A construção de recursos lingüístico-computacionais para o português do Brasil: o projeto Unitex-PB

Marcelo Caetano Martins Muniz

A construção de recursos lingüístico-computacionais para o português do Brasil: o projeto Unitex-PB

Marcelo Caetano Martins Muniz

Profª Drª Maria das Graças Volpe Nunes

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP - São Carlos
Fevereiro/2004

Dedicatória

Aos meus pais, João Acássio e Célia.

Agradecimentos

Á minha família, pela dedicação e apoio incondicional em todos momentos da minha vida.

Á Deus, por me dar forças para chegar até aqui.

Aos meus grandes amigos Rodrigo e Dalton, pelos grandes momentos de descontração e por me agüentarem nos momentos de mau humor.

À Graça, por sua orientação e voto de confiança.

Aos amigos do NILC, que sempre me ajudaram com prontidão, em especial ao Ivair pela grande contribuição a esse trabalho.

Ao pessoal do Intermídia, meu segundo laboratório.

Á Capes, pelo apoio financeiro.

A todos amigos e pessoas que me ajudaram de uma forma ou de outra neste dois anos em São Carlos.

Muito obrigado!

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

A escassez de recursos lingüístico-computacionais é um dos maiores entraves para o avanço das pesquisas, e conseqüente desenvolvimento de sistemas, na área de Processamento de Língua Natural (PLN) no Brasil. Este trabalho documenta a construção de uma série recursos lingüístico-computacionais para português brasileiro seguindo os formalismos utilizados pela ferramenta de processamento de córpus Unitex. Foram construídos léxicos computacionais, regras de resolução de ambigüidades e bibliotecas para acesso a léxicos compactados, assim como algumas ferramentas para validar esses recursos. Os desafios encontrados durante todo o processo são discutidos nessa dissertação.

Abstract

The lack of computational linguistic resources represents one of the major challenges to the development and research activities related to Natural Language Processing. This work documents the project and development of various computational linguistic resources that support the Brazilian Portuguese language according to the formal methodology used by the corpus processing system called Unitex. The delivered resources include computational lexicons, rules to solve ambiguity, libraries to access compressed lexicons, and additional tools to validate those resources. Some aspects about the main challenges encountered during the course of this project are also addressed.

CAPÍTULO 1

Introdução

Um dos maiores entraves para o avanço das pesquisas, e conseqüente desenvolvimento de sistemas, na área de Processamento de Língua Natural (PLN) no Brasil é a escassez de recursos lingüístico-computacionais que, em última análise, fornecem todo o conhecimento do domínio necessário nessa área. Por serem muito especializados, volumosos e complexos, sua construção exige equipes interdisciplinares treinadas, cujo custo de manutenção tem impedido que as pesquisas em PLN/português cheguem a patamares compatíveis com os da língua inglesa. A importância desses recursos para o desenvolvimento dessa área é visível nos inúmeros trabalhos para a língua inglesa que compartilham estudos, córpus e sistemas dependentes de língua.

Os recursos necessários para o desenvolvimento de aplicações robustas e abrangentes podem ser divididos em dois grupos: aqueles que oferecem conhecimento lingüístico, mas não o processam automaticamente, como os Dicionários Eletrônicos (*Machine-Tractable Dictionaries- MTDs*), Córpus (bancos de textos autênticos) e *Thesaurus*; e aqueles que se caracterizam por processar a língua para efeito de algum resultado pré-definido, como os *Taggers* (etiquetadores morfossintáticos) ou os analisadores morfológicos, sintáticos (Parsers) e semânticos. Pode-se afirmar que a maior parte do tempo e esforço necessários para o desenvolvimento de uma aplicação de PLN é dedicada à construção dos recursos lingüísticos que dão suporte ao funcionamento da mesma.

Nos últimos anos, têm se notado um grande esforço dos pesquisadores da área de PLN para a padronização na construção desses recursos, visando principalmente a reusabilidade. Alguns padrões e ferramentas têm se desta-

cado no cenário internacional e vêm sendo utilizados por vários grupos de pesquisas em muitos países. Um exemplo de padrão foi o desenvolvido no LADL (*Laboratoire d'Automatique Documentaire et Linguistique*) na França, o DELA (*Dictionnarie Electronique du LADL*), juntamente com a ferramenta de análise de córpus INTEX[1] (Silberztein 2000a).

O DELA tornou-se um padrão de dicionários eletrônicos (também conhecidos como léxicos computacionais) que é utilizado pela rede de pesquisa européia Relex[2]. Esses dicionários foram primeiramente desenvolvidos para serem utilizados pela ferramenta de análise de córpus, o INTEX, e mais atualmente para sua versão de código fonte aberto, o Unitex[3].

O Unitex é um ambiente de desenvolvimento lingüístico que inclui dicionários e gramáticas de grande cobertura de várias línguas (Espanhol, Inglês, Francês, Grego, Russo, Português de Portugal, Tailandês), e processa textos com milhões de palavras em tempo real. Ele inclui ferramentas para criar e manter recursos lexicais, criar e manter gramáticas de remoção de ambigüidades, e esses dicionários e gramáticas podem ser aplicados aos textos para localizar padrões morfológicos, lexicais e sintáticos, remover ambigüidades, e etiquetar palavras simples e compostas.

O *Núcleo Interinstitucional de Lingüística Computacional* (NILC)[4] tem desenvolvido, desde sua criação em 1991, vários aplicativos e recursos lingüísticos para o português brasileiro (PB). Entre os aplicativos desenvolvidos, destaca-se o Revisor Gramatical ReGra (Martins et al. 1998), desenvolvido com o apoio da Itautec-Philco, da Fapesp, do CNPq e da Finep, que está comercialmente disponível no produto Microsoft Office versão português, desde 2000.

Este revisor conta com um léxico de aproximadamente 500 mil entradas de palavras simpes (incluindo derivações), cada uma podendo pertencer a uma ou mais categorias sintáticas, com atributos específicos e distintos.

Os dados do léxico, mais as unidades de verbos flexionados de ênclise e mesóclise, totalizam mais de 1.500.000 entradas (Nunes et al. 1996) e fazem parte ainda de uma base de dados lexicais, Diadorim, disponível na Web para consulta[5]. Essa base de dados centraliza todas as informações lexicais do NILC, resultado de cerca de 10 anos de pesquisas. Porém, o acesso de aplicações diretamente a essa base de dados é extremamente lento, uma vez que esses dados estão em um banco de dados relacional e suas tabelas possuem muitos milhares de entradas (Greghi 2002).

Uma saída para esse problema é a utilização de métodos de compactação e

[1]Veja `http://www.nyu.edu/pages/linguistics/intex/`
[2]Veja `http://ladl.univ-mlv.fr/Relex/introduction.html`
[3]Veja `http://www-igm.univ-mlv.fr/~unitex/`
[4]Veja `http://www.nilc.icmc.usp.br/`
[5]Disponível para consulta em `http://www.nilc.icmc.usp.br/`, Tool & Resources, DIADORIM

manipulação de léxicos baseados em autômatos finitos. O Revisor Gramatical ReGra utiliza, no seu léxico, essa tecnologia desenvolvida por Kowaltowski, Lucchesi e Stolfi (Kowaltowski & Lucchesi 1993; Kowaltowski et al. 1995b; Kowaltowski et al. 1995a), mas não se tem acesso ao seu código.

Como o Unitex é uma ferramenta de código fonte aberto e possui métodos de compactação e manipulação de léxicos igualmente baseados em autômatos finitos, é nossa intenção utilizá-lo para compactar os dados lexicais atualmente disponíveis na Diadorim, para que outras aplicações de PLN possam acessar essas informações de forma eficiente. Além disso, pretende-se criar recursos lingüístico-computacionais para português brasileiro para que eles possam ser distribuídos juntamente com a ferramenta Unitex.

Para isso, deve ser construído um filtro no intuito de converter os dados da Diadorim para o formato DELA, objetivando manter todas informações gramaticais já disponíveis na base de dados. Com o dicionário convertido, ele pode ser utilizado tanto pela ferramenta Unitex, quanto por qualquer aplicação que necessite de um acesso eficiente a um léxico, o que seria de grande utilidade para qualquer grupo de pesquisa em PLN que necessite de um léxico de grande cobertura e de acesso rápido.

Deverão ser também utilizados outros recursos desenvolvidos pelo NILC para, por exemplo, criar regras de remoção de ambiguidades, funcionalidade esta implementada no Unitex.

Dessa forma, os objetivos desse projeto podem ser resumidos em: a) construir léxicos computacionais para o português brasileiro baseado no formalismo DELA, incorporando todas informações disponíveis na Diadorim, daqui em diante referenciado como dicionários do Unitex-PB; b) construir regras de remoção de ambigüidades lexicais para português brasileiro, seguindo o formalismo utilizado para essas regras no Unitex; c) construir uma biblioteca de acesso e manipulação a esses léxicos, para que outras aplicações, além do Unitex, possam utilizá-los; d) disponibilizar tais recursos não só para o NILC e instituições vinculadas a ele, como também para outros grupos de pesquisa que, porventura, se interessem em utilizá-los para consulta, análise ou compilação de dados da língua portuguesa.

Esta monografia está organizada da seguinte forma: no Capítulo 2 será apresentada uma revisão bibliográfica sobre léxicos computacionais, seus modos de representação e alguns trabalhos relacionados; no Capítulo 3 será apresentada a ferramenta UNITEX, além de seus formalismos; nos Capítulos 4 e 5 são relatados o processo de modelagem, desenvolvimento e validação dos recursos lingüístico-computacionais construídos. O Capítulo 6 traz algumas considerações finais sobre esse trabalho. Finalmente, o Apêndice A apresenta o formato dos dicionários do Unitex-PB e o Apêndice B apresenta

um programa desenvolvido para gerar o dicionário dos verbos.

CAPÍTULO
2

Léxicos Computacionais

Nos últimos anos, como resultado de pesquisas na área de PLN, tornou-se evidente que os recursos lingüísticos e, em particular, os recursos lexicais são de fundamental importância para qualquer sistema de processamento de língua natural. Na verdade, a carência de dados lingüísticos de dimensões reais e, em particular, de léxicos e gramáticas de grande cobertura foi ressaltada com o aumento da quantidade de aplicações lingüístico computacionais (Ranchhod 2001). Atualmente, o léxico não pode ser considerado como uma simples lista de palavras, como nas décadas de 1960 e 1970, mas supõe-se que devam estar contidas num léxico quase todas as informações morfológicas, sintáticas, semânticas e fonológicas de uma língua (Tiberius 1999).

Neste capítulo estaremos abordando o que é um léxico computacional, suas formas de representação e alguns trabalhos relacionados.

2.1 Léxico Computacional

O léxico computacional, ou dicionário, é uma estrutura fundamental para a maioria dos sistemas e aplicações de PLN. Trata-se de uma estrutura de dados contendo os itens lexicais de uma língua e as informações correspondentes a estes itens. Esses itens podem ser palavras isoladas (como *lua, mel, casa, modo*) ou composições de palavras com um significado específico (por exemplo, *lua de mel* ou *Casa de Cultura* ou *a grosso modo*). Entre as informações associadas aos itens lexicais destacam-se as referentes a categoria gramatical (part-of-speech) do item, além de valores para variáveis morfo-sintático-semânticas como gênero, número, grau, pessoa, tempo, modo, regência ver-

bal ou nominal etc. Também são associadas ao item lexical, no léxico, uma ou mais representações ou descrições semânticas. No entanto, associações a representações contextuais são raramente encontradas.

Os léxicos são muito importantes pois são utilizados na fase de análise lexical, uma das primeiras fases durante o processo de processamento de um texto. Sendo assim, se os dicionários utilizados não foram adequados, quer do ponto de vista de cobertura lexical, quer do ponto de vista de formalização, eles podem afetar a qualidade do processamento das fases subseqüentes.

O processo de construção manual ou semi-automática de léxicos é muito custoso, pois demanda tempo e recursos humanos especializados. Isso tem levado muitos pesquisadores a considerar como fonte potencial de informações lexicais as versões eletrônicas de dicionários impressos, que podem ser convertidos de forma automática ou semi-automática em léxicos próprios para PLN.

Em termos gerais, pode-se identificar ao menos cinco tipos de conhecimento que são relevantes para qualquer sistema de PLN (Nunes et al. 1999) e que podem ser associados às entradas lexicais de um léxico computacional. São eles:

1. fonético-fonológico: fonemas, utilizados para depreender a identidade sonora dos elementos que constituem a palavra.

2. morfológico: unidades mínimas dotadas de significado chamadas morfemas, são utilizadas para a compreensão do processo de formação e flexão das palavras.

3. sintático: funções gramaticais que os itens lexicais exercem, usadas para determinar quais funções as palavras desempenham na sentença.

4. semântico: traços semânticos e conhecimento ontológico, utilizados para identificação dos objetos no mundo.

5. pragmático-discursivo: informações extralingüísticas, quando a força expressiva das palavras remete à identificação dos objetos do mundo em termos do seu contexto de enunciação e condições de produção discursiva.

2.2 Modos de representação

Até o início dos anos 80, o processo de desenvolvimento de léxicos e bases de informação lexical era realizado sem grandes preocupações com a padronização na elaboração e organização dos dados utilizados ou mesmo na construção

do recurso propriamente dito, o que tornava a modificação e a reutilização dos dados duas tarefas praticamente impossíveis de serem executadas. A partir de então, vários pesquisadores passaram a se preocupar com a reutilização dos dados e, conseqüentemente, com a diminuição do esforço inicial para o desenvolvimento de novas aplicações (Evans & Kilgarriff 1995).

As principais formas de representação que então surgiram foram: dicionário legível por máquina (*machine-readable dictionary - MRD*), dicionário tratável por máquina (*Machine Tractable Dictionary - MTD*) e, posteriormente, base de dados lexicais (*lexical database*) e base de conhecimento lexical (*lexical knowledge base*). A tipologia utilizada nesse trabalho é baseada nos trabalhos de (Correia 1994; Correia 1996).

2.2.1 Dicionários legíveis por máquinas

A oposição entre dicionário legível por máquina e dicionário tratável por máquina é proposta por (Wilks et al. 1988).

Os MRDs são dicionários feitos por lexicógrafos e concebidos para uso humano. São geralmente dicionários que, ou foram inicialmente construídos em formato digital, ou foram criados no formato papel e posteriormente transferidos para formato digital. Desses dicionários são publicadas versões impressas e versões digitais. A denominação MRD pode, portanto, corresponder a produtos diferentes em termos de concepção e metodologia de trabalho. No entanto, todos estes produtos apresentam como denominador comum as características de serem concebidos para uso humano e de se encontrarem disponíveis em formato digital.

Segundo (Correia 1994), os MRDs, embora se beneficiando das virtudes do formato digital, que se traduzem em grande diversificação e aumento de possibilidades de consulta, não são susceptíveis de serem utilizados diretamente em sistemas de PLN, devido fundamentalmente ao fato de serem concebidos para uso humano, isto é, a informação é dada em língua natural, pouco formalizada, não reconhecível pelos programas de PLN, que pressupõem grande formalização da informação.

A estrutura interna desses dicionários é semelhante a dos dicionários impressos, isto é, basicamente as unidades lexicais são descritas em artigos distintos, apresentando a estrutura tripartida clássica, como visto na Figura 2.1.

Entrada - categoria - definição (eventualmente, exemplificação)

Figura 2.1: Exemplo da estrutura de um dicionário legível por máquina.

Uma outra característica desses dicionários é a ausência de certas flexões para as palavras, como por exemplo, todas flexões de tempo para um verbo. Mas o conhecimento implícito ou mesmo explícito dos humanos que consultam este tipo de obras ajudará a contornar no todo ou em parte esta lacuna.

2.2.2 Dicionários tratáveis por máquina

Segundo (Wilks et al. 1988), MTD é um MRD transformado, apresentando um formato que o torne apto a ser usado em sistemas de PLN. Esta aptidão resulta basicamente na descrição do conhecimento lexical num formalismo no qual o sistema possa facilmente reconhece-lo, traduzindo a informação que nos dicionários humanos é apresentada em língua natural, bem como na explicitação de todo o conhecimento que nos dicionários para uso humano permanece implícito na sua descrição. Os MTDs são, em primeira instância, apenas utilizáveis em sistemas de PLN.

Este tipo de dicionário busca possuir todas as flexões para as palavras, pois lacunas ou incoerência resultantes da não inclusão de certas flexões bastariam por si só para tornar impraticável a utilização do dicionário no processamento automático de texto.

Os dicionários MTDs podem ser construídos de forma exaustiva, onde cada entrada é uma flexão, ou ter como entradas, canônicas associadas às regras de flexões, onde as flexões podem ser geradas de forma automática.

Este trabalho utilizou este formato de dicionário.

2.2.3 Base de dados lexicais

Uma *base de dados lexicais* (BDL) é uma estrutura computacional criada para ser capaz de suportar os mais variados tipos de conhecimento sobre cada unidade lexical, permitindo estabelecer conexões, tanto entre unidades lexicais distintas, quanto entre características pertencentes a unidades lexicais distintas. Isto permite observar e acessar as unidades lexicais sob as mais variadas formas.

Uma das principais características das BDLs, do ponto de vista teórico, é o fato de corresponderem a uma concepção de léxico bastante diferente da dos dicionários: numa BDL, o léxico é entendido como uma complexa rede de relações (morfológicas, sintagmáticas, semânticas, paradigmáticas), onde o conhecimento sobre uma unidade lexical é composto de vários níveis ou camadas. Por outro lado, nos dicionários em geral (dicionários impressos, MRDs ou MTDs), o léxico é encarado como uma listagem de unidades a descrever de forma atomística, não sendo potenciadas (ou, pelo menos, não de modo sistemático e/ou exaustivo) as relações interlexicais (Calzolari 1990).

Por outro lado, nos dicionários em geral (dicionários impressos, MRDs ou MTDs), o léxico é encarado como uma listagem de unidades a descrever de forma atomística, não sendo potenciadas (ou, pelo menos, não de modo sistemático e/ou exaustivo) as relações interlexicais (Calzolari 1990).

A informação contida numa BDL é primeiramente destinada a ser utilizada em sistemas de PLN. Geralmente estas bases são utilizadas como repositórios de informações lexicais de uma determinanda língua, passiveis de serem utilizadas como fonte para diferentes sistemas de PLN.

Dados lexicais são muito mais complexos do que os tipos de dados usados para a maioria das pesquisas na área de banco de dados (Ide & Véronis 1994). Dessa forma, é necessário que seja feito um levantamento sobre os possíveis modelos de representação dos dados e também sobre *sistemas de gerenciamento de bancos de dados* (SGBDs) disponíveis, para que se encontre um modelo adequado à implementação.

Um exemplo de BDL é a Diadorim, projeto de BDL para o português brasileiro desenvolvido no NILC (Greghi et al. 2002). O NILC tem desenvolvido, desde sua criação em 1991, vários aplicativos e recursos lingüísticos para o português brasileiro e a Diadorim é uma BDL que foi desenvolvida incorporando as informações presentes no léxico do NILC[1] e as informações presentes num *thesaurus* da língua portuguesa (da Silva et al. 2000). Seu objetivo é centralizar todas essas informações em uma única base de dados.

Atualmente essa BDL possui cerca de 1,5 milhão de entradas lexicais (palavras simples), representadas em um banco de dados relacional e está acessível em:

http://www.nilc.icmc.usp.br/nilc/tools/intermed.htm

2.2.4 Bases de conhecimento lexical

Uma base de conhecimento lexical (BCL) representa explicitamente uma teoria do léxico, sendo, por isso, um corpo de informação representada num tipo de notação especial - a LRL (*lexical representation language*), que contém uma sintaxe e uma semântica explícitas e que suporta operações lexicais capazes de realizar transformações válidas dessa informação (Briscoe 1991).

Em outras palavras, enquanto uma BDL é concebida como uma representação estática das propriedades das unidades lexicais extraíveis de MRDs, uma BCL é concebida como uma representação dinâmica, à medida que, além de conter informação lexical estruturada, pressupõe a construção de uma LRL capaz de analisar essa informação e de gerar produções lingüísticas. A definição dessa LRL é feita explicitamente de acordo com uma teoria semântica

[1] Maiores detalhes em (Nunes et al. 1996).

determinada. No interior da BCL, é possível *navegar* pelo léxico, caminhando nele através dos conceitos ou relações semânticas, o que o faz se assemelhar conceitualmente a um *thesaurus* (Correia 1996).

2.3 Trabalhos relacionados

A seguir, serão apresentados alguns exemplos de projetos que tratam de construção de léxicos.

2.3.1 Rede Relex

Relex é uma rede informal de laboratórios de grupos de pesquisas europeus[2] (França, Alemanha, Itália, Portugal) que trabalham no domínio de lingüística computacional para a construção de léxicos eletrônicos e gramáticas. Cada grupo trabalha em sua língua nacional e todas as equipes estão usando métodos idênticos. Pelo menos uma vez ao ano eles se encontram para confrontar seus problemas, apresentar seus resultados e adotar futuras padronizações.

Os dicionários obtidos são numerosos e coerentes. Dicionários de tamanhos significativos foram construídos para cada língua e programas que incorporam esses dicionários foram construídos para processar *corpora*. Uma característica muito importante deste trabalho é que, em todos os níveis, os grupos de pesquisa trabalharam nos mesmos itens (dicionários e gramáticas) e que seus resultados parciais têm sido unidos sem grandes dificuldades. A metodologia comum garante a acumulação de dados.

Este projeto utiliza, como formato padrão para os dicionários, o padrão DELA (Silberztein 1990; Courtois 1990) desenvolvido na França (esse padrão será visto com mais detalhes no capítulo 3). Nestes grupos de pesquisa estão sendo desenvolvidos dicionários de palavras simples, de palavras compostas e dicionários fonológicos.

Alguns dicionários desse projeto, como o dicionário do português de Portugal[3], já estão sendo disponibilizados *on-line*. Atualmente, o dicionário do português de Portugal possui 1.250.000 entradas de palavras simples flexionadas e 25.000 entradas de palavras compostas flexionadas[4] (Ranchhod et al. 1999).

Este projeto de mestrado traz contribuições diretas à rede Relex, uma vez que visa a construção dos recursos para português brasileiro, segundo os padrões estabelecidos para esta rede.

[2]Veja `http://ladl.univ-mlv.fr/Relex/introduction.html`
[3]Veja `http://label.ist.utl.pt/pt/resources/resources.htm`
[4]Os conceitos de palavras simples e compostas são apresentados no Capítulo 3.

2.3.2 Projeto ISLE

O projeto ISLE é a continuação da iniciativa EAGLES (*Expert Advisory Group for Language Engineering Standards*), que foi um projeto da União Européia fundado em 1993 ((EAGLES 1993; EAGLES 1996)).

ISLE (*International Standards for Language Engineering*) está dentro do programa *Human Language Technology* (HLT) que faz parte de um projeto de cooperação de pesquisa internacional entre os Estados Unidos e a União Européia e objetiva desenvolver e promover padrões de HLT, desenvolver *guidelines* e recomendações de boas práticas para recursos lingüísticos e desenvolver ferramentas que utilizem esses recursos. Os objetivos do EAGLES/ISLE são recursos lingüísticos de larga escala (como córpus de textos escritos e falados e léxicos computacionais), meios de manipular estes conhecimentos via formalismos de lingüística computacional, linguagens de marcação e várias ferramentas e meios de avaliação (Calzolari et al. 2001).

Os atuais alvos do projeto ISLE[5] estão em três áreas: léxicos computacionais multilinguais, interação natural e avaliação de sistemas de HLT.

Para léxicos computacionais multiliguais, o ISLE está trabalhando em: estender o trabalho do EAGLES em semântica lexical, necessária para estabelecer relações inter-línguas; projeto e proposta de padrões para léxicos multilinguais, desenvolvimento de um protótipo de ferramenta para implementar padrões e guidelines para léxicos; desenvolvimento de exemplos de léxicos e corpora com propósito de validação; e desenvolver procedimentos de avaliação padronizados para léxicos (Calzolari et al. 2002).

O grupo de trabalho com léxicos computacionais do ISLE (*Computational Lexicon Working Group* - CLWG) tem como objetivo principal a definição da entrada multilingual lexical do ISLE (*Multiligual ISLE Lexical Entry* - MILE), que é um esquema geral de codificação de informações lexicais multilinguais.

O MILE visa uma arquitetura altamente modular e dividida em camadas e será construído um ambiente que ajudará na construção destas entradas lexicais, e que é constituído de três componentes: um XML DTD como um modelo de entidade relacional; um repositório de dados de categorias lexicais que serão usados de uma maneira fácil para construir entradas lexicais no formato MILE; e a estação lexicográfica ISLE, que irá organizar o modelo entidade relacional em um banco de dados relacional, que ainda incluirá uma interface gráfica de entrada, navegação e procura de dados de uma maneira amigável (Atkins et al. 2002).

Ultimamente o grupo está se ampliando para incluir também línguas asiáticas e algumas contribuições já podem ser vistas como a participação de lín-

[5]Veja `http://www.ilc.pi.cnr.it/EAGLES96/isle/ISLE_Home_Page.htm`

guas como chinesa, japonesa, coreana e tailandesa em alguns workshops.

2.3.3 WordNet

WordNet é um sistema de referência lexical do inglês, *on-line*[6], desenvolvido por um grupo de pesquisadores no *Cognitive Science Laboratory*, na Universidade de Princeton, nos EUA.

Esse sistema baseou-se em teorias psicolingüísticas concernentes à organização do léxico na memória humana, ou seja, o léxico mental (Miller et al. 1990). Ele tenta organizar as informações lexicais em termos do significado das palavras, mais do que de suas formas, o que torna o sistema mais semelhante a um *thesaurus* do que a um dicionário propriamente dito.

A WordNet foi desenvolvida para o tratamento da língua inglesa e ela divide o léxico em cinco categorias: substantivos, verbos, adjetivos, advérbios e palavras funcionais. Porém, esse sistema atualmente possui substantivos, verbos, adjetivos e advérbios. Esse tipo de categorização o diferencia de um dicionário tradicional e, apesar de causar certa redundância nas informações armazenadas (algumas palavras podem ser classificadas em mais de uma categoria), traz a vantagem de que diferenças fundamentais na organização semântica dessas categorias sintáticas podem ser claramente observadas e facilmente exploradas.

A idéia básica utilizada na WordNet é a representação das palavras e de seus significados em uma matriz lexical. O mapeamento entre as formas e seus significados é N:N, ou seja, algumas formas podem ter diferentes significados, e alguns significados podem ser expressos por várias formas diferentes.

A WordNet distingue relações semânticas de relações lexicais. O significado de uma palavra P1 pode ser representado por uma lista de palavras que podem ser usadas para expressar P1 {S1, S2,...}. Uma matriz lexical consiste, assim, em um mapeamento entre palavras e conjuntos de sinônimos (*synsets*). Os sinônimos são relações lexicais entre palavras. A Tabela 2.1 é um exemplo de ilustração de uma matriz lexical. As formas das palavras (*word forms*) estão nos cabeçalhos das colunas e os significados (*word meanings*) nos cabeçalhos das linhas. Uma entrada em uma célula da matriz implica que a forma naquela coluna pode ser usada (em um contexto apropriado) para expressar o significado daquela linha. Deste modo, a entrada $E_{1,1}$ significa que a forma F_1 pode ser usada para expressar o significado M_1. Se existem duas entradas na mesma coluna, a forma é polissêmica. Se existem duas entradas na mesma linha, as duas formas são sinônimos (relativos a um contexto).

O modelo da WordNet tem sido amplamente adotado em outros proje-

[6]Veja `http://www.cogsci.princeton.edu/~wn/`

Tabela 2.1: Ilustrando o Conceito de Matriz Lexical.

Significado das Palavras	Forma das Palavras				
	F_1	F_2	F_3	...	F_n
M_1	$E_{1,1}$	$E_{1,2}$			
M_2		$E_{2,2}$			
M_3			$E_{3,3}$		
$\vdots$				$\ddots$	
M_m					$E_{m,n}$

tos de mesma natureza para outras línguas. Temos, como exemplo, o projeto WordNet-BR (da Silva et al. 2002; da Silva 2003), para o português brasileiro, em desenvolvimento no NILC[7], o qual deu origem a um aplicativo que, acoplado a ferramentas de auxílio à escrita, possibilita ao usuário consultar e escolher palavras sinônimas e antônimas durante o processo de escrita, chamado TeP[8] (*Thesaurus Eletrônico para o Português do Brasil*).

2.4 Considerações finais

Neste capítulo foram apresentados a definição de Léxico Computacional, que é um recurso fundamental para qualquer aplicação de PLN, suas formas de representação e trabalhos importantes que resultaram na construção de léxicos. Este projeto de mestrado utilizou os padrões da rede Relex e uma das ferramentas utilizadas nesta rede é o Unitex que será descrito no próximo capítulo.

[7]Veja `http://www.nilc.icmc.usp.br/nilc/projects/wordnetbr.htm`
[8]Veja `http://www.nilc.icmc.usp.br/nilc/tools/Bentotep.htm`

CAPÍTULO 3

Ferramenta Unitex

O Unitex[1] é um ambiente de desenvolvimento lingüístico que pode ser utilizado para analisar córpus de muitos milhões de palavras em tempo real. As descrições lingüísticas são formalizadas através de dicionários eletrônicos (léxicos) e gramáticas de grandes dimensões, representadas por autômatos de estados finitos.

Este sistema utiliza tecnologias criadas pelo *Laboratoire d'Automatique Documentaire et Linguistique*[2] (LADL), fundado em 1967 por Maurice Gross, na *Université de Marne-la-Vallée*, na França. O Unitex é uma versão de código fonte aberto de uma outra ferramenta, também desenvolvida no LADL, o INTEX. O INTEX foi criado em 1992 e sua versão inicial era para NextStep, sendo que em 1996 ele foi completamente integrado a uma interface gráfica (versão 3.0) e começou a ser distribuído para centros de pesquisa como um ambiente lingüístico de pesquisa (Silberztein 2000b).

Somente em outubro de 2002 foi lançada a primeira versão do Unitex, tendo como principal programador Sébastien Paumier, pesquisador que trabalhou no LADL. O sistema é distribuído livremente sob os termos da General Public License[3] (GPL). Portanto, todos têm acesso ao código fonte da aplicação e podem modificá-lo seguindo os termos da licença GPL.

Hoje em dia, mais de 200 laboratórios de pesquisa em vários países utilizam o INTEX e/ou Unitex como uma ferramenta de pesquisa ou educacional. Alguns usuários estão interessados nas funcionalidades de processamento de córpus (na análise literária de textos, pesquisando informações em jornais ou documentos técnicos, etc); outros estão utilizando esta plataforma para for-

[1] Veja `http://www-igm.univ-mlv.fr/~unitex/`
[2] Veja `http://ladl.univ-mlv.fr/`
[3] Veja `http://www.gnu.org/licenses/gpl.html`

malizar certos fenômenos lingüísticos (por exemplo, descrevendo morfologia, léxico e expressões da língua), ou ainda por seu poder computacional (análise automática de textos).

Países como Alemanha, Coréia, Eslovênia (Vitas & Krstev 2001), Espanha, França, Grécia (Anastasiadis-Symeonidis et al. 2000), Itália (Vietri & Elia 2000), Noruega, Polônia, Portugal (Ranchhod et al. 1999) e Tailândia entre outros, estão trabalhando para a construção de seus próprios dicionários lexicais para o sistema INTEX/Unitex.

A Figura 3.1 apresenta a arquitetura geral da ferramenta Unitex, que é utilizada neste projeto e que será apresentada, bem como seus formalismos, neste capítulo.

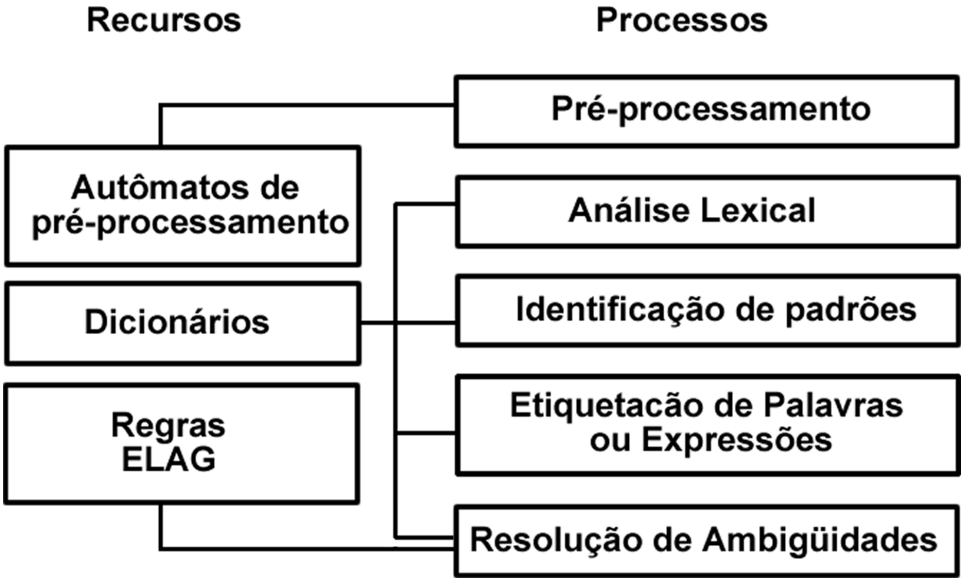

Figura 3.1: Arquitetura da ferramenta - processos e recursos.

3.1 Formalismo dos dicionários: Padrão DELA

O sistema Unitex utiliza um conjunto de dicionários eletrônicos em um formalismo concebido pelo LADL para o francês conhecido como DELA (*Dictionnarie Electronique du LADL*). Este formalismo permite declarar entradas lexicais simples e compostas de uma língua, que podem estar associadas a informações gramaticais e às regras de formação de flexões. Esses dicionários são instrumentos lingüísticos especificamente concebidos para serem utilizados em operações automáticas de processamento de texto.

Os dicionários utilizados pelo Unitex para identificar palavras em um texto são os dicionários de palavras flexionadas, DELAF (DELA de palavras Flexionadas) ou o DELACF (DELA de palavras Compostas Flexionadas). Esses dicionários são geralmente gerados automaticamente a partir dos dicionários

DELAS (DELA de palavras Simples) (Courtois 1990) e DELAC (DELA de palavras Compostas) (Silberztein 1990).

3.1.1 Dicionários de palavras simples - DELAS e DELAF

Os dicionários DELAS e DELAF correspondem à representação das palavras simples. Os dicionários de palavras simples são basicamente formados de uma lista de palavras e estão associadas, a cada uma delas, informações sobre a sua categoria gramatical e sobre seu modelo de flexão. As informações relacionadas às palavras variam de acordo com a classe gramatical e dizem respeito à variação em gênero, número, grau, caso (para pronomes), tempo, modo e pessoa. Essas informações são fundamentalmente morfológicas, mas esse formalismo contempla a possibilidade de se acrescentar progressivamente informações sintáticas e semânticas.

O DELAS, como dito anteriormente, é o dicionário de palavras simples, que são seqüências de caracteres alfabéticos delimitadas por separadores. Um separador é um caractere não alfanumérico.

As entradas do DELAS possuem a seguinte estrutura:

<palavra>, <descrição formal>

onde *palavra* representa a forma canônica (o lema) de uma unidade lexical simples. O lema é representado, no caso dos verbos, como a sua forma no infinitivo, no caso dos substantivos e adjetivos, pela sua forma no masculino singular (quando aplicável) ou pelo feminino singular quando só se tem esse gênero, etc. As categorias invariáveis (a maioria dos advérbios, preposições, conjunções e alguns determinantes) são representadas pela sua única forma. Em alguns raros casos, as formas canônicas são representadas por formas plurais (adeus, pâncreas, húmus, etc.) que não variam em gênero. Apesar de serem invariáveis, o plural tem de ser explicitamente marcado, a fim de permitir concordância a nível sintático. A vírgula separa o lema do código de flexão (Ranchhod 2001). Na Figura 3.2 estão exemplos reais de entradas no dicionário DELAS para português brasileiro.

mato, N001D026A01

beijar,V005

melodicamente,ADV

Figura 3.2: Exemplos de entrada paro o dicionário DELAS.

A primeira informação sobre a palavra é a indicação da categoria gramatical a que pertence. Nos exemplos: substantivo (*N*), verbo (*V*) e advérbio (*ADV*). O

código numérico indica o modelo de flexão da categoria em questão. *Mato*, por exemplo, é um substantivo e a sua regra de flexão de gênero e número é a 001. Os códigos *D* e *A*, seguidos de um código numérico adicional, permitem gerar, respectivamente, o(s) diminutivo(s) e o(s) aumentativo(s) adequado(s). Vale lembrar que, no caso de uma palavra se enquadrar em mais de uma categoria gramatical, para cada categoria, deverá haver uma entrada no DELAS, por exemplo, a entrada flexionada *mato* pode ser gerada pela seguintes entradas no DELAS: *matar,V030* e *mato,N001D026A01*.

Cada código representa uma regra de flexão, que é formalizada num transdutor de estados finitos (*FST - Finite State Transducer*). Os FST flexionais associam conjuntos de sufixos às informações das entradas lexicais do dicionário de lemas (DELAS) e geram as correspondentes formas flexionadas. Por exemplo, a flexão representada pelo código *N001D026A01* (substantivo *mato*, por exemplo) é descrita pelo transdutor *N001D026A01* da Figura 3.3.

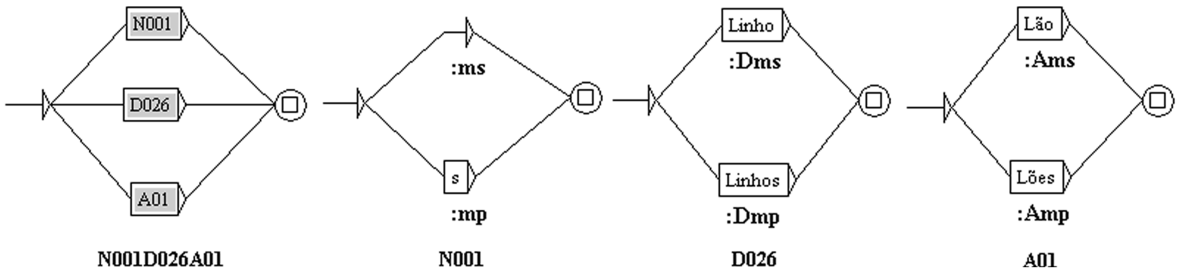

Figura 3.3: Autômatos que representam a regra de flexão *N001D026A01*.

Este FST contém três sub-transdutores: *N001*, que descreve e gera a flexão de gênero e número, *D026* que gera as formas diminutivas, e *A01*, que gera as formas aumentativas. Todos substantivos, que no dicionário de lemas estejam codificados como *N001D026A01*, são flexionados por aplicação destes transdutores. Substantivos igualmente terminados em *-o* mas que não admitam diminutivo nem aumentativo estão classificados como *N001* e a sua flexão é gerada apenas pelo transdutor *N001*.

A flexão é gerada de forma automática através dos FSTs. No caso da regra *N001*, dois caminhos são possíveis. O primeiro não modifica a forma canônica e acrescenta o código flexional *:ms*. O segundo acrescenta o sufixo *s* e o código flexional *:mp*. Já na regra *D026*, também dois caminhos são possiveis, mas em ambos é utilizado o operador *L*, que retira uma letra antes de acrescentar o sufixo (*inho* e *inhos*). Dois operadores são possíveis: *L* (*Left*) que tira uma letra da entrada e *R* (*Right*) que recoloca uma letra na entrada.

As formas flexionadas de todas as entradas simples constituem o dicionário DELAF (DELA de palavras flexionadas), gerado automaticamente a partir do DELAS. As flexões das entradas *mato, N001D026A01*, *beijar,V005* e *melodica-*

mente,ADV são apresentadas na Figura 3.4.

Assim, as entradas do DELAF são constituídas da palavra, seu lema, a sua categoria gramatical e a flexão que corresponde a essa forma (por exemplo:*N:Amp*, substantivo masculino plural no aumentativo). No caso dos verbos, existem, sobretudo se forem regulares, várias formas homógrafas. Por exemplo, *beijamos* é potencialmente a primeira pessoal do plural do presente do indicativo (*P1p*) e do pretérito simples (*J1p*) de *beijar*.

As contrações são um caso especial de entradas no dicionário de palavras simples. Apesar de, sob o ponto de vista lingüístico, não poderem ser assim classificadas, pois são constituídas por duas palavras pertencentes ou não a duas categorias distintas, apresentam-se ortograficamente como se fossem palavras simples. Por essa razão, elas recebem tratamento específico. No DELAF, a estrutura deste tipo de entrada é exemplificada na Figura 3.5.

No caso das contrações, a noção de lema é uma aproximação grosseira. As categorias a que pertencem os constituintes das formas contraídas estão delimitados por "*X*": *daqui* é constituída por uma preposição e por um advérbio (*PREPXADV*), ambos categorias invariáveis. Quando as palavras de base flexionam, a informação sobre a respectiva flexão vem imediatamente à direita de cada elemento, especificada por ":". *Das*, por exemplo, corresponde à contração da preposição *de* e do artigo definido feminino plural *as* (*Art+Def:fp*).

3.1.2 Dicionários de palavras compostas

Os dicionários DELAC e DELACF correspondem à representação das palavras compostas. Palavras compostas são unidades lexicais constituídas por uma combinação fixa de palavras simples, que representam uma parte significante de um léxico de qualquer língua. Essas palavras apresentam restrições às propriedades que as palavras teriam individualmente.

A estrutura das entradas dos dicionários de compostos (DELAC) é um pouco diferente da do dicionário de palavras simples. Ainda está em estudos a formalização deste dicionário, mas um exemplo da proposta atual utilizada pelos membros da rede RELEX pode ser visto na Figura 3.6.

A palavra composta *bom gosto* é um substantivo do tipo morfológico adjetivo+substantivo (*AN*). Este lema é masculino singular e não flexiona. *Belprazer* é um substantivo do tipo morfológico substantivo+substantivo (*NN*). Ele está sujeito a flexão de número (+*N*), seu lema está no masculino singular. O último membro desta palavra composta é a flexão masculino singular do lema *prazer* e seu código de flexão é *N003*. A última palavra composta do exemplo é *lateral esquerdo*, que é um substantivo do tipo morfológico substantivo+adjetivo (*NA*). Seu lema está no masculino singular e ele está sujeito a flexão de número (+*N*). O primeiro membro da palavra composta é a flexão

beija,beijar.V:P3s:Y2s
beijada,beijar.V:K
beijadas,beijar.V:K
beijado,beijar.V:K
beijados,beijar.V:K
beijai,beijar.V:Y2p
beijais,beijar.V:P2p
beijam,beijar.V:P3p
beijamos,beijar.V:P1p:J1p
beijando,beijar.V:G
beijar,beijar.V:W1s:W3s:U1s:U3s
beijara,beijar.V:Q1s:Q3s
beijaram,beijar.V:J3p:Q3p
beijaras,beijar.V:Q2s
beijardes,beijar.V:W2p:U2p
beijarei,beijar.V:F1s
beijareis,beijar.V:F2p
beijarem,beijar.V:W3p:U3p
beijaremos,beijar.V:F1p
beijares,beijar.V:W2s:U2s
beijaria,beijar.V:C1s:C3s
beijariam,beijar.V:C3p
beijarias,beijar.V:C2s
beijarmos,beijar.V:W1p:U1p
beijará,beijar.V:F3s
beijarás,beijar.V:F2s
beijarão,beijar.V:F3p
beijaríamos,beijar.V:C1p
beijaríeis,beijar.V:C2p
beijas,beijar.V:P2s
beijasse,beijar.V:T1s:T3s
beijassem,beijar.V:T3p
beijasses,beijar.V:T2s
beijaste,beijar.V:J2s
beijastes,beijar.V:J2p
beijava,beijar.V:I1s:I3s
beijavam,beijar.V:I3p
beijavas,beijar.V:I2s
beije,beijar.V:S1s:S3s:Y3s
beijei,beijar.V:J1s
beijeis,beijar.V:S2p
beijem,beijar.V:S3p:Y3p
beijemos,beijar.V:S1p:Y1p
beijes,beijar.V:S2s
beijo,beijar.V:P1s
beijou,beijar.V:J3s
beijáramos,beijar.V:Q1p
beijáreis,beijar.V:Q2p
beijásseis,beijar.V:T2p
beijássemos,beijar.V:T1p
beijávamos,beijar.V:I1p
beijáveis,beijar.V:I2p
matinho,mato.N:Dms
matinhos,mato.N:Dmp
mato,mato.N:ms
matos,mato.N:mp
matão,mato.N:Ams
matões,mato.N:Amp
melodicamente,melodicamente.ADV

Figura 3.4: Flexões das entradas *mato*, *beijar* e *melodicamente* no Dicionário DELAF.

```
das,do.PREPXDET+Art+Def:fp
daqui,daqui.PREPXADV
naquele,naquele.PREPXDET+Dem:ms
```

Figura 3.5: Exemplos de contrações no dicionário DELAF.

```
bom gosto,N+AN:ms
bel-prazer(prazer.N003:ms),N+NN:ms/+N
lateral(lateral.N007:ms) esquerdo,N+NA:ms/+N
```

Figura 3.6: Exemplos de entradas no DELAC.

masculino singular do lema *lateral* e seu código de flexão é *N007*.

O arquivo DELAC é dividido em subarquivos. Cada subarquivo contém todas palavras compostas de mesma topologia e paradigma de flexão.

Cada subarquivo começa com um preâmbulo que descreve o número de membros com que a palavra composta é formada, o membro principal da palavra composta (+) e membros não principais (-) (Savary 2000). Veja um exemplo na Figura 3.7.

```
#-/+
bom gosto,N+AN:ms
bel-prazer(prazer.N003:ms),N+NN:ms/+N
#+/-
lateral(lateral.N007:ms) esquerdo,N+NA:ms/+N
```

Figura 3.7: Exemplo da estrutura do arquivo DELAC.

O resultado da flexão do DELAC é o dicionário DELACF. O resultado da flexão dos exemplos da Figura 3.7 são as entradas do DELACF na Figura 3.8.

3.2 Formalismo de eliminação de ambigüidades lexicais: Padrão ELAG

O sistema Unitex, ao processar um córpus, realiza análise lexical utilizando as informações contidas em dicionários e é comum encontrar como resultado desta análise muitas palavras com ambigüidade devido à homografia. Para re-

bom gosto, bom gosto.N+AN:ms

bel-prazer,bel-prazer.N+NN:ms

bel-prazeres,bel-prazer.N+NN:mp

laterais esquerdo,lateral esquerdo.N+NA:mp

lateral esquerdo,lateral esquerdo.N+NA:ms

Figura 3.8: Exemplo de entradas do DELACF.

mover essas ambigüidades Eric Laporte e Anne Monceaux desenvolveram um formalismo no qual é possivel, através de gramáticas, remover ambigüidades lexicais. Este formalismo está integrado ao Unitex e será apresentado nesta seção. O texto é baseado no artigo de Laporte & Monceaux (1998). (Laporte & Monceaux 1998).

3.2.1 Ambigüidade lexical no Unitex

Para os usuários do sistema Unitex, um dos maiores problemas é a presença de ambigüidades lexicais artificiais em etiquetas associadas às palavras durante o processo de análise lexical. Isto ocorre pois o sistema utiliza léxicos de grande cobertura, recuperando todas as etiquetas possíveis para uma palavra, mas não leva em consideração o contexto em que se encontra a palavra. O termo ambigüidade artificial é utilizado para enfatizar que essas palavras com várias etiquetas geralmente não são consideradas ambíguas em um contexto por leitores humanos, ao contrário de ambigüidades efetivas, isto é, sentenças que têm várias possibilidades de interpretação.

A ambigüidade lexical existe quando dois ou mais elementos lingüísticos se escrevem exatamente da mesma forma, isto é, quando são palavras homógrafas. Por exemplo na Figura 3.9, nas duas frases:

A casa está fechada.

Ela se casa amanhã na Igreja São Paulo.

Figura 3.9: Exemplo de ambigüidade por homografia.

As duas formas *casa*, um substantivo e um verbo, são idênticas e são pronunciadas do mesmo modo, mas têm definições completamente diferentes. Há ainda casos em que as palavras são idênticas porém as pronuncias são diferentes, mas em ambos existe a ambigüidade lexical. Embora um leitor humano não detecte essas ambigüidades, a necessidade de associar duas

etiquetas diferentes a um verbo e a um nome, como etapa preparatória ao reconhecimento da estrutura sintática das frases, por exemplo, é evidente.

Este problema se intensifica de acordo com o aumento de informações contidas nas etiquetas do léxico e com a quantidade de informações lexicais que são exatamente os objetivos que queremos atingir utilizando léxicos de grande cobertura e o sistema Unitex.

A principal solução para este problema é a análise sintática, operação esta que determina a etiqueta correta, ou as etiquetas corretas, para cada palavra. Para alguns casos, somente a análise sintática pode resolver todas as ambigüidades lexicais.

Entretanto, em muitos casos, uma simples restrição envolvendo informações lingüísticas básicas sobre as palavras num contexto imediato basta para resolver muitas das ambigüidades lexicais artificiais em um texto.

Este tipo de solução é o objetivo do sistema de resolução de ambigüidade lexical, que pode ser expresso como:

- Remover o máximo possível das análises incorretas, o que não implica que esperamos remover todas, com a estrita restrição de nunca descartar uma análise correta.

A principal finalidade de um sistema de resolução de ambigüidade é atingir esse objetivo explorando somente o contexto local das palavras marcadas como ambíguas após a etiquetagem do léxico.

3.2.2 O formalismo ELAG

O formalismo ELAG, do inglês *Elimination of Lexical Ambiguities by Grammars*, que significa Eliminação de Ambigüidades Lexicais por Gramáticas, foi desenvolvido por Éric Laporte e Anne Monceuax (Laporte & Monceaux 1998) na França. Este sistema foi desenvolvido para ser compatível com os sistemas INTEX/Unitex.

As seguintes premissas são seguidas pelo formalismo ELAG: análises corretas não devem ser removidas; os resultados de análise sintática não podem ser explicitamente utilizados, uma vez que eles não estão disponíveis quando a resolução de ambigüidade lexical é aplicada ao texto; a análise lingüística que desejamos aplicar à sentença deve ser levada em consideração, o que implica que o criador das gramáticas de resolução de ambigüidade lexical tem visões particulares sobre o resultado desejado da análise sintática.

ELAG é compatível com Unitex, e o conjunto de etiquetas que podem ser aplicadas às palavras é o conjunto de etiquetas do Unitex. Estas etiquetas aparecem em várias formas no Unitex, mas no ELAG elas aparecem como <*canônica.classe_gramatical+subcategoria:informações_de_flexão*>, que inclui:

- a canônica ou lema da unidade lexical. Em <*chorar.V:J3s*>, por exemplo, esta forma é *chorar*, enquanto que a forma flexionada descrita por essa etiqueta é *chorou*. A forma flexionada ocorre no texto e não é representada explicitamente numa etiqueta. A canônica pode ser um nome composto;

- a classe gramatical, No exemplo, *V* para verbo;

- se a classe gramatical permite, uma série de informações de flexão. No exemplo *J3s* para modo indicativo pretérito, tempo perfeito, terceira pessoa do singular. Quando uma palavra, por exemplo *chove* é ambigüa devido a vários valores de flexão para uma mesma canônica: <*chorar.V:S1s*>, <*chorar.V:S3s*> e <*chorar.V:Y3s*> *chore*, cada uma destas etiquetas é uma etiqueta completa; o formato <*chorar.V:S1s:S3s:Y3s*> é uma abreviação;

- imediatamente à direita da classe gramatical, a etiqueta lexical pode incluir uma subcategoria, como em <*cujo.PRO+Rel:ms*>, onde *Rel* indica que o pronome é relativo.

A informação lexical coberta pelas etiquetas depende do dicionário. Desde que as etiquetas abranjam todas as informações armazenadas no léxico, podemos chamá-las de etiquetas completas. Etiquetas completas são usadas na representação de textos etiquetados. No ELAG, elas podem também aparecer em regras formalizando restrições que são específicas para uma palavra em particular. Por exemplo, vamos expressar formalmente que a palavra *também* não pode ser etiquetado como uma conjunção quando ela é sucedida por um delimitador de sentença. Vamos assumir que o Unitex reconhece delimitadores de sentença e que o dicionário descreve *também* como (pelo menos) um advérbio. A regra ficaria da seguinte maneira:

> Se <*também.CONJ*> é seguido por uma pontuação,
> Então a pontuação não pode ser um delimitador de sentença.

No ELAG uma regra inclui uma parte "*se*" e uma ou várias partes "*então*". Para expressar esta restrição formalmente, nós iremos usar o símbolo *{S}* que representa o delimitador de sentença. Outros símbolos não verbais são representados por sua própria forma: , / ‘ e - todos são considerados etiquetas. <*PNC*> é uma variável que representa que todos estes símbolos, e <*!{S}.PNC*> representa qualquer <*PNC*> exceto *{S}*. Com essa notação, nossa restrição se torna:

> Se <*também.CONJ*> é seguida por <*PNC*>,
> Então este <*PNC*> é <*!{S}.PNC*>

Desta forma, a parte "*se*" descreve o padrão <*também.CONJ*> <*PNC*> e a parte "*então*" descreve somente <*!{S}.PNC*>. Neste padrão a parte "*se*" é sempre delimitada por três caixas com <*!*> e a parte "*então*" por três caixas com <=>, e a regra seria como na Figura 3.10.

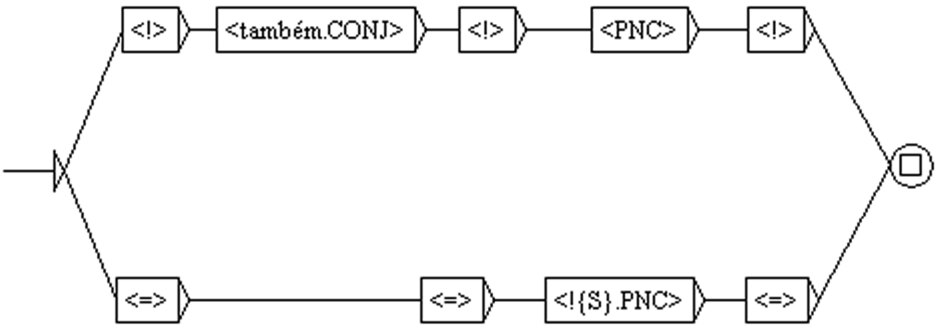

Figura 3.10: Grafo correspondente à regra de restrição.

As caixas com <*!*> e <=> são delimitadores usados para reconhecer as estruturas das regras. Todas as outras caixas possuem elementos lingüísticos que são procurados nas sentenças de entrada quando as regras são aplicadas ao texto. Os símbolos <*!*> e <=> da esquerda e da direita são usados para deixar a regra mais legível. Os símbolos <*!*> e <=> centrais são os únicos elementos de sincronização entre a parte "*se*" e a parte "*então*". Quando a regra da Figura 3.10 é aplicada a uma sentença, o ELAG verifica se toda vez que <*também.CONJ*> é seguido imediatamente por <*PNC*>, o <*PNC*> tem que obedecer a restrição <*!{S}.PNC*> em todas análises, e removerá todas análises que não obedecem à restrição. Quando *também* finaliza uma sentença, a etiqueta *CONJ* será removida corretamente pois ela desobedece a restrição, mas a etiqueta *ADV* irá permanecer intacta pois ela não é atingida pela regra.

O reconhecimento do contexto geralmente envolve o uso de informações contidas nas etiquetas da análise lexical como, por exemplo, a classe gramatical, como <*N*>, que representa qualquer substantivo. Esta é uma etiqueta variável, pois representa um conjunto de etiquetas lexicais. A formalização das etiquetas variáveis é um elemento importante no formalismo de descrição das restrições gramaticais. Nas etiquetas variáveis, a classe gramatical pode ser combinada com:

- um ou vários valores de flexão: <*A:s*> para qualquer adjetivo no singular, <*V:2p*> para qualquer verbo na segunda pessoa do plural;
- ou com a forma canônica: <*menino.N*>, <*menino.N:f*>, etc;
- ou com uma ou mais formas canônicas e pontos de exclamação, representando uma etiqueta variável de negação, por exemplo, <*!ser!estar.V*>

que representa qualquer verbo, exceto ser e estar.

Quando uma forma flexionada é ambígua entre vários valores de flexão para a mesma forma canônica, por exemplo, <*morrer.V:P1p*> e <*morrer:J1p*>, a forma <*morrer.V:P1p:J1p*> pode ser vista com uma etiqueta variável ou como uma abreviação de uma lista de etiquetas completas.

Estas convenções são um pouco diferentes das convenções utilizadas pelo Unitex para formalizar identificação de padrões em textos (Veja a subseção 3.3.2). Para identificar padrões o Unitex aceita palavras sem informações associadas. Elas representam toda etiqueta de que faz parte e que esteja presente no dicionário. Por exemplo, a palavra *feito*, representa pelo menos 4 etiquetas: <*feitar.V:P1s*>, <*fazer.V:K*>, <*feito.N:ms*> e <*feito.A:ms*>.

Já no ELAG, qualquer etiqueta deve incluir no mínimo uma categoria gramatical, como <*feito.A:ms*>, <*feito.A*>, <*A:fs*>, exceto a etiqueta universal <> que representa qualquer etiqueta. Com essa convenção, não é possivel considerar *menininho* como etiqueta, mas sim <*menino.N:Dms*>.

Para especificar uma regra ELAG, a forma mais simples seria uma regra com somente uma parte "*então*". As caixas com <*!*> e <=> delimitam os padrões R1, R2,C1, C2, como na Figura 3.11. Qualquer um desses quatro padrões podem ser vazios.

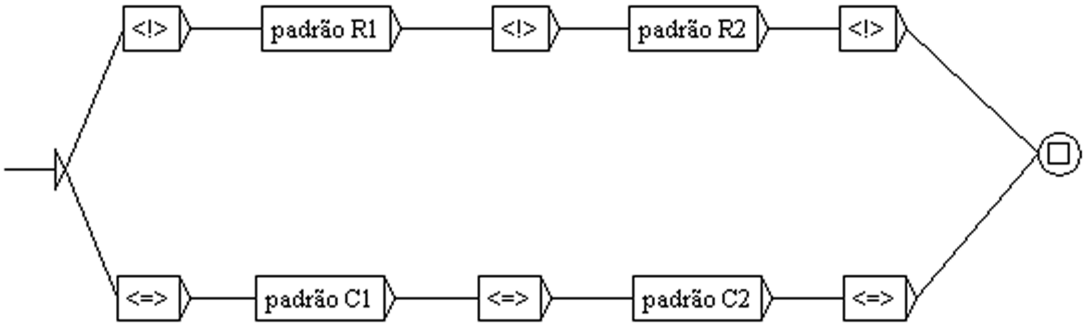

Figura 3.11: Forma geral de uma regra ELAG com uma parte "*então*".

A restrição expressa pela regra na Figura 3.11 indica que toda vez que ocorra R1 imediatamente sucedido pela ocorrência de R2 numa análise de uma sentença, então a ligação entre eles deve obedecer à ocorrência de C1 seguido da ocorrência de C2 na mesma análise. O efeito desta regra é remover todas análises que não obedecem a essa restrição.

Não existe restrição quanto ao tamanho da seqüência a que pertencem aos padrões R1, R2,C1 e C2. Na realidade, cada um destes padrões representa um conjunto de seqüência de etiquetas que podem ou não ter o mesmo tamanho.

O padrão ELAG tem uma excelente característica: as restrições gramaticais expressas como regras separadas são independentes umas das outras e

não mostram interação. Uma dada regra tem um efeito específico ao texto, e esse efeito seria o mesmo caso essa regra seja usada com outras regras ou não. Quando se adiciona uma nova regra para uma gramática existente, o formalismo assegura que o efeito da gramática existente não será modificado. A nova regra somente pode rejeitar mais análises. Isto quer dizer que quando é adicionada uma nova regra à gramática, a taxa de redução de ambigüidade lexical pode aumentar mas nunca diminuir.

Além disso, a ordem de aplicação das regras é indiferente: um conjunto de regras pode ser aplicado em qualquer ordem, sem nenhuma modificação no resultado final.

3.3 Funcionalidades da ferramenta Unitex

O Unitex é um sistema baseado no uso de grandes dicionários lexicais. Ele pode ser usado para analisar textos de muitos milhões de palavras. Inclui vários dicionários e gramáticas embutidos representados como autômatos de estados finitos, porém, o usuário pode adicionar seus próprios dicionários e gramáticas. Estas ferramentas são aplicadas ao texto para localizar padrões léxicos e sintáticos, gerar dicionários lexicais, remover ambigüidades e etiquetar palavras simples como também expressões complexas. Ele pode ser utilizado por lingüistas para análise de córpus, mas também pode ser visto como um sistema de recuperação de informação.

3.3.1 Pré-processamento e análise lexical

Todos textos que podem ser utilizados pelo Unitex devem ser gravados no formato Unicode[4] e com a codificação *little endian* (codificação padrão para processadores da família Intel). Esta conversão deve ser feita antes que o arquivo seja aberto pelo Unitex.

Depois de aberto o texto pelo sistema, ele passa por um pré-processamento. Durante esta fase, primeiro são identificados todos os *tokens* (palavras, sinais de pontuação, delimitadores de sentenças e algarismos). Os resultados desta fase ou de qualquer outra operação no Unitex são sempre salvos em arquivos auxiliares, mantendo sempre o texto original intacto.

Após a identificação dos *tokens*, é aplicado ao texto um autômato no modo *Merge*, que segmenta o texto em frases. Este autômato, na verdade, é uma gramática específica, representada por um FST, que insere o símbolo *{S}* entre frases consecutivas. Esta gramática consegue identificar quando uma pontuação é ou não um delimitador de sentença.

[4]Veja `http://www.unicode.org/`

Identificadas as sentenças, é então aplicado ao texto um autômato no modo *Replace*, que tem como função decompor as contrações, reconstituindo e etiquetando os elementos de que são formados.

Tanto os autômatos de reconhecimento de sentença quanto o de substituição de formas contraídas são dependentes da língua e devem ser construídos cuidadosamente levando em consideração as singularidades de cada idioma. Exemplo da janela com as opções de pré-processamento no Unitex é mostrado na Figura 3.12.

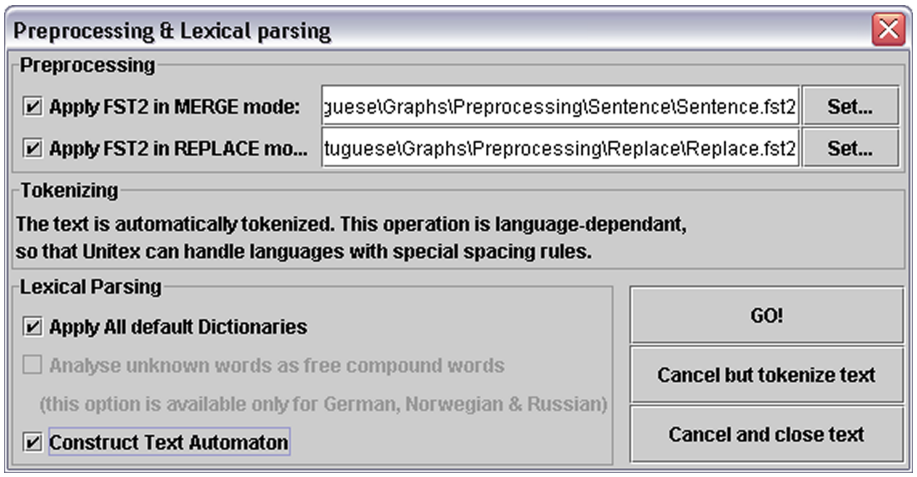

Figura 3.12: Exemplo da janela com as opções de pré-processamento.

Após a fase de pré-processamento é realizada então a análise lexical. Nesta fase são utilizados os dicionários. Na análise lexical, a todos os *tokens* identificados na fase anterior são associadas as informações contidas nos dicionários de palavras flexionadas simples e compostas (DELAF, DELACF). Como resultado desta fase, são gerados três dicionários: o dicionário das palavras simples contidas no texto, o dicionário das palavras compostas contidas no texto e o dicionário das palavras simples desconhecidas (palavras que estão no texto mas não estão nos dicionários), que geralmente são palavras mal formadas ou nomes próprios não cadastrados.

Durante a análise lexical também é possível construir o autômato de todas sentenças do texto (FST-TEXT), onde em cada senteça os token estão associado às suas etiquetas com as informações contidas nos dicionários. O exemplo da Figura 3.13 corresponde ao autômato da sentença: *"Nós chegamos a São Paulo"*.

Depois de realizados o pré-processamento e a análise lexical, o usuário terá acesso a uma série de estatísticas do texto (como número de sentenças, número de *tokens*, número de palavras simples, número de palavras compostas, etc.), acesso aos dicionários de palavras contidas no texto (dicionário de palavras simples, compostas e de palavras desconhecidas), acesso à lista de *tokens* que pode ser ordenada por freqüência no texto ou alfabeticamente,

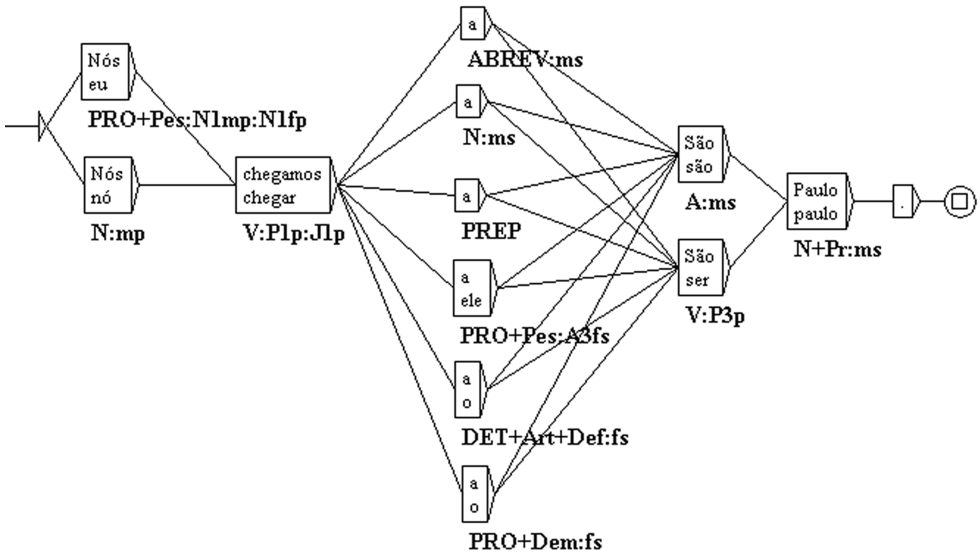

Figura 3.13: Representação do autômato de uma sentença.

e acesso ao autômato do texto.

O FST-TEXT é um tipo especial de grafo no Unitex. Ele é criado automaticamente (ligando a opção *Create text automaton* na fase de pré-processamento ou *Construct FST-TEXT.*. no menu *Text*). Existe um grafo FST para cada sentença de entrada. Esses grafos representam todas análises possíveis da sentença, levando em consideração os dicionário de palavras flexionadas.

O FST-TEXT é um excelente ponto de partida para remover as ambigüidades das sentenças de forma manual, pois o usuário somente tem que remover as etiquetas que não são adequadas à sentença.

Outra saída neste ponto é o usuário aplicar as regras ELAG para remover de forma automática as ambigüidades identificadas pelas regras. O resultado pode ser visualizado habilitando a opção *ELAG Frame*.

3.3.2 Identificação de padrões

Após ser efetuado o pré-processamento, é possível então localizar padrões morfossintáticos num córpus através de expressões regulares ou grafos. Todas expressões regulares podem ser representadas por grafos. Tais padrões podem ser:

- Uma dada palavra ou uma lista de palavras. Por exemplo, pode-se localizar em um texto todas as ocorrências da flexão do verbo *cantar* conjugado no futuro, ou todas as ocorrências de palavras compostas.
- Uma dada categoria gramatical, como todos verbos conjugados na terceira pessoa do singular (*V:3s*) ou os nomes femininos no plural (*N:fp*). Seguem, abaixo, alguns exemplos de categorias (quaisquer códigos gramaticais propostos pelo usuário criador do dicionário podem ser usados).

A:p	Adjetivo (*A*) no plural (*p*)
ADV	Advérbio
PRO	Pronome
DET:f	Determinante (*DET*) no feminno (*f*)
V+t:ms	Verbo (*V*) transitivo (*t*) no masculino singular (*ms*)

- Uma expressão regular ou um grafo. O grafo da Figura 3.14 representa a expressão regular: *(<dever>+<poder>) (<ADV> + <E>) <V:W>*.

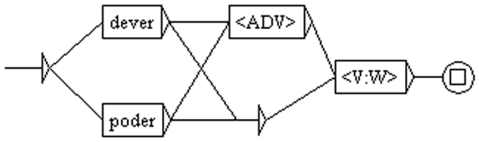

Figura 3.14: Exemplo de grafo para identificar padrão.

Este padrão reconhece qualquer seqüência começando com o verbo *dever* ou *poder*, seguido de um advérbio opcional (<*E*> significa uma palavra vazia) e um verbo na forma infinitiva (<*V:W*>). Note que as categorias são reconhecidas tanto para palavras simples quanto para palavras compostas. Um exemplo de seqüência reconhecida por esse padrão está na Figura 3.15.

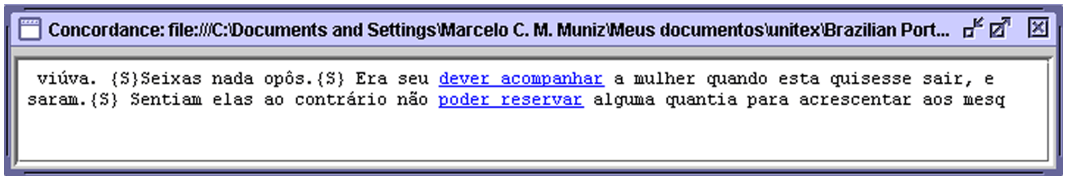

Figura 3.15: Seqüência reconhecida pelo grafo da Figura 3.14.

- Conjunto de expressões de sinônimos. Grafos de diferentes línguas podem ser ligados, para que cada seqüência reconhecida numa língua fonte seja automaticamente associada a um grafo correspondente na língua alvo. Um grafo pode representar todas as expressões que designam entidades ou um processo. Indexando estes grafos (ao invés de meras palavras) pode-se recuperar informações em córpus grandes com alta precisão.

- Gramáticas locais de uma língua. O Unitex inclui um editor de grafos (veja Figura 3.16), o qual pode ser utilizado para edição de gramáticas locais. Operações padrões em grafos permitem ao usuário construir facilmente sistemas com centenas de grafos.

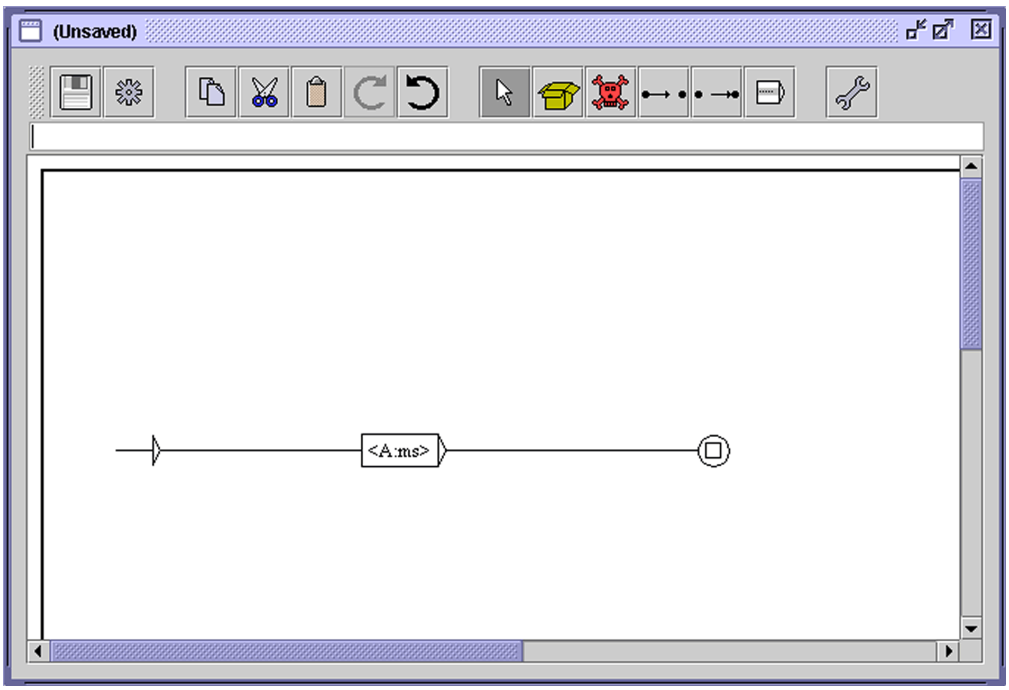

Figura 3.16: Editor de grafos no Unitex.

3.3.3 Resolução de ambigüidade

A ferramenta mais eficiente para remoção de ambigüidades lexicais no Unitex é a utilização de regras ELAG conforme definido na Seção 3.2. Porém, pode ser utilizada outra estratégia. Como o DELAF e o DELACF são dicionários que têm, em geral, uma grande cobertura, ambos contêm palavras cuja classificação morfossintática pode ocorrer somente em domínios específicos. Como conseqüência, tais palavras podem ser analisadas de forma imprópria, ou seja, quando elas têm mais de uma classificação morfossintática. Uma saída para resolver ambigüidade é utilizar dicionários filtrados, isto é, quando o usuário sabe que em um dado córpus, uma entrada ambígua do dicionário só pode ter uma classificação morfossintática, ele remove do dicionário as outras entradas.

Muitas das palavras compostas podem ser ambíguas, pois elas podem ser analisadas como seqüências de palavras simples, entretanto, algumas palavras compostas não são ambíguas, ou porque elas contêm constituintes não autônomos ou porque são termos técnicos. Inserindo estas palavras compostas não ambíguas num dicionário filtrado, o usuário previne o Unitex de procurar em dicionários por palavras simples, uma vez que o sistema não mais processa essas palavras compostas como ambíguas. Dicionários filtrados no sistema Unitex devem ter nomes terminado um - (sinal de menos) antes da extensão "*.dic*", como "*filtro-.dic*".

3.3.4 Etiquetação de palavras ou expressões

O Unitex, além de pesquisar um texto por padrões, pode ser utilizado para inserir informações em textos. O usuário pode adicionar informações aos grafos de busca de padrões que, ao serem reconhecidos, adicionam ao texto as informações contidas no grafo. Esses grafos especiais são chamados de *transdutores* e podem tanto ser utilizados para inserir informações (etiquetar palavras ou expressões) quanto para substituir informações, isto é, quando uma expressão for reconhecida, ela pode ser substituída pela informação contida no grafo.

Um exemplo de transdutor pode ser observado na Figura 3.17, onde o padrão reconhecido são *nomes* no *masculino singular*. Caso esse transdutor seja utilizado para inserir informação em um texto (modo *Merge*), toda vez que tal padrão for encontrado, será adicionado antes do padrão a etiqueta *{Substantivo}*. Caso seja utilizado para substituir informação em um texto (modo *Replace*), tada vez que o *transducer* encontrar esse padrão, ele será substituído no texto pela etiqueta *{Substantivo}*.

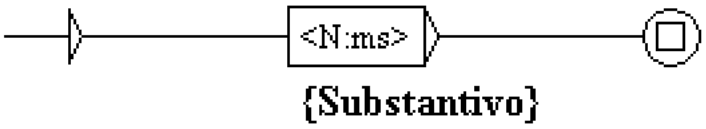

Figura 3.17: Exemplo de um transdutor.

3.4 Considerações Finais

Neste capítulo foi apresentada a ferramenta de análise de córpus Unitex, assim como os formalismo utilizados nesta ferramenta. Os recursos desenvolvidos para este projeto seguiram os formalismos aqui apresentados e serão descritos no próximo capítulo.

CAPÍTULO 4

Unitex-PB

O objetivo deste trabalho é construir recursos lingüísticos computacionais para português brasileiro para ferramenta Unitex. Para atingir esse objetivo foram utilizadas informações disponíveis em recursos desenvolvidos pelo NILC.

O processo de construção destes recursos foi dividido em 4 etapas, a saber:

1. Levantamento de requisitos
2. Modelagem e implementação do DELAS e DELAF para português brasileiro
3. Modelagem e implementação do DELACF para português brasileiro
4. Desenvolvimento de uma biblioteca de acesso e manipulação do léxico Unitex-PB
5. Construção de regras de remoção de ambigüidade lexicais para português brasileiro

As etapas descritas acima, bem como a metodologia utilizada para execução de cada uma delas, serão detalhadas nas seções seguintes.

4.1 Levantamento de requisitos

O primeiro passo realizado nesse projeto foi o levantamento de requisitos para um bom léxico computacional. Foram pesquisadas quais informações deveriam estar contidas no léxico a partir de necessidades de aplicações de PLN e de experiências prévias reportadas no NILC e pela literatura. Este estudo foi realizado em conjunto com especialistas em lingüística. Como o nosso

objetivo é construir dicionários compatíveis com o sistema Unitex, foi também pesquisado o formalismo de dicionários utilizados no Unitex, o formalismo DELA, e léxicos já existentes para português neste formalismo.

Não foi encontrado nenhum estudo de dicionário para português brasileiro seguindo este formalismo na literatura, mas sim para português de Portugal (Ranchhod, Mota, & Baptista 1999). O dicionário de português de Portugal faz parte da distribuição padrão da ferramenta Unitex.

O levantamento de requisitos para os dicionários do Unitex-PB foi elaborado juntamente com lingüistas do NILC, e optou-se pela utilização das mesmas informações contidas no léxico do ReGra para o léxico de Unitex-PB e pela utilização das notações de classes gramaticais e flexionais do dicionário do português de Portugal.

4.2 Modelagem e implementação do DELAS e DELAF para português brasileiro

Atualmente a base de dados lexicais do NILC, a Diadorim, possui aproximadamente 1.500.000 entradas lexicais e ela foi a nossa maior fonte de informação neste projeto. A Diadorim é um banco de dados relacional que centraliza todas informações lexicais do NILC.

Além dessas informações no banco de dados, existe uma versão desses dados em arquivo texto que é utilizado para a criação do léxico do ReGra. Como as duas fontes de informações possuem os mesmos dados, com exceção de informações de sinonímia e antonímia, que somente a Diadorim possui, a versão desses dados em arquivo texto foi utilizada por ser de mais fácil manipulação e porque não iríamos utilizar as informações de sinonímia e antonímia.

O processo de criação do dicionário de palavras simples seguiu o fluxograma apresentado na Figura 4.1.

A primeira etapa deste processo foi a conversão do léxico do ReGra para o formato DELAF utilizando as notações do dicionário de Portugal.

A estrutura do léxico do ReGra é composta de entradas constituídas por uma palavra ou, no máximo, palavras compostas hifenizadas. O exemplo abaixo ilustra duas entradas da palavra "mata", uma para cada canônica possível: [matar], verbo, e [mata], substantivo. Pode-se notar que, além das informações morfológicas, cada verbete traz informações sobre sua(s) classe(s) gramatical(is)[1].

```
mata=<V.[][PRES.ELE.]N.[][matar]0.#S.F.SI.N.[]?.3.[mata]0.>
```

[1]Leia-se V: verbo; PRES: presente do indicativo; ELE: 3^a pessoa; N: colocação pronomial nenhuma; S: substantivo; F: feminino; SI: singular; N: grau nulo

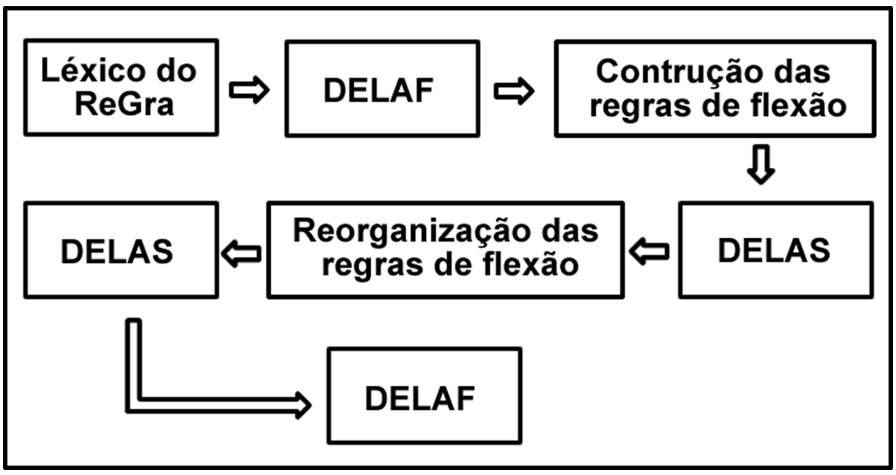

Figura 4.1: Processo de criação do dicionário de palavras simples.

A partir do léxico do ReGra, foi proposto um modelo de conversão para o formato DELAF. Foram projetados quais campos e códigos seriam utilizados para cada classe gramatical. O Apêndice A apresenta com detalhes os campos e códigos do léxico Unitex-PB. Em seguida foi implementado um filtro para a conversão desses dados.

O filtro tratou de 14 classes gramaticais (substantivo, adjetivo, artigo, preposição, conjunção, numeral, pronome, nomes próprios, verbo, advérbio, prefixos, siglas, abreviaturas e interjeição), as mesmas que são suportadas pelo léxico do ReGra.

Esse filtro gerou um dicionário com 1.542.563 entradas flexionadas, que chamaremos de DELAF intermediário, estando presentes nesse dicionário verbos flexionados com ênclise e mesóclise, nomes próprios e algumas palavras compostas separadas por hífen.

O passo seguinte foi remover as palavras compostas e flexões de verbos com ênclise e mesóclise do DELAF intermediário, pois nessa etapa estávamos trabalhando apenas com palavras simples. Com isso o número de entradas caiu para 454.304. Como o objetivo do trabalho era criar um DELAF a partir de um DELAS, seguimos então para a criação das regras de flexão. Vale relembrar que o DELAS é um dicionário formado por canônicas e as regras de flexão associadas a elas.

Para facilitar a criação das regras de flexão, o DELAF intermediário foi dividido em arquivos, cada um com uma classe gramatical. Como as palavras das classes advérbio, conjunção, interjeição, prefixos e siglas não flexionam, não foi necessário criar suas regras de flexão. Somente foi associado o nome da classe ao lema para gerar o DELAS dessas classes. Apesar das abreviaturas conterem informações flexionais de gênero e número, foi decidido que não seria gerado um DELAS para essa classe e que esse dicionário seria distribuído separadamente. O grande desafio foi criar as regras de flexão para as

outras classes gramaticais que representavam 99.99% do total de entradas do dicionário.

Os artigos, numerais, preposições e pronomes foram verificados por lingüistas, comparados com o léxico de Portugal para possível inserção de novas entradas e as regras de flexão, e o DELAS correspondente foi criado manualmente, por se tratarem de poucas entradas.

Para criar as regras de flexão e o DELAS dos substantivos e adjetivos foi adotada uma outra estratégia. Os dicionários dos substantivos e adjetivos foram filtrados para permanecerem somente as canônicas. Então os dicionários com as canônicas foram divididos em arquivos separados pela terminação das canônicas, por exemplo, um arquivo com as canônicas dos substantivos terminados em "*o*", outro para os terminados em "*a*", etc. A estratégia foi restringir nosso problema que era descobrir todas as regras de flexão para os substantivos e adjetivos, em pequenos problemas como descobrir as regras de flexão para os substantivos terminados em "*o*".

Em seguida, para cada arquivo com determinada terminação, tentou-se observar de forma empírica qual seria a regra de flexão mais comum associada à canônica. Por exemplo, observou-se que, para os substantivos terminados em "*o*", a regra que se aplicava à maioria das palavras era a *N001* (veja Figura 4.2), que possui dois caminhos. Em um, não é acrescentado nada à forma canônica e é adicionado o código flexional *:ms*, e no outro é acrescentado o sufixo *s* e o código flexional *:mp*. Essa regra gerada manualmente foi então associada de maneira automática a todas as canônicas do arquivo que estava sendo analisado, originando logo um arquivo no formato DELAS.

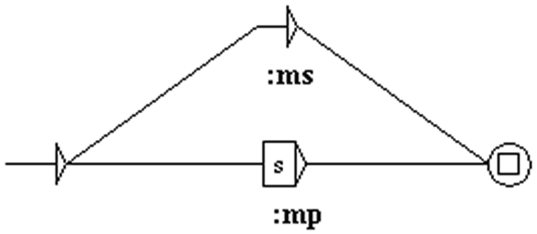

Figura 4.2: Grafo com a regra de flexão *N001*.

Esse arquivo no formato DELAS foi então flexionado de maneira automática e comparado, também de maneira automática, com o arquivo de palavras flexionadas do DELAF intermediário daquela classe e com as canônicas com a mesma terminação. Como resultado dessa comparação, gerou-se um arquivo com os erros, isto é, palavras para as quais a regra de flexão associada não é a correta. Essas foram analisadas manualmente e regras corretas foram criadas e associadas manualmente às canônicas.

Caso a quantidade de erros fosse expressiva, tentava-se repetir o processo

de descobrir mais uma regra geral de flexão para essas palavras erradas, associar essa regra às canônicas, gerar o dicionário de palavras flexionadas e comparar o dicionário respectivo ao DELAF intermediário, até que todas as palavras estivessem associadas com a respectiva regra correta.

Nesse processo de criação do DELAS para os substantivos e adjetivos, primeiramente foram geradas as regras de flexão de gênero e número, e numa segunda fase, foram acrescentadas as regras de flexão de grau. Foram encontrados aproximadamente 1.000 erros no dicionário do ReGra (flexões erradas, omissão de alguma forma flexionada, canônicas erradas) e foram consumidos aproximadamente 6 meses de trabalho.

Para gerar o DELAS dos verbos, não foram utilizados os verbos dicionarizados pelo léxico do Regra (6.672 verbos), mas optou-se pelo emprego dos verbos dicionarizados pelo pesquisador Oto Vale (14.284 verbos), verbos estes que já estavam associados às regras de flexão, e as regras já estavam formalizadas, o que facilitou sua conversão para o padrão DELA. Foi desenvolvido um programa que converteu de maneira automática as regras do formato criado pelo pesquisador Oto Vale para o formato DELA. Mais detalhes sobre este programa podem ser vistos no Apêndice B.

Depois de criadas as regras para todas as classes gramaticais, chegou-se à primeira versão do DELAS para português brasileiro. Porém, as regras de flexão foram nomeadas conforme foram criadas e como muitas regras para substantivos eram as mesmas para adjetivos, mas estavam com nomes diferentes, decidiu-se por reorganizar as regras para os substantivos e adjetivos.

Foram criadas 792 regras de flexão (veja Tabela 4.1) para se flexionar todas as palavras do português brasileiro. Destas, 242 são para adjetivos e 378 para substantivos. Depois de criadas, as regras para substantivos e adjetivos foram reorganizadas por gênero e número, e grau para aumentativos, diminutivos e superlativos (veja Tabela 4.2). Todas as regras dos substantivos e adjetivos foram então renomeadas adotando o seguinte padrão:

[*gênero e número*] [*diminutivo*] [*superlativo*] [*aumentativo*]

Assim, a regra *A252D144A71* corresponde à regra de flexão de gênero e número *A252* juntamente com a regra de flexão em grau no diminutivo *D144* e no aumentativo *A71*. A regra *N252D144A71* corresponde à mesma regra porém para os substantivos. Nessa etapa, os nomes dos arquivos dos grafos com as regras foram renomeados, assim como foram trocadas as regras associadas às canônicas no arquivo DELAS para substantivos e adjetivos.

Com o DELAS pronto, pode-se então gerar o DELAF final. O DELAF final foi compactado utilizando a ferramenta Unitex e passou-se para a fase de testes em córpus, também na ferramenta Unitex.

Descobriu-se que muitos substantivos que poderiam ser classificados tam-

Tabela 4.1: Quantidade de regras de flexão.

Classe Gramatical	**Número de Regras de Flexões**
Substantivo	378
Adjetivo	242
Artigo e Numeral	14
Preposição	17 (incluindo as regras para contrações)
Conjunção	1
Pronome	35 (incluindo as regras para contrações)
Verbo	102
Advérbio	1
Prefixo	1
Abreviaturas	1
Interjeição	1

Tabela 4.2: Quantidade de regras de flexão para substantivos e adjetivos.

Tipos de Flexões	**Número de Regras de Flexão**
Gênero e número	149
Diminutivo	153
Superlativo	78
Aumentativo	72

bém como adjetivos (por exemplo: dorminhoco, roliço, etc.) não estavam classificados. O mesmo acontecia para adjetivos que também poderiam ser classificados como substantivos (por exemplo: dourado, blindado, etc.). Esse erro existia pois o dicionário do ReGra foi otimizado para ser utilizado pelo revisor gramatical, o que entra em choque com a filosofia dos dicionários no formato DELA, pela qual as entradas devem estar associadas a todas classificações possíveis.

Para corrigir esse erro, todos substantivos do dicionário do português de Portugal foram comparados com o dicionário DELAF final e caso uma palavra estivesse classificada no DELAF somente como adjetivo e não substantivo, ela era então guardada em um arquivo separado. O mesmo foi feito com os adjetivos, verificando se eles estavam somente marcados como substantivos. Essas palavras encontradas foram então dicionarizadas, e dessa maneira, conseguimos aumentar o número de entradas dos substantivos de 49.303 para 66.381, o que corresponde a um aumento de 34% no número de entradas, e aumentar o número de entradas dos adjetivos de 57.420 para 59.309, o que corresponde a um aumento de 3.2 %. Os números finais de entradas do dicionário de palavras simples podem ser vistos na Tabela 4.3.

Com as correções feitas no DELAF intermediário (que corresponde ao léxico do ReGra), a utilização dos verbos dicionarizados pelo pesquisador Oto Vale e as correções nas etiquetas dos substantivos e adjetivos, o número de entradas

Tabela 4.3: Estatísticas dos dicionários DELAS e DELAF.

Dicionário de Palavras Simples - DELAS			
Número total de entradas		67.466	
Dicionário de Palavras Flexionadas - DELAF			
Classe	Sigla	Número de entradas	Número de Canônicas
Abreviaturas	ABREV	214	214
Adjetivos	A	59.349	17.658
Advérbios	ADV	2.628	2.628
Artigos	DET+Art	8	2
Conjunções	CONJ	44	44
Interjeições	INTERJ	23	23
Numerais	DET+Num	238	95
Prefixos	PFX	55	55
Preposições	PREP	39	39
	PREPXDET	68	17
	PREPXPREP	1	1
	PREPXPRO	88	33
	PREPXADV	5	5
Pronomes	PRO	228	58
	PROXPRO	38	9
Siglas	SIGL	468	468
Substantivos	N	66.381	27.285
	N+Pr	4.904	4.893
Verbos	V	743.316	14.284
Total		878.095	61.135
Tamanho do dicionário		48.8 MB	
Tamanho do dicionário compactado		0.99 MB	
Taxa de compressão		97.9 %	

do dicionário subiu de 454.304 para 878.095, portanto 93.28%.

4.3 Modelagem e implementação do DELACF para português brasileiro

Durante o desenvolvimento do dicionário de palavras simples, uma das primeiras fases foi a remoção das palavras compostas separadas por hífen que estavam no dicionário do ReGra. Essas palavras foram separadas em um arquivo e serviram de fonte para a construção do dicionário de palavras compostas.

Ao contrário do dicionário de palavras simples, não foi construído para o dicionário de palavras compostas o dicionário correspondente ao DELAS, o DELAC, mas somente o dicionário de palavras compostas flexionadas, o DELACF. Essa decisão foi tomada, pois a ferramenta que converte o DELAC para DELACF e o padrão DELAC ainda estão em fase de padronização pelos pesquisadores do LADL.

Como o padrão DELACF está bem definido, partimos então para a conversão das palavras compostas no padrão do dicionário do ReGra para o padrão DELACF. Foi desenvolvido um filtro para essa conversão.

O DELACF além da informação de classe gramatical a que a palavra composta pertence, necessita da informação da classificação dos constituintes da palavra composta. Por exemplo, para *rabos-de-tatu* os constituintes são um substantivo, uma preposição e outro substantivo. Tentou-se utilizar neste dicionário o mesmo padrão adotado para as palavras compostas do portugês de Portugal. A entrada *rabos-de-tatu* no DELACF seria a seguinte:

rabos-de-tatu,rabo-de-tatu.N+NDN:mp

onde *NDN* representa substantivo + preposição (de) + substantivo. A informação de classificação da classe dos constituintes das palavras compostas não estava presente no dicionário do ReGra, desta forma, foi feita uma classificação manual por lingüistas de todas as palavras compostas. Nesta fase aproveitou-se para verificar se essas palavras realmente eram separadas por hífen e/ou espaço e para verificar se estavam faltando flexões ou se as flexões estavam corretas.

As estatísticas do resultado final do DELACF podem ser observadas na Tabela 4.4.

Tabela 4.4: Estatísticas do DELACF.

DELACF	
Número total de entradas	4.077
Número total de canônicas	2.009
Tamanho do dicionário	301 KB
Tamanho do dicionário compactado	121 KB

4.4 Desenvolvimento de uma biblioteca de acesso e manipulação do léxico Unitex-PB

Depois de criados, os dicionários no formato DELA podem ser compactados. O Unitex possui uma ferramenta de compactação de dicionários baseada em autômatos finitos. Além dessa ferramenta, o Unitex possui funções de acesso e manipulação desses dicionários compactados. O código de acesso e manipulação é interno aos aplicativos da ferramenta Unitex e não existe uma biblioteca de programação específica com essas funcionalidades.

Um dos objetivos desse projeto foi desenvolver uma biblioteca de acesso e manipulação desses dicionários compactados, independente do Unitex, para que qualquer aplicação de PLN possa utilizá-los.

Para desenvolver essa biblioteca, o código fonte do Unitex foi estudado, uma vez que a ferramenta Unitex é distribuída sob a licença GPL e todos podem ver e modificar seu código fonte dentro das condições da licença.

A partir desse estudo foi criada uma biblioteca de programação independente do Unitex, com simples funções de acesso e manipulação a dicionários compactados. Primeiramente foi construída uma biblioteca em ANSI C, que possui três funções: carregar o dicionário na memória, realizar a busca por uma palavra e remover o dicionário da memória. Essa biblioteca implementa uma classe e utiliza 5 classes da ferramenta Unitex.

A ferramenta de compactação de dicionários do Unitex pode ser configurada para indexar as palavras pela forma flexionada ou pela canônica. Caso o dicionário carregado na memória esteja ordenado pelas formas flexionadas, a função desenvolvida na biblioteca que realiza a busca de uma palavra devolve uma String vazia, caso a palavra não seja encontrada, ou uma String com todas as etiquetas da palavra. Caso o dicionário carregado na memória esteja ordenado pelas canônicas, a função que realiza a busca de uma palavra, devolve uma String vazia, caso a canônica não seja encontrada, ou uma String com todas as etiquetas de todas flexões daquela canônica. Por exemplo, se procurarmos pela palavra *casa* em um dicionário ordenado por flexão, a função devolverá:

casa,.N:fs
casa,casar.V:P3s:Y2s

Além da biblioteca em ANSI C foram criadas mais duas versões dessa biblioteca: uma versão DLL dessa biblioteca que pode também ser acessada por programas em Java via JNI (*Java Native Interface*), e uma outra versão dessa biblioteca totalmente em Java.

A versão dessa biblioteca em Java, além de ter portado todas classes de C para Java, possui uma classe a mais, chamada *DelaEntry*, com funções para tratar a saída da função de busca por palavra no dicionário. A classe *DelaEntry* tem funções como, por exemplo, acessar o número de etiquetas que a palavra descompactada possui, e ordenar e acessar as etiquetas por índice.

4.5 Construção de regras de remoção de ambigüidade lexical para português brasileiro

A partir da versão 1.2 beta, foi incorporada ao Unitex uma ferramenta que implementa o padrão ELAG. Como visto na seção 3.2, o ELAG é um padrão de resolução de ambigüidades lexicais baseado em gramáticas. Algumas ferramentas desenvolvidas pelo NILC implementam resolução de ambigüidades lexicais, um exemplo é o revisor gramatical ReGra. Com o objetivo de divulgar a tecnologia ELAG, algumas regras de resolução de ambigüidades presentes no ReGra foram convertidas para o padrão ELAG.

Alguns problemas foram encontrados nesse processo de conversão. Algumas regras presentes no revisor eram dependentes da seqüência de análise, isto é, a ordem como as regras eram aplicadas influenciava no resultado final. No ELAG, as restrições e ordem de análise são independentes.

Outro problema encontrado foi a necessidade em algumas regras de análise paralela nas restrições, por exemplo, *se uma palavra estiver marcada como adjetivo E substantivo faça alguma coisa*. O padrão ELAG não implementa o "*e lógico*", ele não consegue verificar se uma palavra está marcada com *n* etiquetas, que caracteriza uma comparação paralela. O formalismo consegue verificar se uma palavra esta marcada com uma ou outra etiqueta, pode até ser que a palavra esteja marcada com as duas, mas se ele encontra a primeira etiqueta, a condição já é satisfeita. Ele não continua a procura por outra etiqueta.

Outra dificuldade foi o modo como as restrições estavam descritas no revisor. As restrições estavam baseadas em palavras sem informações associadas.

A Figura 4.3 é a representação da seguinte regra: "Se antes de *"a"* vier a palavra *"que"* e depois a palavra *"cada"*, então *"a"* é preposição".

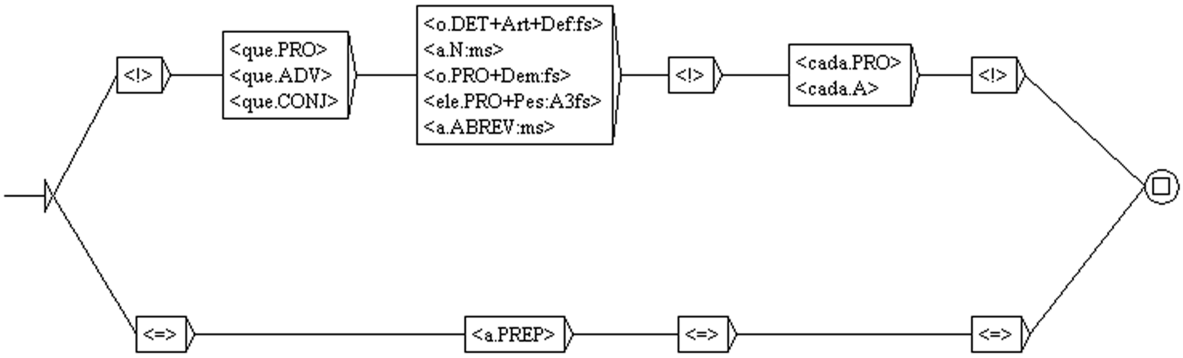

Figura 4.3: Exemplo de regra no padrão ELAG para remover a ambigüidade do *a*.

Outro exemplo de regra ELAG implementada pode ser observado na Figura 4.4. Ele representa a seguinte regra: "Se a palavra *"conta"* é precedida de um substantivo, então *"conta"* é verbo".

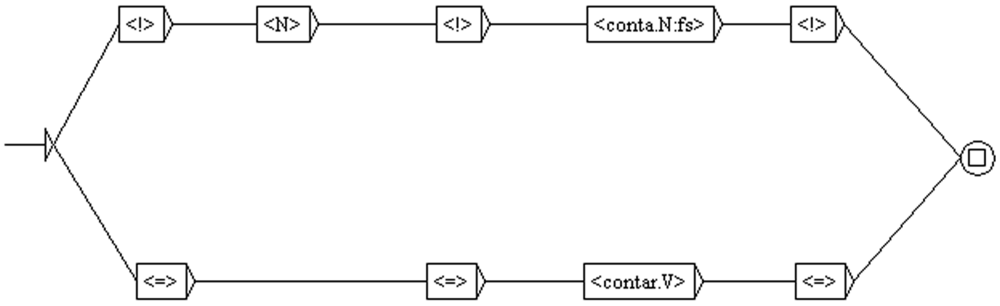

Figura 4.4: Exemplo de regra no padrão ELAG para remover a ambigüidade do verbo *contar*.

Foram desenvolvidas no total 80 regras distribuídas conforme a Tabela 4.5.

Tabela 4.5: Estatísticas das regras ELAG para português brasileiro.

ELAG
16 regras para Adjetivos
30 regras para Advérbios
22 regras para Artigos
12 regras para Substantivos

4.6 Considerações Finais

Neste capítulo foram apresentados os recursos lingüístico-computacionais desenvolvidos para português brasileiro seguindo os padrões utilizados na fer-

ramenta Unitex. O próximo capítulo apresenta os ambientes utilizados para validar esses recursos.

CAPÍTULO 5

Ambientes de acesso

Além dos recursos lingüístico-computacionais desenvolvidos neste projeto, foram utilizados e construídos alguns ambientes para validação desses recursos. Os ambientes são os seguintes:

1. Unitex
2. Interface Web
3. Programa dicionário

Este capítulo tem como objetivo descrever os ambientes utilizados para validação dos recursos do Unitex-PB. Os ambientes serão descritos nas seções seguintes.

5.1 Unitex

Um dos grandes objetivos deste projeto foi construir recursos lingüístico-computacionais para português brasileiro que pudessem ser utilizados e distribuídos pela ferramenta Unitex. Mas para que os dicionários no formato DELA e as regras ELAG pudessem ser utilizados no Unitex, alguns outros recursos necessitaram ser construídos.

Na distribuição padrão do Unitex, recursos para várias línguas estão presentes e esses recursos incluem, além dos dicionários de cada língua, um córpus de referência, um autômato de identificação de fronteira de sentença e, para algumas línguas, regras ELAG.

Como córpus de referencia, foi escolhido o romance *Senhora* de José de Alencar, que é distribuído livremente pelo projeto Lácio Web[1] desenvolvido no

[1]Veja `http://www.nilc.icmc.usp.br/lacioweb/`

NILC e que possui 5.653 sentenças e 176.423 tokens.

Para o português brasileiro, foi utilizado o mesmo autômato de identificação de fronteira de sentença do português de Portugal, mas somente com uma modificação. O autômato foi modificado para reconhecer sentenças de diálogos que começam com hífen, algo que o autômato de Portugal não faz. Foram verificadas abreviaturas para formas de tratamento, endereços, números e tipografia, mas não foi necessário modificar as abreviaturas já inclusas no autômato. A Figura 5.1 representa o autômato de reconhecimento de sentença para o português brasileiro.

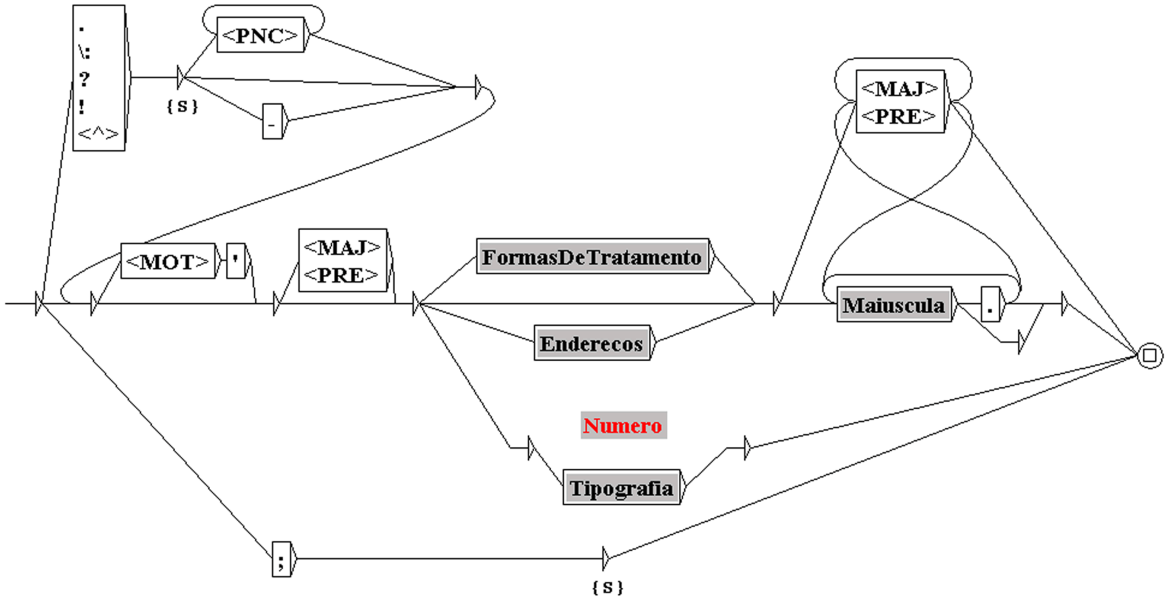

Figura 5.1: Autômato de reconhecimento de sentença para o português brasileiro.

Além do córpus e do autômato de identificação de fronteira de sentença, foi necessário construir um arquivo de definição de língua do dicionário (arquivo com extensão ".lang"), que formaliza todos os códigos utilizados no dicionário de palavras simples do português brasileiro. Esse arquivo de definição de língua é necessário para que se possa compilar e aplicar as regras ELAG. No caso do português brasileiro, foi somente utilizada a definição do dicionário de palavras simples, pois o dicionário de palavras compostas possui um número muito pequeno de palavras e necessita no futuro ser ampliado e remodelado.

Com esses recursos construídos, o Unitex é capaz de analisar qualquer córpus em português brasileiro. Todos esses recursos estarão disponíveis para *download* na próxima distribuição do Unitex e no próprio *site* do projeto Unitex-PB que será apresentado na seção 5.2. A Figura 5.2 exemplifica um córpus em português brasileiro sendo utilizado na ferramenta Unitex. Na janela à direita está o texto, já com as sentenças identificadas e com algumas estatísticas na parte superior da janela, e na janela à esquerda aparece as lis-

tas de palavras (palavras simples, compostas e não identificadas) encontradas no texto.

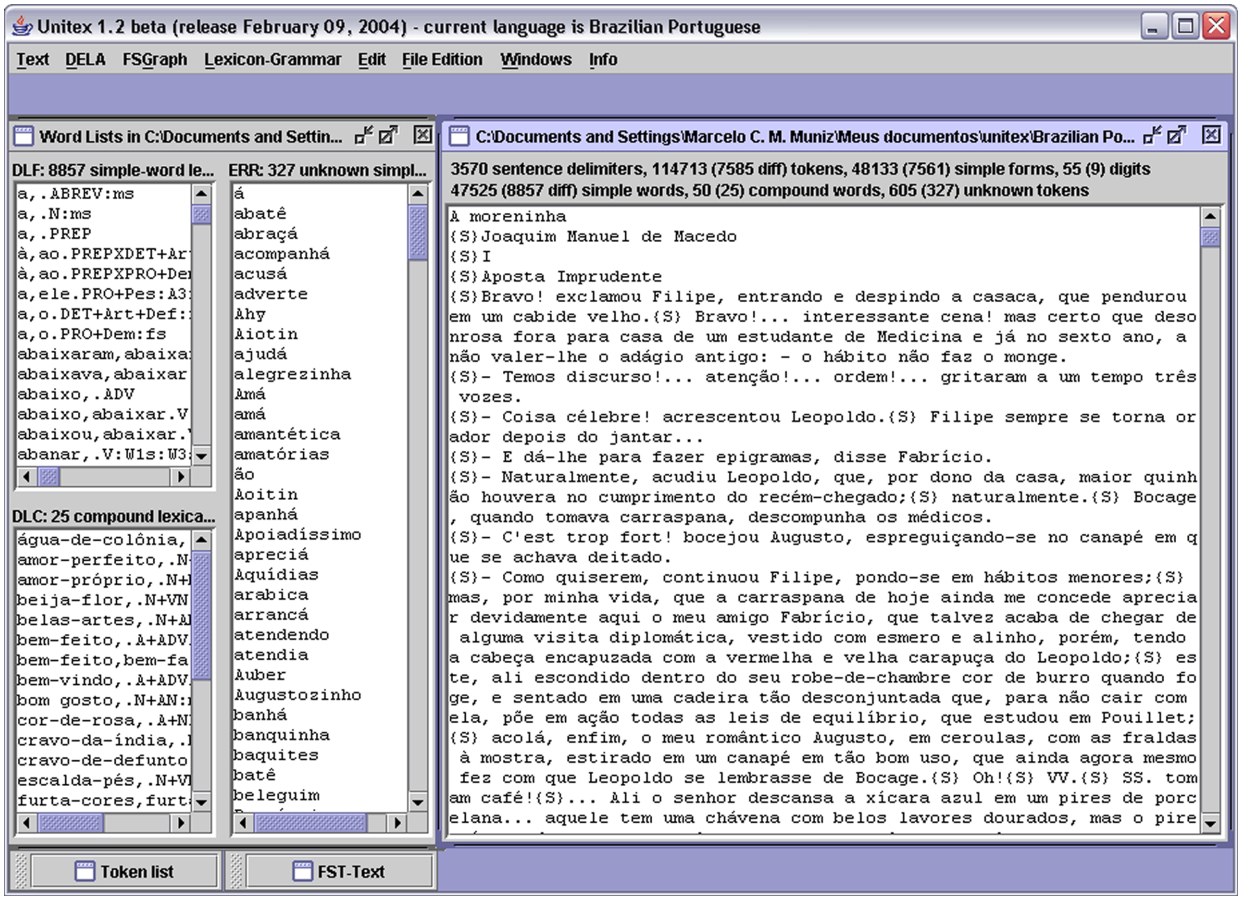

Figura 5.2: Ferramenta Unitex utilizando os recursos do português brasileiro.

5.2 Interface Web

Com o intuito de permitir que qualquer usuário, leigo ou especialista, tenha acesso aos resultados deste trabalho, foi desenvolvido um site para este projeto. O site está disponibilizado no endereço:

http://www.nilc.icmc.usp.br:8180/unitex-pb/

A interface da página principal do site é apresentada na Figura 5.3.

Esse site disponibiliza aos usuários os recursos lingüístico-computacionais desenvolvidos neste projeto e, além disso, implementa duas aplicações Web que utilizam tanto o dicionário de palavras simples DELAF quanto a biblioteca de acesso e manipulação dos dicionários. Essas aplicações são:

1. Busca no dicionário
2. Anotador Morfossintático

Figura 5.3: Página principal do site do projeto.

Essas aplicações foram baseadas num servidor Web Apache/Jakarta[2], que permitem rodar aplicações Web escritas em linguagem de programação Java chamadas de *servlets*. Ambas as aplicações utilizaram a versão totalmente em Java da biblioteca de acesso e manipulação dos dicionários desenvolvida neste projeto. Além disso, foi utilizado para o desenvolvimento dessas aplicações o framework de desenvolvimento de *servlets* Barracuda[3], que nos permite criar poderosas aplicações Web deixando o código da aplicação independente da interface gráfica.

A aplicação Busca no Dicionário oferece ao usuário consulta ao dicionário DELAF. As Figuras 5.4 e 5.5 ilustram essa consulta.

A aplicação Anotador Morfossintático dá a opção ao usuário de anotar textos com as informações presentes no dicionário DELAF. O usuário pode entrar com textos via formulário ou *upload* de arquivo e também tem a opção de escolher quais anotações ele quer: canônica, categoria gramatical, subcategoria gramatical e/ou atributos semânticos e atributos morfológicos. O resultado pode ser visualizado em uma página HTML, em um arquivo TXT ou arquivo TXT compactado. Esse Anotador Morfossintático é de certa forma semelhante

[2]Veja `http://jakarta.apache.org/`
[3]Veja `http://barracudamvc.org/Barracuda/index.html`

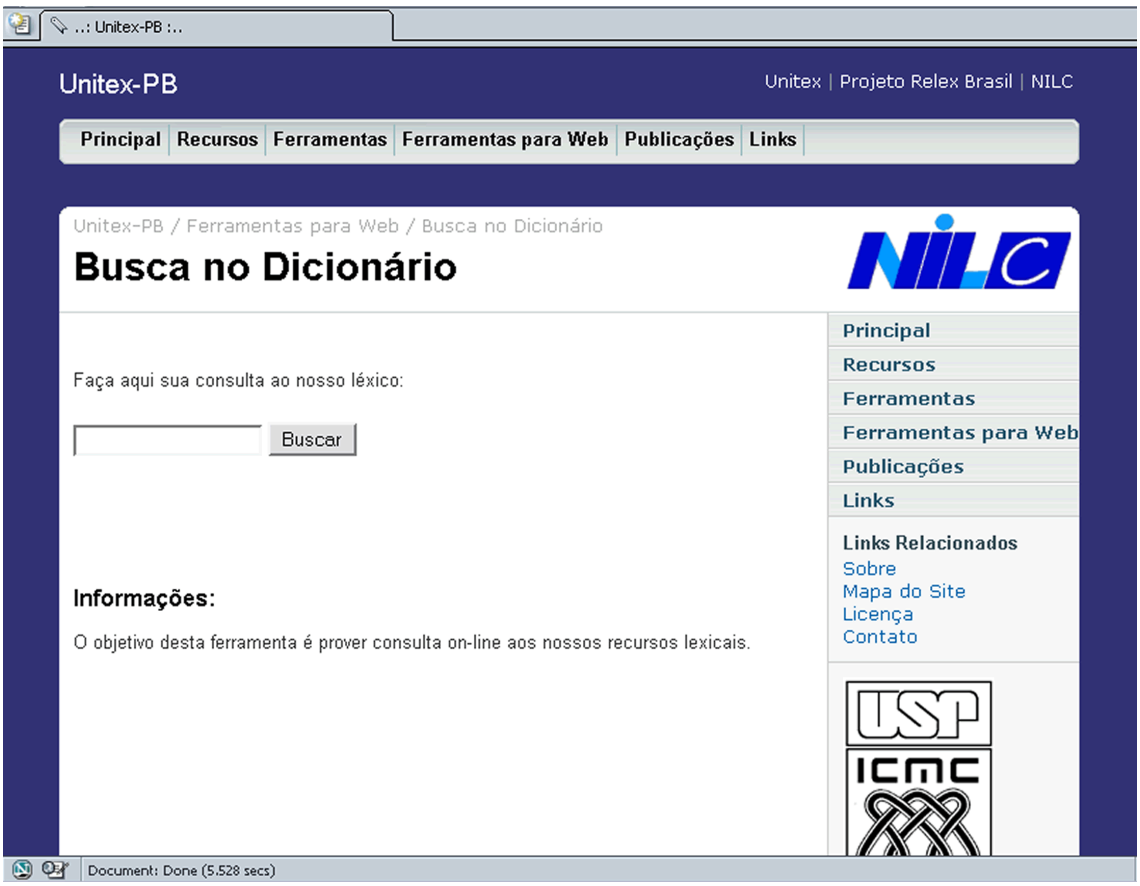

Figura 5.4: Página da aplicação Busca no Dicionário.

à ferramenta construída pelos pesquisadores do LabEL de Portugal, chamada ANELL[4] - Anotador Electrónico LabEL, mas utilizando recursos do português brasileiro.

As Figuras 5.6, 5.7, 5.8 e 5.9 ilustram a aplicação Web Anotador Morfossintático.

5.3 Programa dicionário

O último ambiente desenvolvido para validação dos recursos lingüístico-computacionais foi o programa Dicionário. Ele é um programa escrito em Java que acessa a DLL da biblioteca de acesso e manipulação de dicionários desenvolvida neste projeto via JNI. Esse é um simples programa que pode carregar qualquer dicionário no formato DELA compactado e possui uma interface para realizar buscas às palavras no dicionário.

O código deste programa também será disponibilizado como exemplo de como utilizar a biblioteca de acesso e manipulação de léxicos. As interfaces desse programa podem ser observadas na Figura 5.10.

[4]Veja `http://label10.ist.utl.pt/anell/`

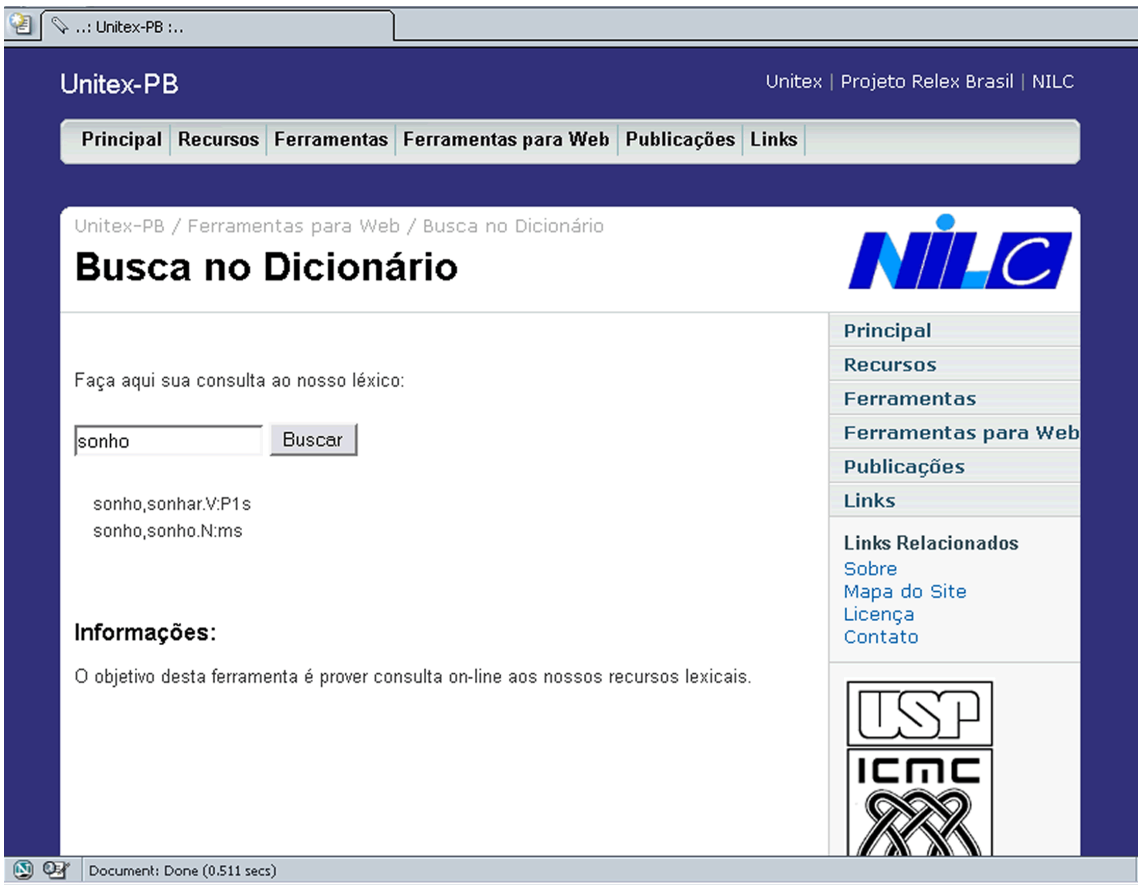

Figura 5.5: Resultado da busca à palavra *sonho*.

Figura 5.6: Página de seleção do modo de enviar o texto a ser anotado.

5.4 Considerações Finais

Neste capítulo foram apresentados alguns ambientes que foram utilizados para validação dos recursos desenvolvidos neste projeto. O próximo capítulo discute as conclusões finais deste trabalho.

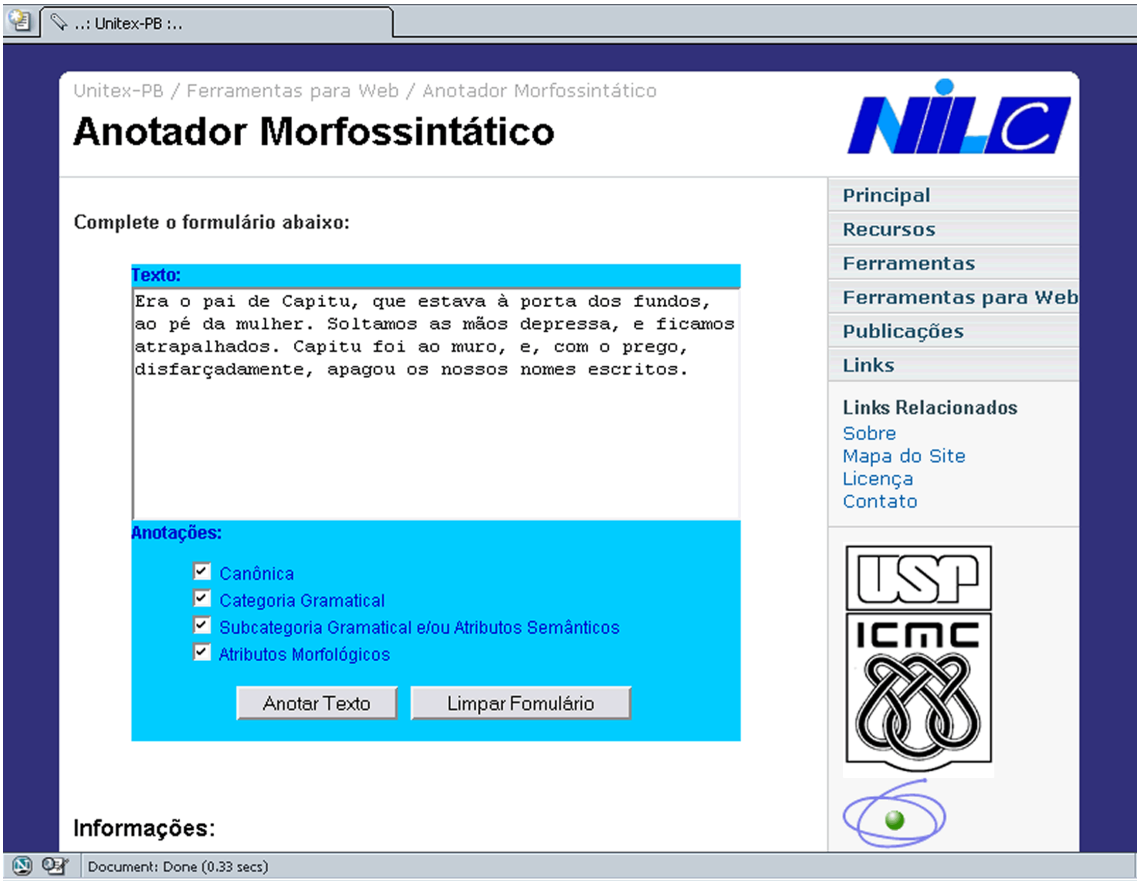

Figura 5.7: Exemplo de como enviar um texto via formulário.

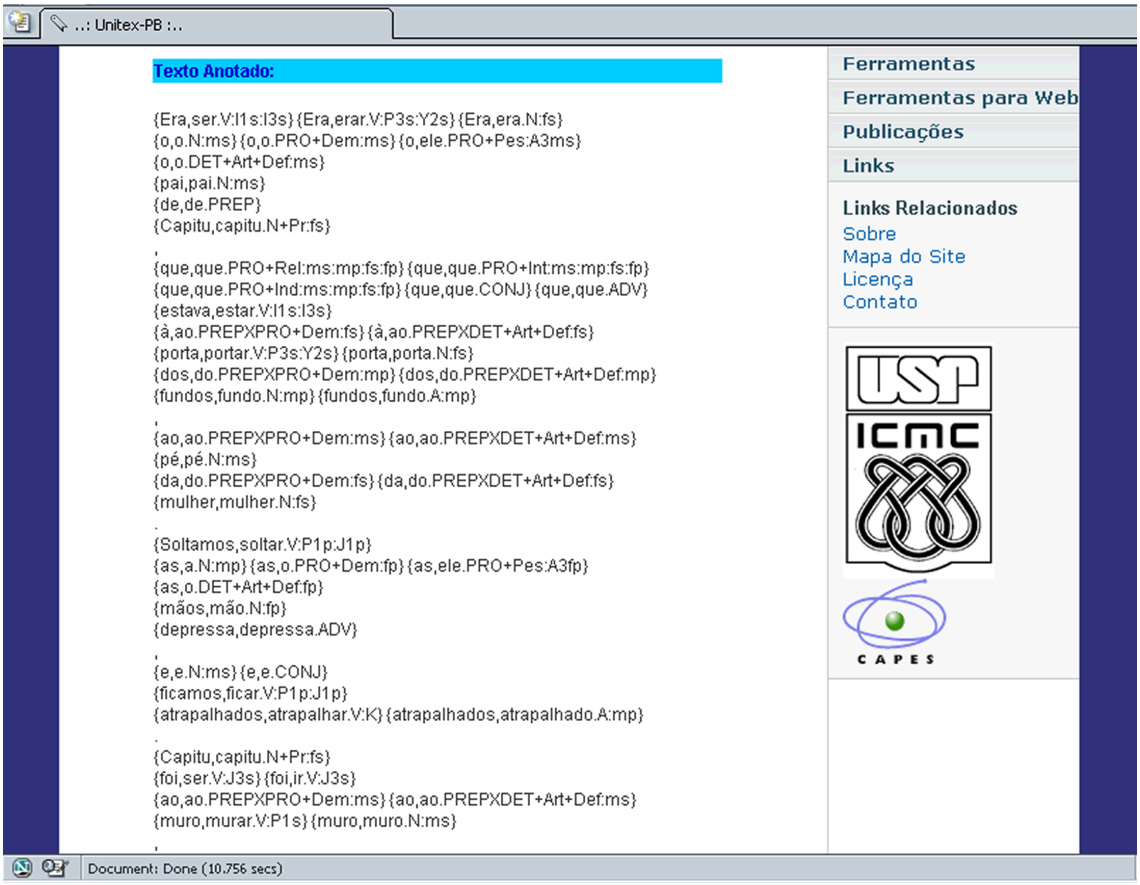

Figura 5.8: Resuldato da anotação em uma página HTML.

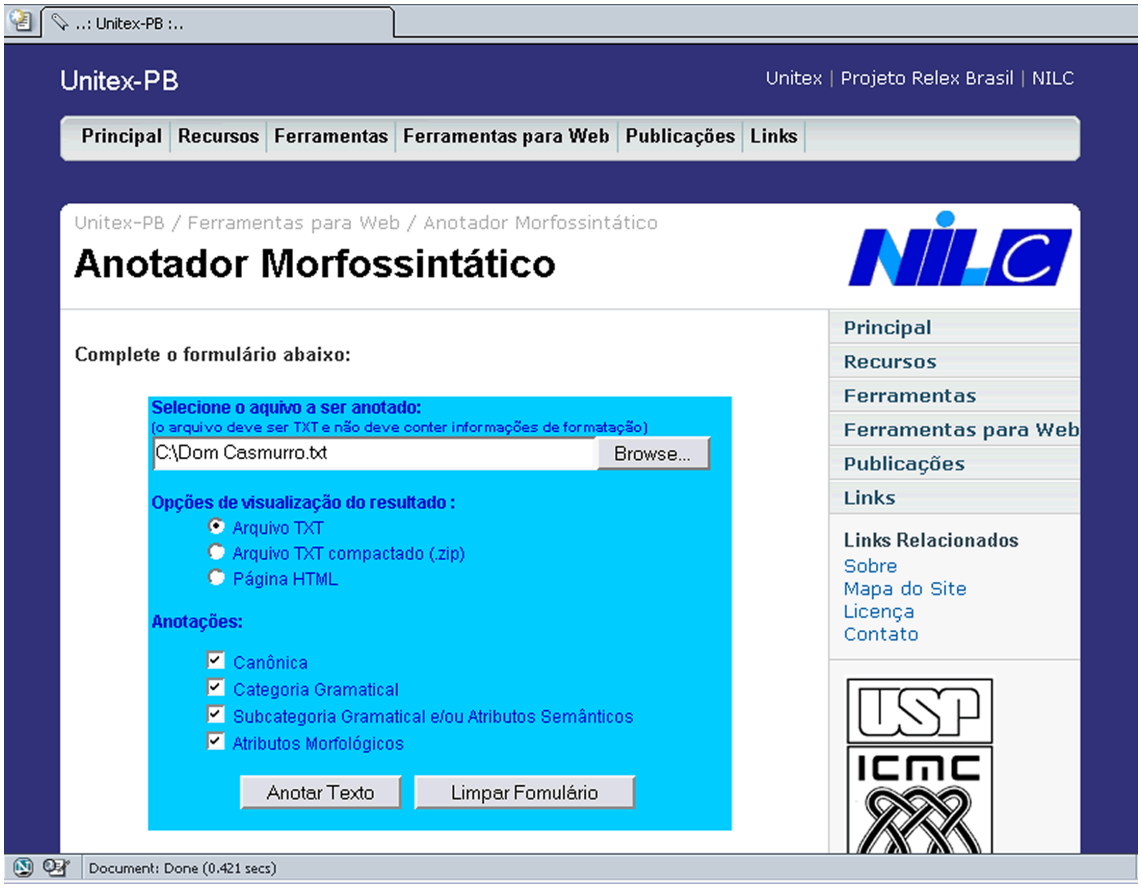

Figura 5.9: Exemplo de como enviar um texto via *upload* de arquivo.

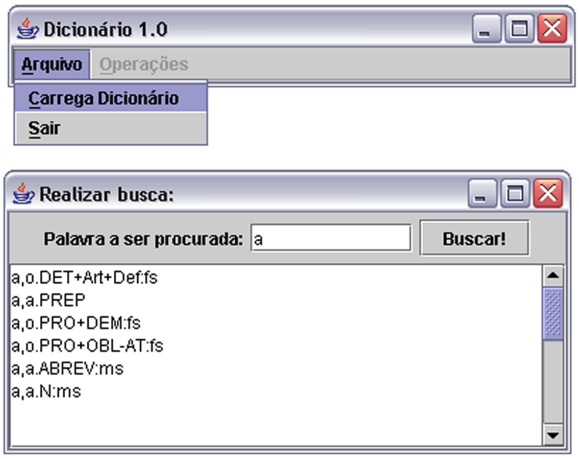

Figura 5.10: Interfaces do programa Dicionário.

CAPÍTULO 6

Conclusões e Contribuições

O objetivo principal deste trabalho - construção de recursos lingüístico-computacionais para o português do Brasil seguindo os formalismos da ferramenta Unitex - foi alcançado com sucesso.

O desenvolvimento dos dicionários foi um trabalho demorado, porém bastente desafiador. Conseguimos gerar um dicionário de palavras simples no padrão DELA para português brasileiro com um número 93.28% maior de entradas que nossa fonte original de dados. Durante o processo de construção desse dicionário foram contruídas e documentadas todas regras de flexão utilizadas para gerar o DELAF, recurso este até então inexistente para o português brasileiro.

O número de entradas do dicionário de palavras compostas, assim como o número de regras de remoção de ambiguidades, ainda é bastante tímido, porém ficou registrado neste trabalho o modo de como construí-los e foram divulgadas e documentadas essas novas tecnologias que ainda não tinham sido utilizadas para o português brasileiro.

As bibliotecas desenvolvidas em ANSI C e Java para acesso a léxicos compactados foram testadas na aplicação Dicionário e nas ferramentas para Web (Busca no dicionário e Anotador Morfossintático) e, juntamente com o recurso de compactação de léxicos presentes no Unitex mostram-se como um recurso eficiente e alternativo às tecnologias proprietárias utilizadas para compactar dicionários utilizando autômatos finitos.

Para o desenvolvimento desses recursos, a colaboração entre lingüistas e cientistas de computação foi essencial, uma vez que seria muito difícil, sem o apoio de um especialista, corrigir e gerar regras de flexão para os dicionários e desenvolver regras de remoção de ambigüidades lexicais.

Os resultados deste trabalho ajudam a consolidar o NILC como um membro da rede Relex.

Finalmente, acreditamos que a disponibilização desses recusos na Web contribui para um maior acesso tanto de usuários comuns como de pesquisadores da área PLN a recursos lingüísticos do português do Brasil, ainda muito escassos e, consequentemente, para o crescimento das pesquisas em PLN no Brasil.

APÊNDICE

A

Formato dos Dicionários Unitex-PB

Este apêndice tem por objetivo apresentar os campos e códigos utilizados nos dicionários que compõem o Unitex-PB.

A.1 Estrutura das entradas:

`Palavra,canônica.Classe+traços:flexão`

Cada verbete poderá estar classificado em mais de uma classe gramatical, neste caso haverá uma entrada para cada classe.

A.2 As categorias (classes) básicas do verbete são:

A.2.1 Substantivo

Classe: `N`

Gênero:

`m`: masculino

`f`: feminino

Número:

`s`: singular

`p`: plural

Grau:

`A`: aumentativo

`D`: diminutivo

`(podendo ser nulo)`

Estrutura:

`Entrada,canônica.N:grau gênero número`

Exemplos:

menino: `menino,menino.N:ms`
meninos: `meninos,menino.N:mp`
meninão: `meninão,menino.N:Ams`
lápis: `lápis,lápis.N:ms:mp`
ajuda: `ajuda,ajuda.N:fs`

A.2.2 Adjetivo

Classe: `A`
Gênero:

`m`: masculino
`f`: feminino

Número:

`s`: singular
`p`: plural

Grau:

`A`: aumentativo
`D`: diminutivo
`S`: superlativo
`(podendo ser nulo)`

Estrutura:

`Entrada,canônica.A:grau gênero número`

Exemplos:

bonito: `bonito,bonito.A:ms`
bonitas: `bonitas,bonito.A:fp`
aprazível: `aprazível,aprazível.A:ms:fs`
simples: `simples,simples.A:ms:mp:fs:fp`
igual: `igual,igual.A:ms:fs`
amabilíssimo: `amabilíssimo,amável.A:Sms`

A.2.3 Artigo

Classe: `DET+Art`
Tipo:

`Def`: Definido
`Ind`: Indefinido

Gênero:

`m`: masculino

`f`: feminino

Número:

`s`: singular

`p`: plural

Estrutura:

`Entrada,canônica.DET+Art+Tipo:gênero número`

Exemplos:

o: `o,o.DET+Art+Def:ms`

umas: `umas,um.DET+Art+Ind:fp`

A.2.4 Preposição

Classe: `PREP`

Estrutura:

`Entrada,canônica.PREP`

Exemplos:

ante: `ante,ante.PREP`

de: `de,de.PREP`

A.2.5 Conjunção

Classe: `CONJ`

Estrutura:

`Entrada,canônica.CONJ`

Exemplos:

mas: `mas,mas.CONJ`

mais: `mais,mais.CONJ`

mal: `mal,mal.CONJ`

A.2.6 Numeral

Classe: `DET+Num`

Tipo:

`C`: cardinal

`O`: ordinal

`M`: multiplicativo

`F`: Fracionário
`L`: Coletivo

Gênero:
`m`: masculino
`f`: feminino

Número:
`s`: singular
`p`: plural

Estrutura:
`Entrada,canônica.DET+Num:tipo gênero número`

Exemplos:
segundo: `segundo,segundo.DET+Num:Oms`
duplo: `duplo,duplo.DET+Num:Mms`

A.2.7 Pronome

Classe: `PRO`

Tipo:
`Dem`: Demonstrativo
`Ind`: Indefinido
`Rel`: Relativo
`Int`: Interrogativo
`Tra`: Tratamento
`Pos`: Possessivo
`Pes`: Pessoal

Forma:
`A`: Forma Acusativa
`D`: Forma Dativa
`N`: Forma Nominativa
`O`: Forma Oblíqua
`R`: Forma Reflexa
`(podendo ser nulo)`

Pessoa:
`1`: Primeira Pessoa
`2`: Segunda Pessoa
`3`: Terceira Pessoa

Gênero:
`m`: masculino
`f`: feminino

Número:

s: singular
p: plural

Estrutura:

`Entrada,canônica.PRO+Tipo:forma pessoa gênero número`

Exemplos:

senhora: `senhora,senhor.PRO+Tra:3fs`
eu: `eu,eu.PRO+Pes:N1ms:N1fs`

A.2.8 Verbo

Classe: `V`

Tempo:

`W`: Infinitivo
`G`: Gerúndio
`K`: Particípio
`P`: Presente do Indicativo
`I`: Pretérito Imperfeito do Indicativo
`J`: Pretérito Perfeito do Indicativo
`F`: Futuro do Presente do Indicativo
`Q`: Pretérito mais que Perfeito do Indicativo
`S`: Presente do Subjuntivo
`T`: Imperfeito do Subjuntivo
`U`: Futuro do Subjuntivo
`Y`: Imperativo
`C`: Futuro do Pretérito

Pessoa:

`1s`: eu
`2s`: tu
`3s`: ele
`1p`: nós
`2p`: vós
`3p`: eles

Estrutura:

`Entrada,canônica.V:tempo pessoa`

Exemplos:

cantaríamos: `cantaríamos,cantar.V:C1p`
cantarias: `cantarias,cantar.V:C2s`
cantaria: `cantaria,cantar.V:C1s:C3s`

A.2.9 Advérbio

Classe: ADV
Estrutura:
Entrada,canônica.ADV

Exemplos:
abaixo: abaixo,abaixo.ADV
misericordiosissimamente:
misericordiosissimamente,misericordiosissimamente.ADV
mesmo: mesmo,mesmo.ADV

A.2.10 Prefixos

Classe: PFX
Estrutura:
Entrada,canônica.PFX

Exemplos:
super: super,super.PFX
pós: pós,pós.PFX
sub: sub,sub.PFX

A.2.11 Siglas

Classe: SIGL
Estrutura:
Entrada,canônica.SIGL

Exemplos:
ONU: ONU,ONU.SIGL
PDT: PDT,PDT.SIGL
OTAN: OTAN,OTAN.SIGL
USP: USP,USP.SIGL

A.2.12 Abreviaturas

Classe: ABREV
Gênero:
m: masculino
f: feminino
Número:

s: singular

p: plural

Estrutura:

`Entrada,canônica.ABREV:gênero número`

Exemplos:

ml: `ml,ml.ABREV:ms`

`mm: mm,mm.ABREV:ms`

A.2.13 Interjeição

Classe: `INTERJ`

Estrutura:

`Entrada,canônica.INTERJ`

Exemplos:

Ah: `Ah,Ah.INTERJ`

Ih: `Ih,Ih.INTERJ`

Olá: `Olá,Olá.INTERJ`

Oi: `Oi,Oi.INTERJ`

APÊNDICE

B

Programa para gerar as flexões dos verbos

O programa que gera os autômatos compilados das regras de flexão dos verbos para o Unitex, funciona da seguinte maneira:

Existem dois arquivos de entrada:

1. `sufixos.txt` - Arquivo com os sufixos.
2. `regras.txt` - Arquivo com os paradigmas das conjugações

O programa gera como resultado arquivos "`.fst2`" (um arquivo para cada paradigma de conjugação), que correspondem às regras associadas a cada verbo, no arquivo DELAS dos verbos.

Com a união dos arquivos "`.fst2`" e o arquivo DELAS dos verbos é possível gerar, de forma automática, o arquivo DELAF dos verbos, que possuem os verbos conjugados em todos os tempos.

No arquivo `regras.txt`, estão explicitados para cada paradigma:

- o número do paradigma
- a forma do verbo padrão do paradigma
- as regras que geram as formas das raízes
- os sufixos

- a configuração dos tempos e pessoas verbais

Assim, num paradigma como:

```
V011 colocar LLLc LLLqu t1
r1 r1 r1
r1 r1 r2(1)r1(23456) r1 r1
r2 r1 r1 r1(25)r2(346) r1
```

Na primeira linha tem-se:

`V011` - o número do paradigma
`Colocar` - o verbo padrão do paradigma (com a ênfase na desinência)
`LLLc` - a regra que gera a forma da raiz `r1`
`LLLqu` - a regra que gera a forma da raiz r2
`t1` - é o conjunto de sufixos *default* a ser buscado no arquivo `sufixo.txt`

Assim temos as seguintes raízes:

Aplicando a regra `LLLc` ao verbo `colocar`, gerará `coloc`, isto é, a regra `LLLc` significa que se deve voltar três letras (*left* três vezes) e acrescentar a letra `c`.
`r1=LLLc`
De forma análoga está `r2` (que gerará a raiz `coloqu`).
`r2=LLLqu`

Nas três linhas seguintes estão os tempos:

```
W G K
P I J F Q
S T U Y C
```

Em que:

`W` = infinitivo
`G` = gerúndio
`K` = particípio
`P` = presente do indicativo
`I` = pretérito imperfeito do indicativo
`J` = pretérito perfeito do indicativo
`F` = futuro do presente do indicativo
`Q` = pretérito mais que perfeito do indicativo

S = presente do subjuntivo
T = imperfeito do subjuntivo
U = futuro do subjuntivo
Y = imperativo
C = futuro do pretérito

As pessoas verbais são numeradas de 1 a 6:

1 - primeira pessoa do singular
2 - segunda pessoa do singular
3 - terceira pessoa do singular
4 - primeira pessoa do plural
5 - segunda pessoa do plural
6 - terceira pessoa do plural

Assim, temos na terceira linha da regra a seguinte configuração para o tempo J (i.e. pretérito perfeito do indicativo):

```
r2(1)r1(23456)
```

Isso significa, que nesse tempo você tem a raiz r2 aplicada à primeira pessoa (o 1 entre parênteses logo após o r2) e a raiz r1 aplicada às demais pessoas (os 23456 entre parênteses logo após o r1) e, que todos os sufixos aplicados são o conjunto de sufixos *default* t1, que foi buscado no arquivo sufixos.txt. Nele encontramos a seguinte linha para o tempo J:

```
ips.t1 = ei, aste, ou, amos, astes, aram,
```

O programa junta a regra da raiz r2 com o primeiro sufixo da lista e adiciona também a informação de tempo e pessoa verbal, gerando a seguinte regra:

```
LLLqu + ei -> LLLquei:J1s
```

E junta a regra da raiz r1 com os demais sufixos da lista:

```
LLLc + aste -> LLLcaste:J2s
LLLc + ou -> LLLcou:J3s
LLLc + amos -> LLLcamos:J1p
LLLc + astes -> LLLcastes:J2p
LLLc + aram -> LLLcaram:J3p
```

Essas regas são então passadas para o formato ".fst2" gerando um arquivo com o nome do paradigma. Neste exemplo, estas regras estariam incluídas no

arquivo "`V011.fst2`". É gerado um arquivo para cada paradigma de conjugação.

Todos os arquivos "`.fst2`" devem então ser colocados no diretório *Inflection* da língua corrente, que esta sendo utilizada no Unitex, para que uma vez carregado o arquivo DELAS dos verbos, possa então ser gerado o arquivo DELAF.

Além dos arquivos "`.fst2`", o programa gera um arquivo texto com a documentação de todos paradigmas de conjugação. As documentações são todas as regras para todos os paradigmas. O nome desse arquivo é "`documentação.txt`". Um exemplo de fragmento desse arquivo é o seguinte:

```
V011
Ex:  colocar

LLLcar:W1s:W3s:U1s:U3s
LLLcares:W2s:U2s
LLLcarmos:W1p:U1p
LLLcardes:W2p:U2p
LLLcarem:W3p:U3p
LLLcando:G
LLLcado:K
LLLcada:K
LLLcados:K
LLLcadas:K
LLLco:P1s
LLLcas:P2s
LLLca:P3s:Y2s
LLLcamos:P1p:J1p
LLLcais:P2p
LLLcam:P3p
LLLcava:I1s:I3s
LLLcavas:I2s
LLLcávamos:I1p
LLLcáveis:I2p
LLLcavam:I3p
LLLquei:J1s
LLLcaste:J2s
LLLcou:J3s
LLLcastes:J2p
```

```
LLLcaram:J3p:Q3p
LLLcarei:F1s
LLLcarás:F2s
LLLcará:F3s
LLLcaremos:F1p
LLLcareis:F2p
LLLcarão:F3p
LLLcara:Q1s:Q3s
LLLcaras:Q2s
LLLcáramos:Q1p
LLLcáreis:Q2p
LLLque:S1s:S3s:Y3s
LLLques:S2s
LLLquemos:S1p:Y1p
LLLqueis:S2p
LLLquem:S3p:Y3p
LLLcasse:T1s:T3s
LLLcasses:T2s
LLLcássemos:T1p
LLLcásseis:T2p
LLLcassem:T3p
LLLcai:Y2p
LLLcaria:C1s:C3s
LLLcarias:C2s
LLLcaríamos:C1p
LLLcaríeis:C2p
LLLcariam:C3p
```

Para pegar um paradigma um pouco mais complexo, vejamos o do verbo `dar`:

```
V017 dar LL LLê LLá t1
r1 r1 r1
r1t4(1236)r1(45) r1 r1(1)r1t2(23456) r1 r1t4
r2t3(1236)r1(45) r1t4 r1t2 r3t4(2)r2t4(36)r1(45) r1
```

temos:

`V017` - o número do paradigma
`dar` - o verbo padrão do paradigma (com a ênfase na desinência)
`LL` - a regra que gera a forma da raiz `r1`
`LLê` - a regra que gera a forma da raiz `r2`
`LLá` - a regra que gera a forma da raiz `r3`

`t1` - é o conjunto de sufixos *default* a ser buscado no arquivo `sufixo.txt`

Portanto, temos:

```
r1 = LL
r2 = LLê
r3 = LLá
```

Peguemos o tempo `S` (presente do subjuntivo). Tem-se aí para as pessoas `1`, `2`, `3` e `6` a raiz `r2` com o conjunto de sufixos `t3`. para as pessoas `4` e `5` tem-se a raiz `r1` com o conjunto de sufixos *default*, ou seja, `t1`.

Portanto, o programa vai buscar no arquivo `sufixos.txt` os conjuntos de sufixos `t1` e `t3` para o tempo `S`.

Lá encontramos as seguintes linhas:

```
spr.t1 = e, es, e, emos, eis, em,
spr.t3 = , s, , mos, is, em,
```

Note-se que para o conjunto `t3` temos a primeira e terceira casas vazias. Isso significa que o sufixo ai é nulo.

Assim, temos:

```
LLê + -> LLê:S1s (r2t3)
LLê + s -> LLês:S2s (r2t3)
LLê + -> LLê:S3s (r2t3)
LL + emos -> LLemos:S1p (r1t1)
LL + eis -> LLeis:S2p (r1t1)
LLê + em -> LLêem:S3p (r2t3)
```

Referências

Anastasiadis-Symeonidis, A., T. Kyriacopoulou, E. Sklavounou, I. Thilikos, & R. Voskaki (2000). A system for analysing texts in modern greek: representing and solving ambiguities. In *Proceedings of COMLEX 2000. Computational Lexicography and Multimedia Dictionaries*. Dept. of Electrical and Computer Engineering, Univ. of Patras, Greece. `http://citeseer.nj.nec.com/457067.html` (16/02/2004).

Atkins, S., N. Bel, F. Bertagna, P. Bouillon, N. Calzolari, C. Fellbaum, R. Grishman, A. Lenci, C. MacLeod, M. Palmer, G. Thurmair, M. Villegas, & A. Zampolli (2002). From resources to applications. designing the multilingual isle lexical entry. In *LREC 2002: Third International Conference on Language Resources and Evaluation*, Volume II, pp. 687–693. Las Palmas, Ilhas Canárias.

Briscoe, T. (1991). Lexical issues in natural language processing. *Natural Language and Speech*, 39–68.

Calzolari, N. (1990). Structure and access in a automated lexicon and related issues. In *Linguistica Computazionale vol. II - Computational Lexicology and Lexicography: Special Issue dedicated to Bernard Quemada*, pp. 139–161. Pisa, Giardini Editori e Stampatori.

Calzolari, N., A. Lenci, F. Bertagna, & A. Zampolli (2002). Broadening the scope of the eagles/isle lexical standardization initiative. In *COLING 2002 Workshop on Asian Language Resources and International Standardization*. Taipei.

Calzolari, N., A. Lenci, & A. Zampolli (2001). International standards for multilingual resource sharing: The isle computational lexicons working group. In *39th Annual Meeting and 10th Conference of the European Chapter, Workshop Proceedings: Sharing Tools and Resources*, pp. 71–78. Toulouse.

Correia, M. (1994). Bases digitais lexicais na união europeia. In *Simpósio de*

Lexicologia, Lexicografia e Terminologia, pp. 1–37.

Correia, M. (1996). Terminologia e lexicografia computacional. In *CAMBRÉ, MT. Cicle de conferéncies*, pp. 83–91. Barcelona: Institut Universitari de Lingüística Aplicada.

Courtois, B. (1990). Un système de dictionnaires électroniques pour les mots simples du français, langue française. *Dictionnaires électroniques du français 87*, 11–22.

da Silva, B. C. D., M. F. de Oliveira, & H. R. de Moraes (2002). Groundwork for the development of the brazilian portuguese wordnet. In *PorTAL, Advances in Natural Language Processing, Third International Conference, PorTAL 2002*, Volume 2389 of *Lecture Notes in Computer Science*, pp. 189–196. Faro, Portugal: Springer.

da Silva, B. C. D., M. F. Oliveira, H. R. Moraes, R. Hasegawa, D. Amorim, C. Paschoalino, & A. C. Nascimento (2000). A construção de um thesaurus eletrônico para o português do brasil. In *V Encontro para o processamento computacional da Língua Portuguesa Escrita e Falada (PROPOR'2000)*, pp. 1–11. Atibaia, SP.

da Silva, B. D. (2003). Human language technology research and the development of the brazilian portuguese wordnet. In *Proceedings of the 17th International Congress of Linguists*, pp. 12. Prague: Matfyzpress, MFF UK.

EAGLES (1993). Eagles lexicon architeture - eagles document eag-clwg-lexarch/b.

EAGLES (1996). Eagles computational lexicons working group reading guide - eagles document eag-clwg-fr-2.

Evans, R. & A. Kilgarriff (1995). Mrds, standards and how to do lexical engeneering. Technical Report ITRI-95-19, University of Brighton.

Greghi, J. G. (2002). Uma base de dados lexicais para o português do brasil. Dissertação de Mestrado, ICMC-USP, São Carlos, SP.

Greghi, J. G., R. T. Martins, & M. das Gracas Volpe Nunes (2002). Diadorim: a lexical database for brazilian portuguese. In *Proceedings of the Third International Conference on language Resources and Evaluation. LREC2002.*, Volume IV, pp. 1346–1350. Las Palmas, Ilhas Canárias.

Ide, N. & J. Véronis (1994). A feature-based data model for lexical databases. In *Hockey, S., Ide, N. Research in Humanities Computing IV*, pp. 193–206. Oxford University Press.

Kowaltowski, T. & C. L. Lucchesi (1993). Applications of finite automata representing large vocabularies. *Software-Pratice and Experience 23*(1), 15–20.

Kowaltowski, T., C. L. Lucchesi, & J. Stolfi (1995a). application of finite automata in debugging natural language vocabularies. *Journal of the Brazilian Computing Society 3*(1), 5–11.

Kowaltowski, T., C. L. Lucchesi, & J. Stolfi (1995b). Minimization on binary automata. *Journal of the Brazilian Computing Society 3*(1), 36–42.

Laporte, E. & A. Monceaux (1998). Elimination of lexical ambiguities by grammars. the elag system. *Lingvisticae Investigationes XXII*, 341–367. Amsterdam-Philadelphie: Benjamins.

Martins, T. B. F., R. Hasegawa, M. G. V. Nunes, & O. N. O. Jr. (1998). Linguistic issues in the develepment of regra: a grammar checker for brazilian portuguese. *Natural Language Engineering 4*(4), 287–307.

Miller, G. A., R. Backwith, C. Fellbaum, D. Gross, & K. Miller (1990). Introduction to wordnet: An on-line lexical database. *Journal of Lexicography 3*(4), 234–244.

Nunes, M. G. V., B. C. D. da Silva, L. H. M. Rino, O. N. O. Jr., R. T. Martins, & G. Montilha (1999). Introdução ao processamento de línguas naturais. Technical Report ND-38, ICMC-USP, São Carlos, SP. 91p.

Nunes, M. G. V., F. M. C. Vieira, C. Zavaglia, C. R. C. Sossolote, & J. Hernandez (1996). A construção de um léxico de português do brasil: Lições aprendidas e perspectivas. In *Anais do II Workshop de Processamento Computacional de Português Escrito e Falado (PROPOR'96)*, pp. 61–70. CEFET-PR, Curitiba.

Ranchhod, E., C. Mota, & J. Baptista (1999). A computational lexicon of portuguese for automatic text parsing. In *Proceedings of SIGLEX99: Standardizing Lexical Resources, 37th Annual Meeting of the ACL*, pp. 74–80. College Park, Maryland, USA.

Ranchhod, E. M. (2001). *Tratamento das Línguas por Computador. Uma Introdução à Linguística Computacional e suas Aplicações*, Chapter O uso de dicionários e de autómatos finitos na representação lexical das línguas naturais, pp. 13–47. Lisboa: Caminho.

Savary, A. (2000). *Recensement et description des mots composés - méthodes et applications*. Tese de Doutorado, Université de Marne-la-Vallée, França.

Silberztein, M. (1990). Le dictionnaire électronique des mots composés, langue française. *Dictionnaires électroniques du français 87*, 71–83.

Silberztein, M. (2000a). Intex: a fst toolbox. *Theoretical Computer Sciences 231*, 33–46.

Silberztein, M. (2000b). *INTEX Manuel*. LADL, Université Paris 7. `http://www.nyu.edu/pages/linguistics/intex/downloads/Manuel.pdf` (16/02/2004).

Tiberius, C. (1999). Language sampling for multilingual lexical representation. Technical Report ITRI-99-12, University of Brighton.

Vietri, S. & A. Elia (2000). Electronic dictionaries and linguistic analysis of italian large corpora. In *JADT 2000 : 5es Journées Internationales d'Analyse Statistique des Données Textuelles*, pp. 91–97.

Vitas, D. & C. Krstev (2001). Intex and slavonic morphology. In *Proceedings of the 4th Intex workshop*. Bordeaux, France.

Wilks, Y., D. Fass, C.-M. Guo, J. McDonald, T. Plate, & B. Slator (1988). Machine tractable dictionaries as tools and resources for natural language processing. In *Proceedings of Colling'88*, pp. 750–755. Budapest.