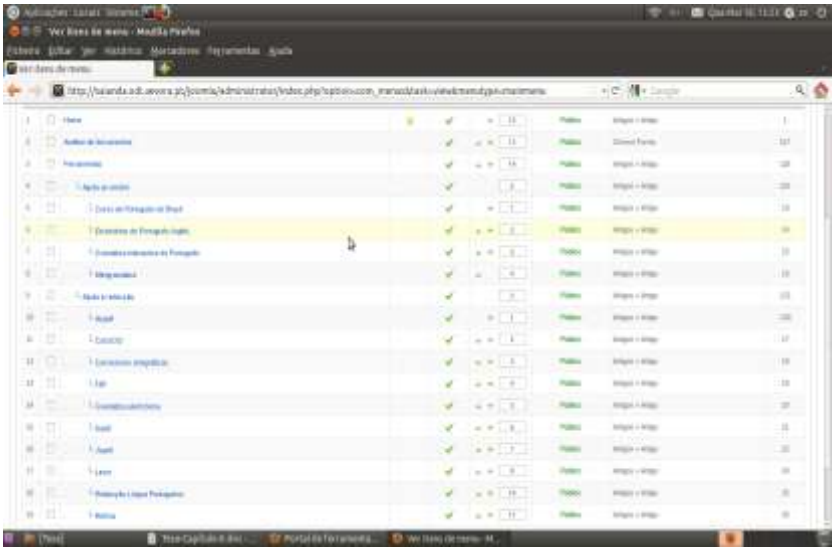


Figura 72 - Elementos que compõem o menu de acesso ao *site*.

Como é possível observar pela Figura 72 foram criados na raíz do menu três items que constituem as opções principais:

– Home - que permite aceder à página inicial do *site*;

– Análise de documentos - que permite introduzir um documento para análise e em seguida escolher a ferramenta adequada para a análise pretendida;

– Ferramentas - que permite o acesso às ferramentas existentes no site como se mostra na Figura 73.

Figura 73 - Acesso ao menu de ferramentas existentes no portal.

Ao aceder a este menu, é apresentado um segundo menu, conforme se mostra na Figura 74, no qual cada item corresponde a uma hiperligação para cada uma das categorias criadas e anteriormente descritas.

Figura 74 - Sub-menu com as categorias de ferramentas existentes no portal.

A partir do sub-menu apresentado na Figura 75 é possível seleccionar a ferramenta pretendida como se mostra na Figura 76.

Figura 75 - Selecção de uma determinada ferramenta a partir do sub-menu anterior.

Figura 76 - Execução de uma determinada ferramenta seleccionada a partir do sub-menu anterior.

3.2.2 Descrição do *FrontEnd*

O acesso ao endereço acima referido possibilita-nos a obtenção do seguinte *screen*:

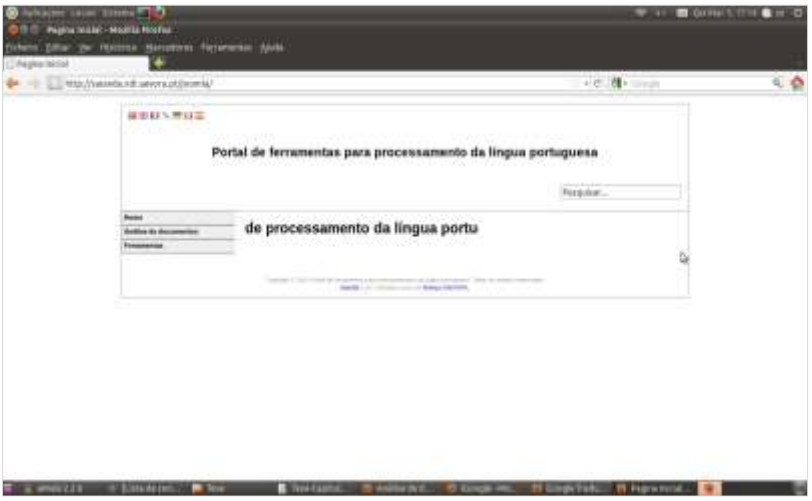


Figura 77 - *Screen* inicial do portal de ferramentas para o processamento da língua portuguesa.

Como se pode observar pela Figura 77 o *screen* inicial do site encontra-se dividido em três partes:

- cabeçalho - constituído por:
  - um grupo de bandeiras correspondentes a diversos idiomas no canto superior esquerdo;
  - o título do site ao centro;
  - um menu de pesquisa no canto inferior direito.

- corpo - local onde vai ser mostrada a informação pretendida pelo utilizador;

- rodapé - constituído pelos direitos de autor do *site.*

Para efectuar a elaboração do template mostrado na Figura 77 foi necessário criar um directório com o nome da *template*, neste caso, 'Mestrado', dentro do directório '*templates*' que se encontra contido no directório de instalação do CMS Jooma!. No nosso caso o referido directório é '/var/www/joomla', como mostra a Figura 78.

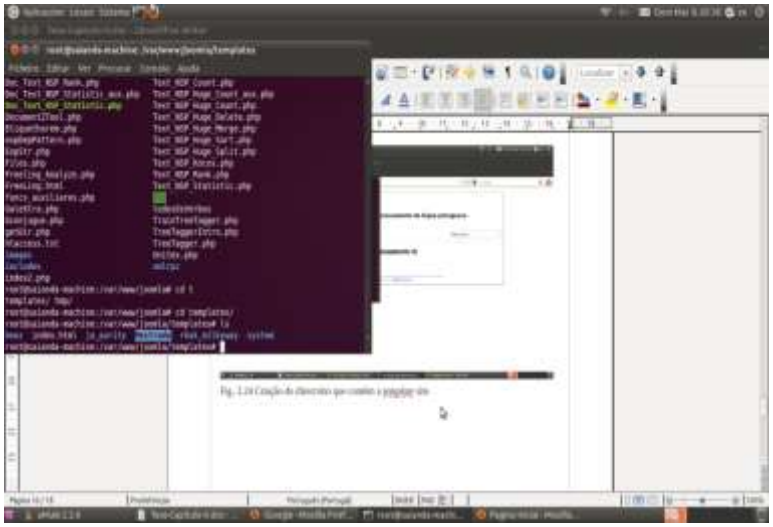


Figura 78 - Criação do directório que contém a *template* do *site*.

Após a criação do directório acima referido foi necessário criar dentro dele quatro ficheiros que a seguir apresentamos:

- templateDetails.xml - este ficheiro é responsável pela instalação da template, que no nosso caso apresenta a configuração ilustrada pela Figura 79.

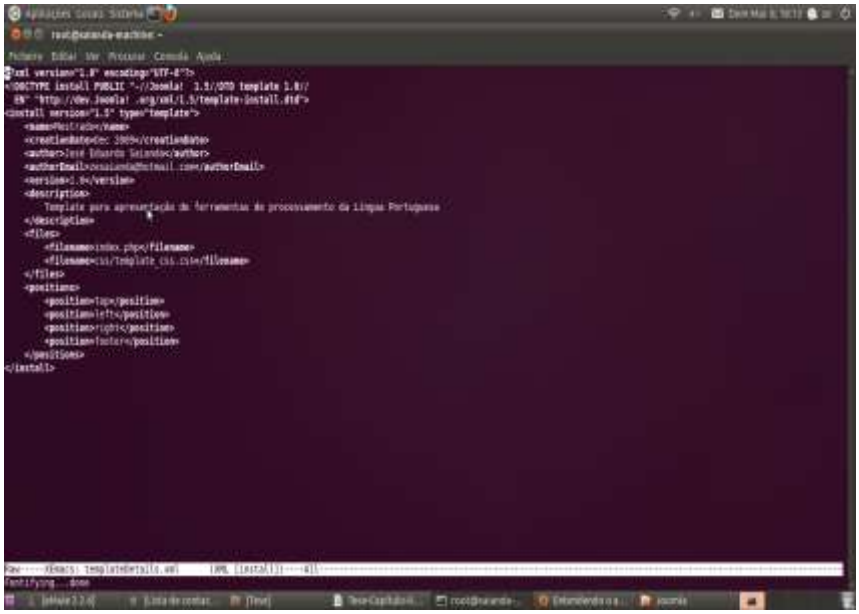


Figura 79 - Ficheiro *templateDetails.xml* responsável pela instalação da *template.*

Como se pode observar através da Figura 79, este ficheiro é constituído por quatro partes:

- <?xml version="1.0" encoding="UTF-8"?> - esta linha declara o cabeçalho do ficheiro xml om a versão do xml e a codificação de idiomas;
- <!DOCTYPE install PUBLIC "-//Joomla! 1.5//DTD template 1.0//

  EN" "http://dev.Joomla! .org/xml/1.5/template-install.dtd"> - aqui é definida a forma como cada navegador deve interpretar o código HTML;

- <install version="1.5" type="template"> - esta linha indica o início do bloco de instalação da template.

```
<name>Mestrado</name>

<creationDate>Dec 2009</creationDate>

<author>José Eduardo Saianda</author>

<authorEmail>zesaianda@hotmail.com</authorEmail>

<version>1.0</version>

<description>

Template para apresentação de ferramentas de processamento da Língua Portuguesa

</description>
```

Este bloco contém a informação que é mostrada no *template manager*.

```
<files>

        <filename>index.php</filename>

        <filename>css/template_css.css</filename>

</files>
```

Este bloco informa o gestor de *templates* acerca de quais os ficheiros que vão para determinados directórios.

```
<positions>

                <position>top</position>

                <position>left</position>

                <position>right</position>

                <position>footer</position>

</positions>
```

Esta secção declara as posições reservadas na template, isto é, identifica posições dentro da template onde serão colocados os outputs dos módulos que lhes estão associados.

- </install> - esta linha encerra o processo de instalação da *template*.

– index.php - ficheiro responsável pela estrutura do *site*.

É neste ficheiro que o CMS Joomla! cria o *site* e estabelece as localizações dos diferentes módulos e componentes nele utilizados.

Este ficheiro contém uma combinação de código PHP e HTML/XHTML.

As Figuras 80 e 81 ilustram a definição deste ficheiro para o caso do nosso *site*.

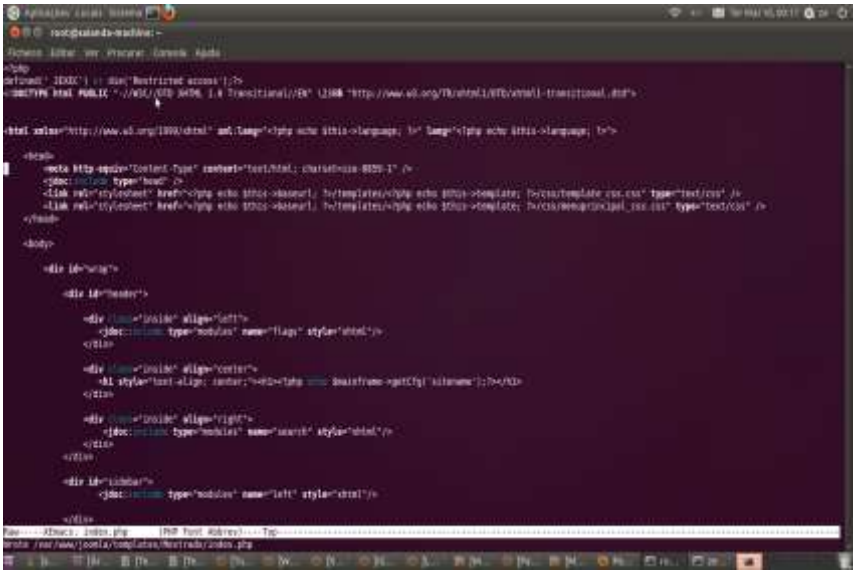


Figura 80 - Definição do ficheiro *index.php*.

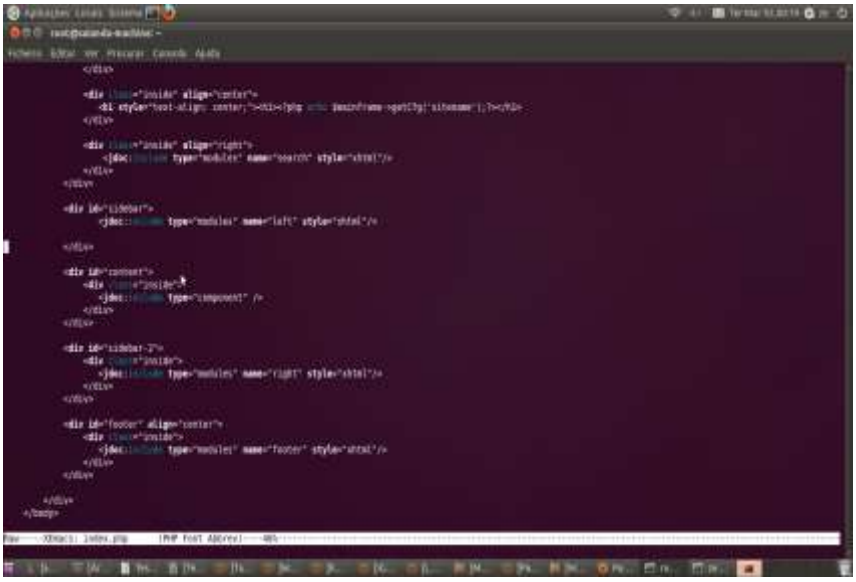


Figura 81 - Definição do ficheiro *index.php* – continuação.

Pela observação das duas Figs. anteriores é possível observar que este ficheiro se encontra dividido em diversas partes:

```
<?php


defined('_JEXEC') or die('Restricted access');?>
```

'_JEXEC' é uma constante que é utilizada para marcar um ponto de entrada seguro no CMS Joomla!.

Se o ficheiro não for carregado normalmente através da página index do CMS Joomla!, nenhuma parte da *template* será mostrada.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" \236�
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Este trecho de código define a forma como cada navegador deve interpretar o código HTML.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this-
>language; ?>" lang="<?php echo $this->language; ?>">
```

Esta trecho de código vai buscar ao ficheiro de configuração do CMS Joomla! - **configuration.php** – o idioma definido por defeito.

```
<head>

    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

    <jdoc:include type="head" />

    <link rel="stylesheet" href="<?php echo $this->baseurl; ?>/templates/<?php
echo $this->template; ?>/css/template_css.css" type="text/css" />

    <link rel="stylesheet" href="<?php echo $this->baseurl; ?>/templates/<?php
echo $this->template; ?>/css/menuprincipal_css.css" type="text/css" />

</head>
```

Este bloco contém referências a folhas de estilos utilizadas na configuração da *template*, assim como a informação acerca do *header* definido na configuração geral do CMS Joomla!.

```
<body>

<div id="wrap">

        <div id="header">

          <div class="inside" align="left">

            <jdoc:include type="modules" name="flags" style="xhtml"/>

          </div>

          <div class="inside" align="center">


            <h1 style="text-align: center;"><h1><?php echo $mainframe->getCfg('sitename');?></h1>

          </div>

          <div class="inside" align="right">

             <jdoc:include type="modules" name="search" style="xhtml"/>

          </div>
        </div>


          <div id="sidebar">
                    <jdoc:include type="modules" name="left" style="xhtml"/>

          </div>

           <div id="content">

                <div class="inside">

                      <jdoc:include type="component" />
                </div>
           </div>


           <div id="sidebar-2">

                    <div class="inside">
                      <jdoc:include type="modules" name="right" style="xhtml"/>

                    </div>
           </div>

                  <div id="footer" align="center">
                    <div class="inside">
                      <jdoc:include type="modules" name="footer" style="xhtml"/>
                    </div>
                  </div>
                </div>
</body>
```

Neste bloco é definida toda a estrutura da *template* que vai ser utilizada no *site*.

– /var/www/joomla/templates/Mestrado/css/menuprincipal_css.css - este ficheiro, cujo código se apresenta no Anexo IX, é responsável pela estilização do menu principal de navegação que se encontra no lado esquerdo;

– /var/www/joomla/templates/Mestrado/css/template_css.css – ficheiro responsável pela estilização da template, cujo código pode ser consultado no Anexo X.

4 Análise da Disponibilidade e Gratuitidade das Ferramentas Existentes no Portal

Após ter sido feito o inventário das ferramentas existentes para o processamento da língua portuguesa procedeu-se ao seu agrupamento daquelas por categorias conforme se mostra na tabela 4.

	Ferramentas	online		download		Paga	
		Sim	Não	Sim	Não	Sim	Não
Ajuda ao ensino	Curso de Português do Brasil		x	x		x	
	Dicionários de Português-Inglês	x			x		x
	Gramática interactiva do Português	x			x	x	
	Minigramática	x			x		x
Ajuda à redacção	Aspell		x	x			x
	CoGrOO		x	x			x
	Correctores ortográficos		x	x			x
	FliP	x			x	x	
	Gramática electrónica	x			x	x	
	Ispell		x	x			x
	Jspell		x	x			x
	Lince	x			x	x	
	Redacção Língua Portuguesa	x			x	x	
	ReGra	x			x		x
	WebJspell	x			x		x
Alinhadores	Alinhador online CEPRIL	x			x		x
	MTTK		x	x			x
	NATools		x	x			x
	VisualLIHLA	x			x		x
	VisualTCA	x			x		x
Analisadores	Concordanciador de um milhão de palavras	x			x		x
	Curupira	x			x		x
	GojolParser	x			x	x	
	Lexificador DeepDict	x			x		x
	Lingua::PT::PLNBase	x		x			x
	LX-Suite	x			x		x
	PoS FreeLing	x		x			x
	Pos Tree-Tagger	x		x			x
	PtStemmer	x		x			x
	Rembrandt	x			x		x
Conjugadores verbais	Conjugue	x		x			x
	Gconjugue	x		x			x
	LX-Conj	x			x		x

	Ferramentas	online		download		Paga	
		Sim	Não	Sim	Não	Sim	Não
Extractores de N-Gramas	NSP	x		x			x
	SENTA	x			x		x
Ferramentas especializadas	DepPattern	x		x			x
	DiZer 2.0	x			x		x
	EELO	x			x		x
	e-Termos	x			x		x
	Etiquet(H)AREM		x	x			x
	HPC		x	x			x
	Indexador estatístico		x	x			x
	Lácio-Web		x	x			x
	Lingua Toolkit	x		x			x
	Multilingual Dependency Parser	x		x			x
	Multilingual Term Extractor	x		x			x
	Navegador MultiWordnet	x			x		x
	NILC's Taggers	x		x			x
	O Constructor	x			x		x
	SciPo	x			x		x
	SciPo-Farmácia	x			x		x
	Sílabas-PT		x	x			x
	Smell	x			x		x
	TeP 2.0 Beta		x	x			x
	Textcat		x	x			x
	TextQuim	x			x		x
	Unitex 2.0		x	x			x
	UNL	x			x		x
	WordNetBr	x			x		x
Processamento de fala	Dixi	x			x		x
	Info-Maker	x			x		x
	Lingua-PT-Speaker		x	x			x
	Páginas Falantes	x			x		x
	SVITD	x			x		x
	Tele-Balcão	x			x		x
	Voice Mail	x			x		x
	Web Wake Up	x			x		x
Sumarizadores	Explosa	x			x		x
	GistSumm		x	x			x

	Ferramentas	online		download		Paga	
		Sim	Não	Sim	Não	Sim	Não
Tradução automática	Apertium machine translation machine and tools		x	x			x
	EPT-Web	x			x		x
	Galician-Portuguese translator	x		x			x
	PUL0	x			x		x
TOTAL	**73**	**54**	**19**	**32**	**41**	**7**	**66**

Tabela 4 - Lista de ferramentas existentes no portal agrupadas por categorias.

A partir da tabela anterior foram construídas duas outras que permitem efectuar a comparação – em percentagem – entre as várias categorias de ferramentas relativamente a várias características, nomeadamente as que a seguir apresentamos:

- % de ferramentas online;
- % de ferramentas para download;
- % de ferramentas online/download;
- % de ferramentas pagas;
- % de ferramentas gratuitas.

Categoria	FERRAMENTAS ONLINE			TOTAL
	PERCENTAGEM			
	Uso exclusivo	Download	Uso exclusivo / Download	
Ajuda ao ensino	75	25	0	100
Ajuda à redacção	55	45	0	100
Alinhadores	60	40	0	100
Analisadores	60	0	40	100
Conjugadores verbais	33	0	67	100
Extractores de N-Gramas	50	0	50	100
Ferramentas especializadas	46	33	21	100
Processamento de Fala	87,5	12,5	0	100
Sumarizadores	50	50	0	100
Tradução automática	50	25	25	100

Tabela 5 - Análise da disponibilidade das ferramentas existentes por categoria.

Categoria	FERRAMENTAS ONLINE		TOTAL
	PERCENTAGEM		
	Gratuitas	Pagas	
Ajuda ao ensino	50	50	100
Ajuda à redacção	64	36	100
Alinhadores	100	0	100
Analisadores	90	10	100
Conjugadores verbais	100	0	100
Extractores de N-Gramas	100	0	100
Ferramentas especializadas	100	0	100
Processamento de Fala	100	0	100
Sumarizadores	100	0	100
Tradução automática	100	0	100

Tabela 6 - Análise da gratuitidade das ferramentas existentes por categoria.

4.1 Análise Global das Diversas Categorias de Ferramentas Quanto à sua Disponibilidade

A tabela 6 permite-nos concluir que a categoria que apresenta maior percentagem de ferramentas disponíveis exclusivamente *online* é a de Sumarizadores seguida pela de Processamento de Fala, factos que também são apresentados no gráfico 1.

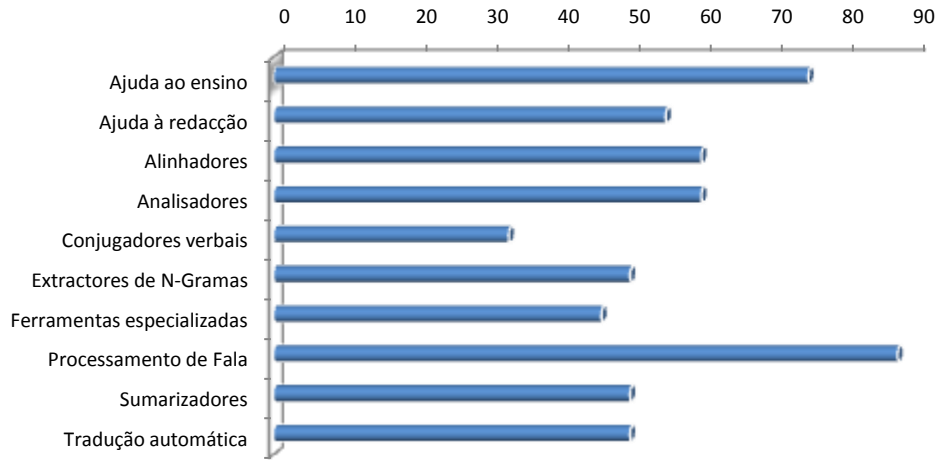


Gráfico 1 - Comparação das percentagens de ferramentas disponíveis exclusivamente *online* nas diversas categorias

A mesma tabela também nos permite deduzir que a maior percentagem de ferramentas disponíveis exclusivamente para download é a de Ajuda à Redacção seguida pela de Alinhadores. Os Analisadores, os Conjugadores Verbais, os Extractores de N-Gramas e os Sumarizadores apresentam percentagem nula.

Estes factos ficam explicitados no gráfico 2.

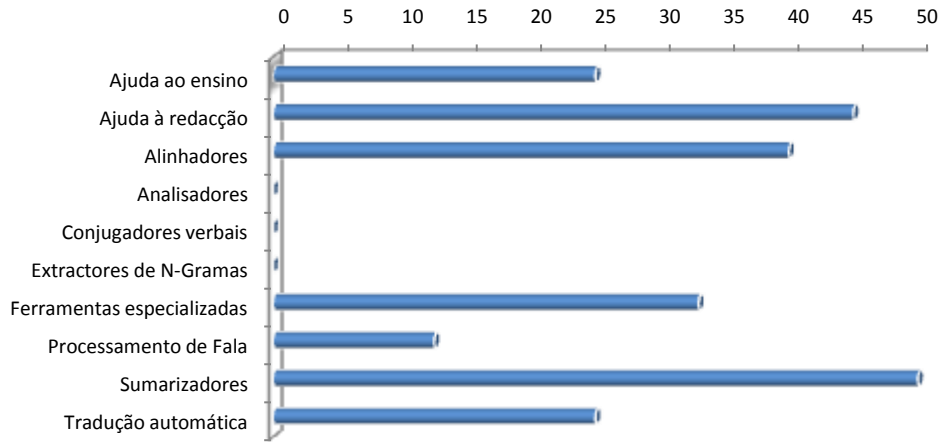


Gráfico 2 - Comparação das percentagens de ferramentas disponíveis exclusivamente para *download* nas diversas categorias

Verifica-se ainda - pela mesma tabela - que apenas 50% das categorias apresenta percentagem não nula de ferramentas disponíveis simultaneamente *online* e para *download*, sendo que a que apresenta percentagem mais elevada é a de Conjugadores Verbais seguida pela de Extractores de N-Gramas, factos ilustrados também no gráfico 3;

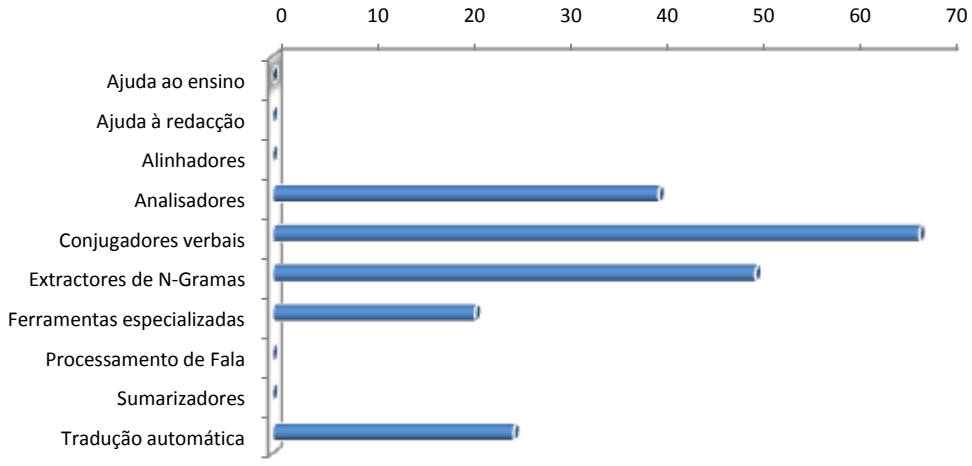


Gráfico 3 - Comparação das percentagens de ferramentas disponíveis tanto *online* como para *download* nas diversas categorias.

É-nos lícito ainda afirmar que apenas 30% das categorias apresenta ferramentas pagas. Constatamos que as categorias 'Alinhadores', 'Conjugadores Verbais', 'Extractores de N-Gramas', 'Ferramentas Especializadas', 'Processamento de Fala', 'Sumarizadores' e 'Tradução Automática' contêm apenas ferramentas gratuitas;

O gráfico 4 ilustra estes resultados.

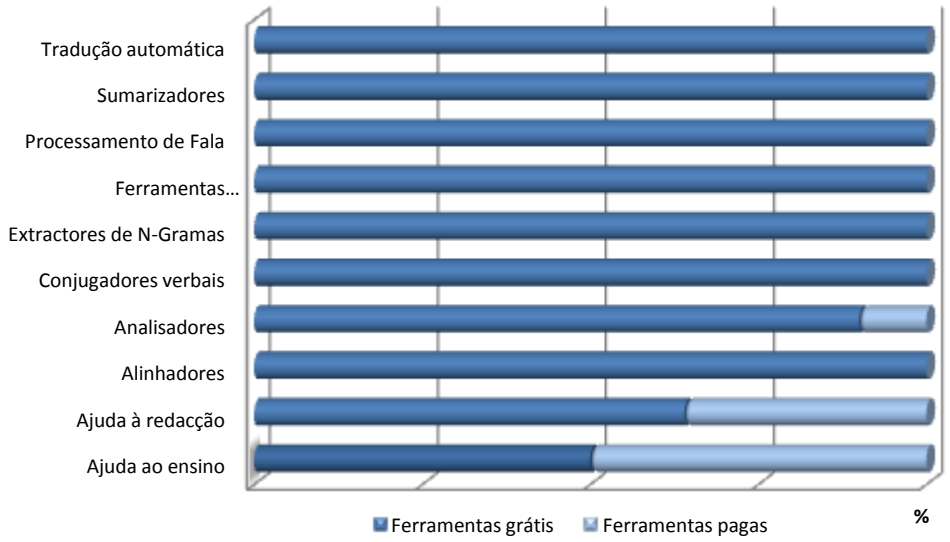


Gráfico 4 - Comparação entre o número de ferramentas pagas e o número de ferramentas gratuitas por categoria.

4.2 Análise de Ferramentas por Categoria

4.2.1 Ajuda ao ensino

A partir da tabela 6 podemos concluir que:

- a maioria das ferramentas existentes nesta secção apenas se encontra disponível exclusivamente *online* – gráfico 5;

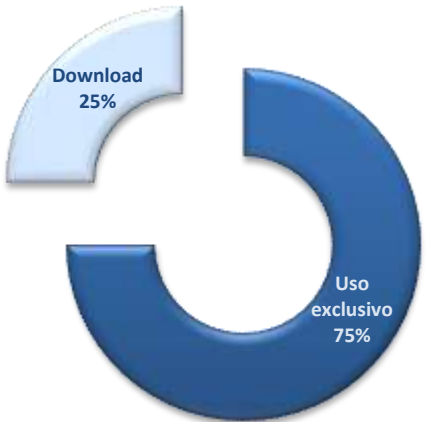


Gráfico 5 - Comparação das diversas formas de disponibilidade de ferramentas na categoria de Ajuda ao ensino.

- não existem ferramentas que estejam simultaneamente *online* e para *download* – gráfico 5;

- a percentagem de ferramentas pagas é idêntica à percentagem de ferramentas gratuitas – gráfico 6.

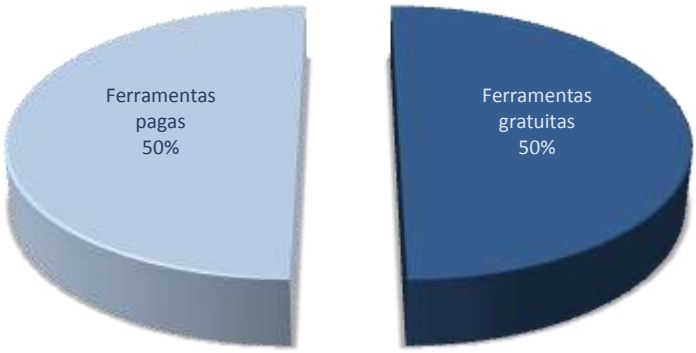


Gráfico 6 - Comparação entre a percentagem de ferramentas pagas e a percentagem de ferramentas gratuitas na categoria de Ajuda ao ensino.

4.2.2 Ajuda à redacção

No que diz respeito a esta categoria podemos constatar que:

- a percentagem de ferramentas disponíveis exclusivamente online é superior à de ferramentas exclusivamente disponíveis para download – gráfico 7;

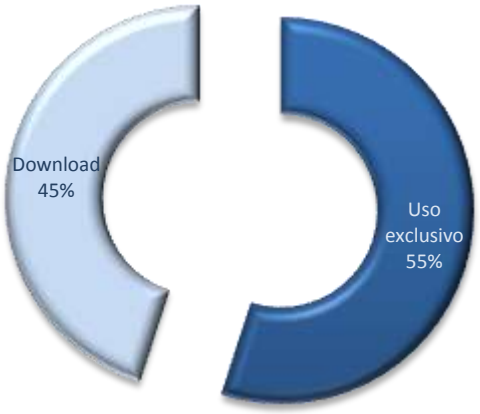


Gráfico 7 - Comparação das diversas formas de disponibilidade de ferramentas na categoria de Ajuda ao à redacção.

- não existem nesta categoria ferramentas disponíveis simultaneamente *online* e para *download*;
- a percentagem de ferramentas gratuitas é praticamente o dobro da percentagem de ferramentas pagas – gráfico 8.

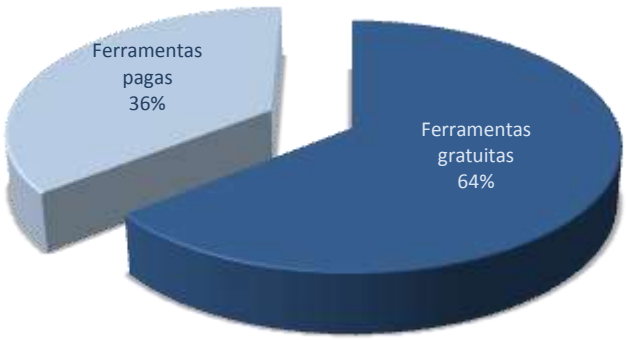


Gráfico 8 - Comparação entre a percentagem de ferramentas pagas e a percentagem de ferramentas gratuitas na categoria de Ajuda à redacção.

4.2.3 Alinhadores

No que se refere à categoria de Alinhadores verificamos que:

- mais de metade das ferramentas existentes nesta categoria se encontram exclusivamente disponíveis *online* e não existem ferramentas disponíveis *online* e para *download* – gráfico 9;

Gráfico 9 - Comparação das diversas formas de disponibilidade de ferramentas na categoria de Alinhadores.

- ainda relativamente a esta categoria podemos chegar à conclusão de que apenas existem ferramentas gratuitas.

4.2.4 Analisadores

Relativamente a esta categoria concluímos que mais de 50% das ferramentas apenas se encontram disponiveis *online* – gráfico 10 – e que não existem ferramentas disponíveis exclusivamente para *download*;

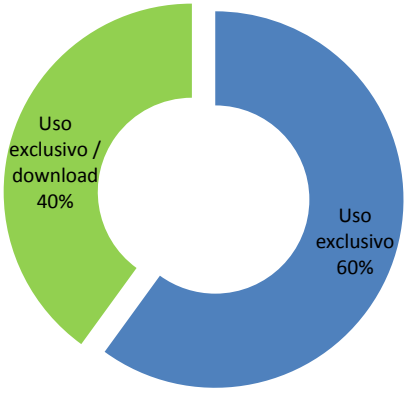


Gráfico 10 - Comparação das diversas formas de disponibilidade de ferramentas na categoria de Analisadores.

Verificamos ainda que apenas uma percentagem muito reduzida das ferramentas são pagas - gráfico 11.

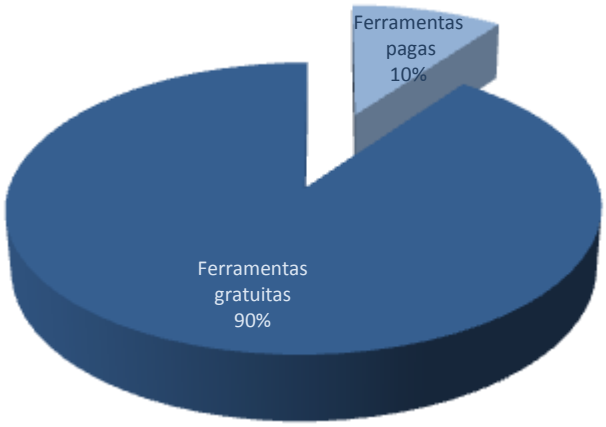


Gráfico 11 - Comparação entre a percentagem de ferramentas pagas e a percentagem de ferramentas gratuitas na categoria de Analisadores

4.2.5 Conjugadores Verbais

No que se refere a esta categoria podemos concluir que a maioria das ferramentas existentes se encontra disponível tanto *online* como para *download* – gráfico 12.

O mesmo gráfico deixa-nos perceber que esta categoria não possui ferramentas disponíveis exclusivamente para *download*.

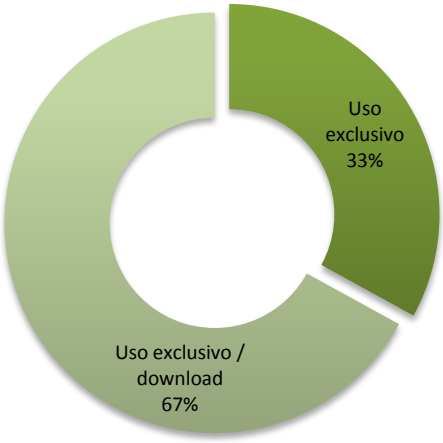


Gráfico 12 - Comparação das diversas formas de disponibilidade de ferramentas na categoria de Conjugadores Verbais.

Tal como na categoria anterior, todas as ferramentas de conjugadores verbais são gratuitas.

4.2.6 Extractores de N-Gramas

No que se refere à categoria de Extractores de N-Gramas podemos concluir que 50% das ferramentas existentes nesta categoria encontram-se disponíveis simultaneamente online e para download, não existindo ferramentas disponíveis apenas para *download* – gráfico 13.

Gráfico 13 - Comparação das diversas formas de disponibilidade de ferramentas na categoria de Extractores de N-Gramas.

No que diz respeito à gratuitidade das ferramentas existentes, podemos concluir que todas elas são gratuitas.

4.2.7 Ferramentas especializadas

No que diz respeito a esta categoria podemos concluir que quase 50% das ferramentas existentes se encontram disponíveis apenas online e que apenas uma percentagem muito pequena se encontra simultaneamente disponível online e para download – gráfico 14;

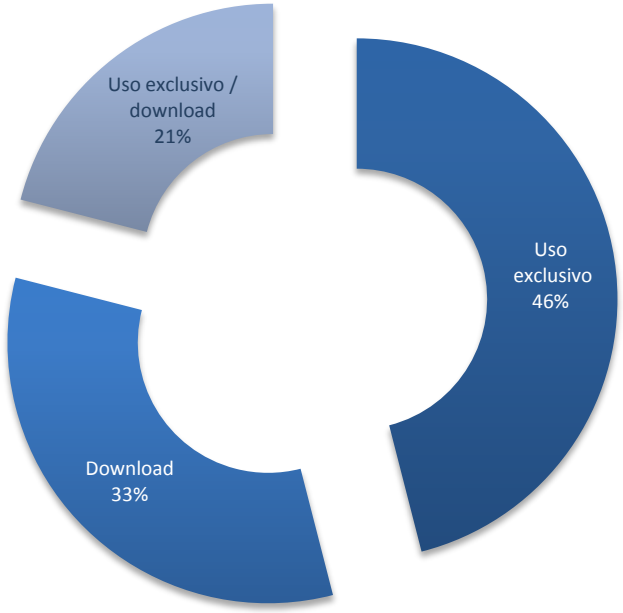


Gráfico 14 - Comparação das diversas formas de disponibilidade de ferramentas na categoria de Extractores de N-Gramas.

Quanto à gratuitidade de ferramentas disponíveis nesta categoria, podemos concluir que todas as ferramentas são gratuitas.

4.2.8 Processamento de fala

Em relação à categoria de 'Processamento de Fala' podemos verificar que a grande maioria das ferramentas existentes apenas se encontra disponível *online*, que apenas uma percentagem mínima (13%) se encontra disponível para *download* e que não existem ferramentas disponíveis simultaneamente *online* e para *download*, factos que ficam explicitados no gráfico 15.

Gráfico 15 - Comparação das diversas formas de disponibilidade de ferramentas na categoria de Processamento de Fala.

Quanto à gratuitidade das ferramentas existentes nesta categoria, verificamos que atinge os 100%.

4.2.9 Sumarizadores

No que diz respeito a esta categoria podemos concluir que apenas existem ferramentas exclusivamente disponíveis online - gráfico 16 - e que todas são gratuitas.

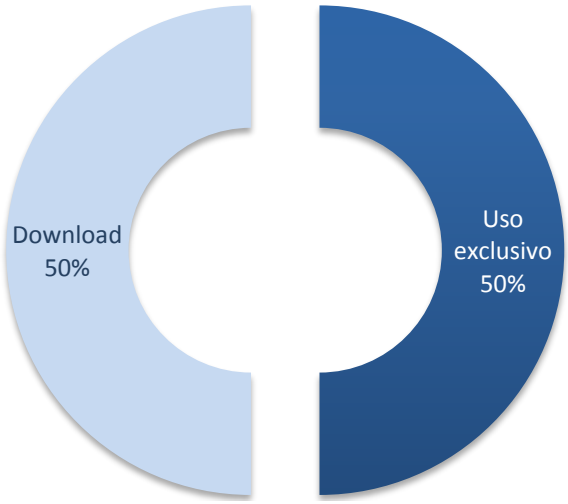


Gráfico 16 - Comparação das diversas formas de disponibilidade de ferramentas na categoria de Sumarizadores.

4.2.10 Tradução automática

A grande maioria das ferramentas existentes para este efeito encontra-se disponível *online*, sendo que 50% está disponível tanto para *download* como para *download* e *online* simultaneamente, facto que o gráfico 17 atesta.

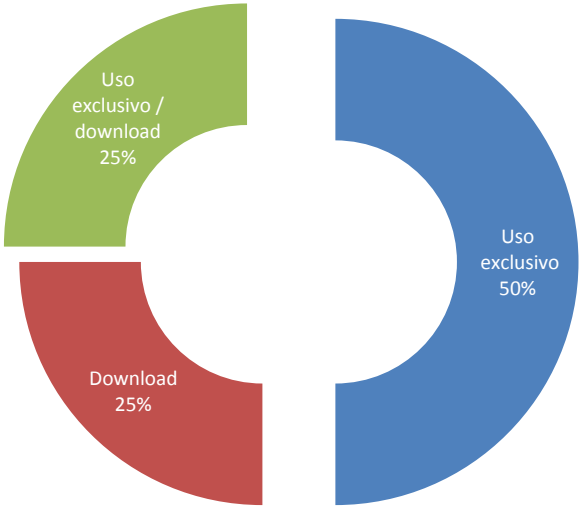


Gráfico 17 - Comparação das diversas formas de disponibilidade de ferramentas na categoria de Tradução Automática

À semelhança das cinco categorias anteriores também esta apresenta apenas ferramentas gratuitas.

5 CONCLUSÃO E TRABALHO FUTURO

5.1 CONCLUSÃO

Terminar um trabalho tem sempre duas vertentes.

Uma é a sensação do dever cumprido, que, naturalmente, proporciona satisfação a quem, com ele, percorreu um longo caminho… O caminho que se esboçou a partir de uma ideia e que o tempo permitiu se concretizasse pouco a pouco. Um caminho difícil. Um caminho que percorremos acompanhado pela solidão de muitas noites sem dormir!

Um caminho que agora, chegado ao fim, nos deixa o amargo sabor da perda: a companhia desfez-se… A preocupação constante de uma tarefa por e para acabar termina aqui.

Mas… um caminho que, apesar de tudo, nos proporcionou muito prazer. O prazer da aprendizagem e da descoberta!

Olhando para trás relembramos o que foi necessário fazer para levar a tarefa a bom porto! E assim queremos deixar aqui o testemunho do muito que aprendemos ao realizar – passo a passo – este trabalho.

Progredimos.

Adquirimos conhecimentos nas áreas de sistemas de gestão de conteúdos (CMS), uma vez que para a implementação do site descrito no capítulo III nos levou a estudar o funcionamento do CMS Joomla!.

Progredimos também na investigação do funcionamento e do processamento de língua natural (PLN), pois a colocação de determinadas ferramentas a disponibilizar online no *site*, assim o exigiu. E até adquirimos conhecimentos provenientes da área dos Estudos Linguísticos…

Descobrimos ferramentas de cuja existência nem sequer suspeitávamos.

Reforçámos a ideia - que já nos vinha de trás - de que a Informática, para além de abrir uma grande janela sobre o mundo do conhecimento, favorece a investigação de outras ciências ao permitir, e mesmo sugerir, avanços que, sem ela, seriam difíceis. É o caso das Ciências da Linguagem que hoje, com a cooperação de ferramentas informáticas, podem estudar *corpora* de milhões de palavras e constituir Bases de Dados terminológicos, imprescindíveis aos tradutores e aos estudiosos de Traductologia. E ainda - last but not least - enriquecer os próprios estudos de Terminologia.

Estudos estes que não só permitem avanços na investigação sobre o discurso (e os vários tipos de discurso), mas que também encontram aplicações práticas, nomeadamente na

constituição de glossários respeitantes a domínios específicos. Os profissionais de Línguas Aplicadas conhecem bem a sua utilidade tanto no domínio da formação como no exercício da profissão.

Terminamos, consciente de, com a pesquisa efectuada, ter contribuído, para tornar mais fácil a investigação de quem procura encontrar na Internet as ferramentas de que necessita para o tipo de trabalho em causa.

Com a consciência também de ter elaborado um *site* que nos proporcionou não só a aquisição de muitos conhecimentos, mas que, também, de algum modo, pode dar um contributo válido para o próprio *site* do Departamento de Informática. Com efeito o apuramento das principais ferramentas existentes no campo do processamento de língua natural, bem como a sua disponibilidade para utilização da comunidade, pode levar a que este grande *site* adquira ainda maior "visibilidade".

Terminamos, finalmente, com a consciência de que muito há ainda a fazer…

5.2 TRABALHO FUTURO

O melhoramento estético do site, uma vez que não desconhecemos a importância do seu aspecto geral para o êxito que ambicionamos, é um dos nossos objectivos.

Tal como o é também a introdução de alterações que possibilitem o acesso a partir de plataformas móveis – telemóveis, PDA's e tablets. Isso permitirá pensar igualmente em adaptações para "ajuda ao ensino" com a inserção de dicionários

- de Língua Portuguesa (e eventualmente outras)
- de sinónimos
- de fraseologia
- de citações
- de conjugação
- de história.

E, finalmente, a criação de mecanismos que possibilitem o acesso de invisuais ao site está também nos nossos horizontes.

BIBLIOGRAFIA

LIVROS

CARVALHO, P. & OLIVEIRA, H., Manual de Utilização do Etiquet(H)AREM. 29 de Abril de 2008.[13-072011]. Disponível em WWW:<URL:http://www.linguateca.pt/aval_conjunta/HAREM/ManualUtilEtiquetHAREM.pdf>

NORTH, B., Joomla! A User's Guide – Building a Successful Joomla! Powered Website. Prentice Hall,2008. [25-03-2011]. Disponível em WWW:<URL: http://www.prenhallprofessional.com/safarienabled> por 45 dias. *ISBN 0-13-613560-9*

SCHMID, H., TreeTagger - a language independent part-of-speech tagger, Março de 2011. Disponível em WWW: <URL:http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>

ARTIGOS CONSULTADOS ONLINE

http://www.apl.org.pt/docs/actas-10-encontro-apl-1994.pdf, ALMEIDA, J. e PINTO, U., «Jspell – um módulo para análise léxica genérica de linguagem natural», *Actas do X Encontro da Associação Portuguesa de Linguística*, Évora, 1995

(última consulta em 25-01-2011)

*aspell.net/0.50-doc/man-html/index.html*Em cache Semelhante, ATKINSON, K., "GNU Aspell 0.50.5.", 10 de Fevereiro de 2004

(última consulta em 26-01-2011)

*cpansearch.perl.org/src/...NSP.../cicling2003.pdf*Semelhante, BANERJEE, S. & PEDERSEN, T., "The Design, Implementation and Use of the Ngram Statistics Package", Abril de 2011

(última consulta em 26-01-2011)

mi.eng.cam.ac.uk/~wjb31/.../hltemnlp05wtop.pd, DENG,Y. & BYRNE, W., "HMM word and phrase alignment for statistical machine translation#, Março de 2008

(última consulta em 02-02-2011)

mi.eng.cam.ac.uk/~wjb31/.../hlt06mttkeabs.pdf, "MTTK: An alignment toolkit for statistical machine translation", Junho de 2006

(última consulta em 02-02-2011)

http://gramatica.usc.es/~gamallo/, GAMALLO, P., Universidade de Santiago de Compostela,

(última consulta em 02-02-2011)

http://scholar.google.pt/scholar?q=Automatic+Thesaurus+(extracted+from+parsed+corpora),&hl=pt-PT&as_sdt=0&as_vis=1&oi=scholart, "Automatic Thesaurus (extracted from parsed *corpora*)", Maio de 2008

(última consulta em 12-03-2011)

gramatica.usc.es/~gamallo/gale.../index2.1.htm, "Extractor multilingüe de términos multipalabra" (versión 2.1), Maio de 2011

(última consulta em 12-04-2011)

<http://gramatica.usc.es/~gamallo/gale-extra/index2.1.htm>,"TRANSLITERADOR/ TRADUTOR, Julho de 2011,

(última consulta em 03-05-2011)

*gramatica.usc.es/pln/tools/user_guide.pdf*Semelhante, GONZÁLEZ, I., Dep Pattern User Manual, Dezembro de 2008

(última consulta em 03-05-2011)

*igm.univ-mlv.fr/~**unitex/UnitexManual2.0**.pdf,* PAUMIER, S., "UNITEX 2.0 USER MANUAL", Outubro de 2008

(última consulta em 03-05-2011)

Universidade de Santiago de Compostela - Grupo ProLNat, "FreeLing User Manual 2.2", Setembro de 2010

(última consulta em 23-05-2011)

SITES CONSULTADOS

http://code.google.com/p/ptstemmer/

(última consulta em 23-03-2011)

http://code.google.com/p/silabaspt/

(última consulta em 15-03-2011)

http://cogroo.sourceforge.net/

(última consulta em 08-01-2011)

http://gconjugue.codigolivre.org.br/

(última consulta em 18-07-2011)

http://gramtrans.com/deepdict/

(última consulta em 15-02-2011)

http://label.ist.utl.pt/pt/apresentacao.php

(última consulta em 17-05-2011)

http://label.ist.utl.pt/pt/eelo_intr_pt.php

(última consulta em 18-05-2011)

http://linguateca.di.uminho.pt/natools/htdocs/tools.html

(última consulta em 25-01-2011)

http://lxcenter.di.fc.ul.pt/services/en/LXServicesSuite.html

(última consulta em 24-03-2011)

http://lxcenter.di.fc.ul.pt/services/pt/LXServicesWordnetPT.html

(última consulta em 20-05-2011)

http://maracujah.net/

(última consulta em 21-03-2011)

http://natura.di.uminho.pt/webjspell/jsol.pl

(última consulta em 25-01-2011)

http://search.cpan.org/~ambs/Lingua-PT-PLNbase-0.24/lib/Lingua/PT/PLNbase.pm

(última consulta em 07-07-2011)

http://search.cpan.org/~ambs/Lingua-PT-Speaker/pt-speak.in

(última consulta em 06-07-2011)

http://search.cpan.org/~fernandes/Text-Statistics-Latin-0.06/lib/Text/Statistics/Latin.pm

(última consulta em 08-01-2011)

http://senta.di.ubi.pt/

(última consulta em 10-04-2011)

http://visl.hum.sdu.dk/itwebsite/port/portgram.html

(última consulta em 21-01-2011)

http://www.brasiliano.it/

(última consulta em 18-07-2011)

http://www.ectaco.com/dictionaries/portuguese.asp

(última consulta em 07-01-2011)

http://www.etermos.cnptia.embrapa.br/

(última consulta em 17-04-2011)

http://www.flip.pt/

(última consulta em 20-01-2011)

http://www.icmc.usp.br/~taspardo/Projects.htm

(última consulta em 03-07-2011)

http://www.ime.usp.br/~ueda/br.ispell/

(última consulta em 07-01-2011)

http://www.ime.usp.br/~ueda/br.ispell/conjugue.html

(última consulta em 20-03-2011)

http://www.it.uc.pt/signal/sait/Despertar.htm
(última consulta em 14-07-2011)

http://www.it.uc.pt/signal/sait/falantes.htm
(última consulta em 13-07-2011)

http://www.it.uc.pt/signal/sait/InfoMaker.htm
(última consulta em 23-06-2011)

http://www.it.uc.pt/signal/sait/telebal.htm
última consulta em 15-07-2011)

http://www.it.uc.pt/signal/sait/voicemail.htm
(última consulta em 16-07-2011)

http://www.leonel.profusehost.net/indexc.htm
(última consulta em 04-06-2011)

http://www.let.rug.nl/vannoord/TextCat/
(última consulta em 15-06-2011)

http://www.lexikon.com.br/gramat/gramat.htm
(última consulta em 20-01-2011)

http://www.linguateca.pt/Repositorio/GojolParser.txt
(última consulta em 07-02-2011)

http://www.nilc.icmc.usp.br/%7Earianidf/WordNet-BR.html
(última consulta em 10-06-2011)

http://www.nilc.icmc.usp.br/~scipo/
(última consulta em 19-07-2011)

http://www.nilc.icmc.usp.br/scipo-farmacia/
(última consulta em 04-07-2011)

http://www.nilc.icmc.usp.br/dizer2/
(última consulta em 11-04-2011)

http://www.nilc.icmc.usp.br/lacioweb/index.htm

(última consulta em 18-06-2011)

http://www.nilc.icmc.usp.br/minigramatica/mini/sejabemvindo.htm

(última consulta em 24-01-2011)

http://www.nilc.icmc.usp.br/nilc/projects/ept-web.htm

(última consulta em 25-07-2011)

http://www.nilc.icmc.usp.br/nilc/projects/hpc/>

(última consulta em 18-07-2011)

http://www.nilc.icmc.usp.br/nilc/projects/LIBRAS2.htm

(última consulta em 26-07-2011)

http://www.nilc.icmc.usp.br/nilc/projects/regra.htm

(última consulta em 21-01-2011)

http://www.nilc.icmc.usp.br/nilc/projects/scipo.htm

(última consulta em 19-05-2011)

http://www.nilc.icmc.usp.br/nilc/projects/unl.htm

(última consulta em 18-06-2011)

http://www.nilc.icmc.usp.br/nilc/projects/visuallihla.htm

(última consulta em 18-02-2011)

http://www.nilc.icmc.usp.br/nilc/toolms/curupira.html

(última consulta em 07-02-2011)

http://www.nilc.icmc.usp.br/nilc/tools/nilctaggers.html

(última consulta em 04-06-2011)

http://www.nilc.icmc.usp.br/nilc/tools/pagina-visualtca/visualtca.htm

(última consulta em 19-02-2011)

http://www.nilc.icmc.usp.br/~scipo-farmacia/

(última consulta em 19-07-2011)

http://www.nilc.icmc.usp.br/~scipo-farmacia/

(última consulta em 19-07-2011)

http://www.speech.inesc.pt/~lco/dixi/dixi_pt.cgi

(última consulta em 18-07-2011)

http://www.speech.inesc.pt/~lco/svit/svitd_pt.cgi

(última consulta em 10-07-2011)

http://www2.dc.ufscar.br/~lucia/PROJECTS/EXPLOSA.htm

(última consulta em 04-07-2011)

http://www2.lael.pucsp.br/corpora/alinhador/

(última consulta em 04-07-2011)

http://www2.lael.pucsp.br/corpora/bp/conc/

(última consulta em 05-07-2011)

http://www6.ufrgs.br/textquim/

(última consulta em 10-01-2011)

http://xixona.dlsi.ua.es/apertium-www/

(última consulta em 08-07-2011)

http://xldb.di.fc.ul.pt/Rembrandt/

(última consulta em 10-07-2011)