INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Identificação Automática de Nomes Compostos

Ricardo Jorge Rosa Portela

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Júri

Presidente:	Doutor João António Madeiras Pereira
Orientador:	Doutor Nuno João Neves Mamede
Co-Orientador:	Doutor Jorge Manuel Evangelista Baptista
Vogais:	Doutor Bruno Emanuel da Graça Martins

Novembro 2011

Agradecimentos

Gostaria de agradecer aos meus orientadores Prof. João Mamede e Prof. Jorge Baptista, pela dedicação, tempo, apoio e desafios mais complexos que me colocaram na realização deste trabalho.

Gostaria também de agradecer a toda a equipa do L^2F, especialmente ao Tiago Luís pela sua ajuda com as ferramentas Hadoop e Condor e à Teresa Mimoso pela sua boa disposição e ajuda com assuntos burocráticos.

Aos meus colegas de trabalho Fernando Gomes e Andreia Maurício, com quem partilhei reuniões, cafés e discuti ideias e soluções ao longo deste percurso.

À Fundação para a Ciência e Tecnologia pela concessão da bolsa de investigação.

À minha grande amiga Claudia Pereira, pelos cafés e conversas nos fins de tarde.

E finalmente queria também agradecer a todos os membros das secções autónomas Grupo de Estratégia Simulação e Táctica e Rádio Zero, pela excelente companhia nos almoços e tempos livres.

A todos o meu profundo agradecimento.

Lisboa, Novembro 2011
Ricardo Jorge Rosa Portela

Aos meus pais.

Resumo

Esta tese centra-se na identificação de nomes compostos na língua portuguesa. Nomes compostos são sequências de palavras cujo significado não pode ser extraído através da composição do significado literal das palavras, mas sim o seu significado figurativo quando certas palavras se encontram juntas. Esta tarefa pertence à área de processamento de língua natural (PLN) e é útil em sistemas de tradução, sistemas Pergunta-Resposta, extracção de informação e sumarização automática. Este documento analisa e compara vários sistemas usados para a identificação de termos compostos, descreve os procedimentos adoptados para a identificação destes mesmos termos e descreve o procedimento a ser efectuado para avaliar os resultados obtidos.

Abstract

This thesis focuses on the identification of multiwords in the Portuguese language. Multiwords are sequences of words whose meaning can not be extracted through the composition of the literal meaning of its words, but its figurative meaning when certain words are together. This task belongs to the area of natural language processing (NLP) and is useful in machine translation systems, question-answer systems, information extraction and automatic summarization. This paper analyzes and compares various systems used for the identification of multiwords, describes the procedures adopted for the identification of these multiwords and describes the procedure to be performed to evaluate the results.

Palavras Chave
Keywords

Palavras Chave

Nome Composto

Métodos Estatísticos

Algoritmos

Critérios Sintácticos

Corpus

Keywords

Multiword

Statistical Methods

Algorithms

Syntactic Criteria

Corpus

Índice

List of Figures

List of Tables

Acronyms

CAM Combined Association Messure

GALEMU Genetic Algorithm for the Extraction of Multiword Units

HDFS Hadoop Distributed File System

HELAS Hybrid Extraction of Lexical Associations

LVQ Learning Vector Quantization

ME Mutual Expectation

NE Normalized Expectation

PLN Processamento de Língua Natural

RuDriCo-2 Rule Driven Converter

SCP Symmetrical Conditional Probability

SMI Specific Mutual Information

SSC Simpson Similarity Coefficient

STRING Statistical and Rule-based Natural Language Processing

XIP Xerox Incremental Parser

Chapter 1

Introdução

1.1 Motivação

A unidade lexical pode ser definida como uma expressão a que se encontra associado um ou mais significados (Azuaga, Faria, Ribeiro, Duarte, & Gouveia 1996). Chama-se palavra composta quando duas ou mais unidades lexicais formam uma combinação em que apresentam um conceito novo, diferente da composição do significado dos elementos componentes. Por exemplo, *chapéu de chuva* é uma palavra composta porque o seu significado (objecto) é diferente da composição dos significados de *chapéu* e de *chuva* separadamente.

A identificação automática de palavras compostas pertence à área do Processamento de Língua Natural (PLN) e é útil em sistemas de tradução, sistemas Pergunta-Resposta, extracção de informação e sumarização automática. Entre outras aplicações que envolvam a identificação das unidades de significado dos textos.

As palavras compostas podem pertencer a diferentes categorias gramaticais: nomes, adjetivos (bonito, alto, grande, etc...), preposições (de, em, para, etc...), conjunções (mas, e, logo, como, etc...), etc. Os nomes compostos constituem provavelmente o conjunto mais numeroso das palavras compostas do léxico de muitas línguas naturais. No caso dos nomes compostos, estes podem apresentar diferentes estruturas morfossintácticas. Por exemplo, o nome *chapéu de chuva* é constituído por dois nomes ligados por uma preposição, já o nome *buraco negro* é constituído por um nome e por um adjetivo. Uma das dificuldades na identificação de termos compostos é justamente o facto de os nomes compostos apresentarem uma estrutura interna idêntica à dos grupos nominais ordinários (chapéu de cabedal/buraco escuro) tal como nas palavras simples, alguns compostos também podem ser ambíguos, permitindo uma leitura composicional (várias unidades lexicais) ou não (um composto), dependendo do contexto em que forem empregues.

Nesse sentido, pode não ser desejável classificar uma dada combinação como um termo composto. Por exemplo, o nome composto *braço direito* pode referir a uma pessoa de confiança,

mas num dado texto pode estar a fazer referência ao membro superior de uma pessoa.

1.2 Objectivos do Trabalho

Pretende-se neste estudo desenvolver um sistema que permita a identificação automática de candidatos a nomes compostos, isto é, combinações de palavras ainda não lexicadas, que formam uma só unidade lexical.

A identificação automática destes candidatos permiteria, por um lado, um muito mais eficiente trabalho de classificação por parte de um linguista e a sua integração nos léxicos de sistemas de PLN. Por outro lado, a ampliação da cobertura dos léxicos já disponíveis deverá resultar numa muito maior precisão das diversas aplicações dependentes da correcta identificação das unidades de sintaxe num texto, nomeadamente a análise sintáctica (parsing) e a extração de informação.

1.3 Estratégia

A identificação de termos compostos será efectuada através de técnicas de processamento de língua natural, fazendo uso entre outros recursos, da ferramenta XIP (Xerox Incremental Parser, (Aït-Mokhtar, Salah; Jean-Pierre Chanod, and Claude Roux 2002)), que é parte da cadeia de processamento de língua natural STRING (Statistical and Rule-based Natural Language Processing) (Mamede 2011) desenvolvido no L^2F[1]. Para a obtenção de padrões e respectivas frequências, usar-se-á o *corpus* CETEMPúblico (Santos & Rocha 2001), um *corpus* de texto jornalístico obtido a partir do diário *Público*[2] contendo 190 milhões de palavras.

Após a identificação de termos compostos, a nova informação será inserida na cadeia STRING. Tal exige que os dados sejam processados de novo pela cadeia de processamento do L^2F. Como este ciclo consome demasiado tempo de processamento, utilizar-se-á a rede de computadores do L^2F (GRID) e a ferramenta Condor, graças à qual, os processos são executados de forma paralela, reduzindo significativamente o tempo de cada ciclo e consequentemente o tempo necessário à identificação e validação de novos candidatos a termos compostos.

Os dados obtidos pela cadeia de processamento do L^2F são extensos sendo armazenados com o auxílio da ferramenta Hadoop, o que ajuda a aplicar os métodos estatísticos e algoritmos, referidos na secção 2, para a identificação dos termos compostos.

[1] http://www.l2f.inesc-id.pt/
[2] http://www.publico.pt/

Figure 1.1: Cadeia de Processamento STRING

1.4 Ferramentas Utilizadas

Esta secção faz uma descrição das ferramentas usadas para o processamento de dados em que este sistema se insere.

1.4.1 Cadeia de Processamento

A cadeia de processamento STRING (Mamede 2011) é composta por vários módulos, sendo cada módulo responsável por efectuar uma tarefa específica. A figura 1.1 apresenta esquemáticamente a sequência de módulos por que é formada a cadeia de processamento.

No primeiro passo da cadeia faz-se a segmentação do texto (*tokenization*) e a identificação de certos tipos de entidades textuais, como por exemplo, endereços, números romanos, números inteiros e decimais, símbolos, sinais de pontuação, abreviaturas e sequências de caracteres não aceites pelo analizador morfossintáctico.

De seguida, faz-se a etiquetagem morfossintáctica das palavras identificadas anteriormente.

O módulo responsável por esta tarefa, LexMan (Diniz 2010), associa às palavras campos específicos (categoria gramatical, subcategoria, modo, tempo, pessoa, número, género, grau, tipos de formação, caso). O sistema considera 11 categorias gramaticais (part-of-speech): nome, verbo, adjetivo, advérbio, pronome, preposição, conjunção, artigo, numeral, interjeição, e último pontuação).

No próximo passo, procede-se à divisão do texto em frases, usando como terminadores os segmentos terminados com ".", "!" ou "?".

A seguir, tem lugar uma desambiguação morfossintáctica por regras. O sistema responsável por esta tarefa tem o nome de Rule Driven Converter (RuDriCo-2) (Pardal 2007; Diniz 2010). Este módulo modifica a segmentação feita pelo analisador morfológico LexMan, aplicando regras de desambiguação morfossintáctica, regras para desfazer as contrações (e.g. de preposições e determinantes como nós = em + os)e regras de identificação de locuções (adverbiais, conjuncionais e outras, e.g. apesar de, ao longo de) não ambíguas.

Segue-se o módulo de desambiguação morfossintáctica estatística, Marv (Ribeiro, Oliveira, & Trancoso 2003), que utiliza o algoritmo de Viterbi para seleccionar a etiqueta mais provável para cada palavra no contexto em que se encontra. Como só usa informação sobre a categoria e subcategoria, se a palavra tiver associadas várias etiquetas após a selecção de categoria e subcategoria, escolhe-se arbitrariamente a primeira etiqueta. Este módulo tem uma precisão com cerca de 96% e foi treinado com um corpus de cerca de 250 mil palavras.

Finalmente é executado o XIP (Xerox Incremental Parser (Aït-Mokhtar, Salah; Jean-Pierre Chanod, and Claude Roux 2002))que introduz nova informação léxical, aplica regras de desambiguação morfossintáctica e gramáticas locais, segmenta as frases em constituintes elementares (*chunks*) e calcula as dependências sintácticas entre estes.

Na medida em que o XIP é um elemento central da informação processada e que foi utilizado para o desenvolvimento deste estudo, apresentaremos este último módulo da cadeia STRING de forma mais pormenorizada na secção seguinte.

1.4.2 XIP

O XIP recebe um texto como entrada e fornece informação lexical acerca do mesmo, faz a desambiguação lexical, segmenta o texto em *chunks* e cria as suas dependências. O sistema em si é completamente independente da língua, sendo a gramática de cada língua particular constituída por um conjunto de ficheiros de entrada. Para realizar estas tarefas, o XIP está dividido em três módulos, que descrevemos as seguir (ver figura 1.2).

O primeiro módulo, o módulo de desambiguação contextual, atribui a leitura mais provável a uma palavra com base no seu contexto imediato atribuindo *features* ou categorias às palavras.

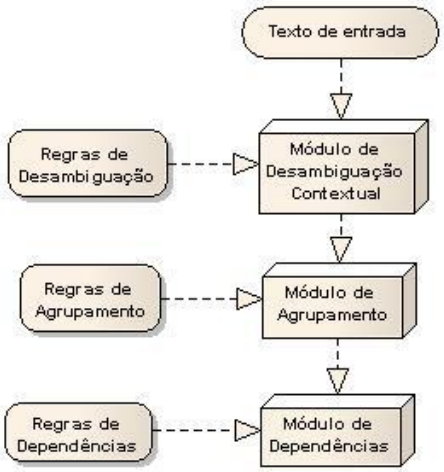


Figure 1.2: Arquitectura XIP

Depois, o módulo de análise sintáctica, faz a segmentação das unidades linguísticas em constituintes elementares (*chunks*) usando regras de agrupamento para agregar as sequências de categorias gramaticais em sintagmas. Finalmente, o módulo de extracção de dependências, determina as relações sintácticas (sujeito, complemento, etc...) entre os sintagmas previamente identificados.

A gramática do XIP para o Português é composta por um conjunto de ficheiros, que contêm as regras que permitem fazer a desambiguação, segmentação e encontrar as relações de dependência num texto.

A gramática é constituída por três tipos de ficheiros:

- *Declarações* das etiquetas usadas para descrever traços, categorias e dependências nas regras do XIP;
- Diferentes tipos de *regras*, que recorrem a operadores e expressões regulares para testar os traços de um nó;
- Um *ficheiro de configuração* onde se encontram declarados todos os ficheiros constituintes da gramática.

O XIP tem três tipos de regras, as Regras de Dominância Imediata (Immediate Dominance Rules), as Regras de Sequência (Sequence Rules) e as Regras de Dependência (Dependency Rules). As regras de dominância imediata, assim como as de sequência, são regras de agrupamento. No entanto, as regras de dominância imediata são aplicadas independentemente da ordem pela qual os nós surgem no lado direito da regra. Para as regras de sequência, é estritamente necessário que os nós no texto de entrada surjam exactamente pela ordem na qual se

encontram no lado direito das regras para que estas possam ser aplicadas. Se for possível aplicar várias regras de dominância imediata, o factor de escolha é baseado na sequência mais longa, sendo o texto de entrada lido da direita para a esquerda. Para as regras de sequência, estas são aplicadas sequencialmente pela ordem definida pelo programador e o texto de entrada também é lido da esquerda para a direita.

Veja-se abaixo uma Regra de Dominância Imediata que cria um nó NP para uma lista de categorias que contenha um determinante, um nome e um adjetivo, em qualquer ordem:

$$NP\text{-} > det, noun, adj.$$

Compare-se agora a regra acima com uma Regra de Sequência que cria um nó NP para uma lista de categorias que contenha um determinante, (facultativamente) um adjetivo e um nome por esta ordem:

$$NP = det, (adj), noun.$$

Ao nível da representação, a única diferença entre as regras de dominância imediata e as de sequência é o caso dos operadores "=" e "->".

As regras de dependência são usadas para a extracção das relações sintácticas entre os diferentes constituintes da frase, como por exemplo entre o sujeito e o verbo; podem ainda ser usadas para adicionar ou remover traços a um nó.

Veja-se um exemplo de uma Regra de Dependência que cria uma dependência sujeito-verbo-objecto para um nó NP que já contenha uma dependência de sujeito e uma dependência de objecto:

$$|NP\{?, \#1[last]\}|if(subject(\#2, \#1)\&object(\#2, \#3))SVO(\#1, \#2, \#3).$$

1.4.3 Condor

Devido à quantidade de dados a serem processados, a sua computação pela cadeia de processamento levaria semanas se tivesse de ser realizada numa única máquina. O Condor (Tannenbaum, Wright, Miller, & Livny 2001) providencia um mecanismo de fila de espera, regime de prioridade, acompanhamento, e gestão de recursos de forma a se poder executar os processos paralelamente sobre uma rede de máquinas, podendo assim processar os dados de uma forma mais rápida.

O Condor pode usar eficientemente o poder computacional desperdiçado de máquinas que estejam paradas. Se o Condor detectar que uma máquina já não está disponível, ele é capaz de produzir um "checkpoint" que marca onde o processamento parou, para migrar o trabalho para uma máquina diferente que esteja parada e assim continuar o processamento onde tinha parado anteriormente.

O Condor providencia um ambiente de trabalho extremamente flexível e expressivo para alocar processos a máquinas. Certos processos têm requerimentos e preferências específicas, assim como as máquinas podem especificar requerimentos e preferências acerca dos processos que estão dispostos a processar. Estas preferências e requerimentos podem ser descritos através de expressões, de forma a que o Condor se possa adaptar a qualquer ambiente de trabalho. O Condor incorpora também protocolos e metodologias de computação GRID.

1.4.4 Hadoop

A Hadoop (Luís 2008) implementa o modelo de programação MapReduce e possui um sistema de ficheiros distribuído chamado Hadoop Distributed File System (HDFS). O HDFS foi desenhado para guardar de forma segura grandes quantidades de dados por várias máquinas de uma rede. Este sistema providencia uma interface que ajuda a executar processos dependendo da localização dos dados, minimizando o consumo da rede e aumentando o fluxo global de processamento

O paradigma MapReduce trabalha exclusivamente sobre pares chave/valor, ou seja, recebe como entrada uma lista de pares chave/valor e produz uma lista de pares chave/valor. Estes pares podem representar qualquer tipo de dados. Este paradigma de programação opera em duas tarefas: A primeira é a Map, em que se produz uma lista de pares chave/valor intermédios. Cada lista é um processo individual que foi corrido numa máquina. A segunda tarefa é a Reduce, em que se cria uma lista de pares mais pequena a partir das listas intermédias que tenham a mesma chave. Esta fase é dividida em outras três fases:

- Shuffle - vai buscar os pares chave/valor relevantes produzidos pelo Map;
- Sort - esta fase ocorre simultaneamente com a fase Shuffle, para agrupar os pares que tenham a mesma chave;
- Reduce - recebe os pares agrupados produzidos pela fase Sort e produz os pares finais.

Existem também quatro controladores da execução das tarefas: (1) O *Partitioner*, que controla o particionamento das chaves dos pares intermédios; O número máximo de partições é igual ao número de tarefas Reduce; (2) O *Combiner*, que faz um Reduce local aos pares chave/valor de saída do Map; (3) O *Input Format*, que controla a divisão do ficheiro de entrada e converte cada uma das divisões numa lista de pares chave/valor; (4) O *Output Format*, que controla o destino dos pares chave/valor finais.

O HDFS fornece um grande fluxo de acesso aos dados e é próprio para aplicações que envolvem grandes quantidades de dados. Possui uma arquitectura mestre/escravo da qual um grupo consiste num NameNode, um servidor mestre que gere o espaço de nomes do sistema

de ficheiros e regula o acesso aos ficheiros pelos clientes. O HDFS também tem DataNodes, normalmente um por cada nó no grupo, que geram o armazenamento dos nós onde correm. Internamente, um ficheiro é dividido em um ou mais blocos e esses blocos são guardados numa lista de DataNodes. O NameNode executa a abertura, fecho e atribuição de nomes aos ficheiros e directorias do espaço de nomes do sistema de ficheiros. O NameNode também determina o mapeamento dos blocos para DataNodes. Estes DataNodes são responsáveis pela gestão de pedidos de leitura e escrita pelos clientes do sistema de ficheiros. Os DataNodes também podem fazer a criação, destruição e replicação de blocos tendo sido instruídos previamente pelo NameNode. O HDFS foi desenhado para guardar grandes ficheiros entre várias máquinas num grupo grande, guardando cada ficheiro como uma sequência de blocos. Os blocos são replicados para fornecer tolerância a faltas, sendo esses blocos todos do mesmo tamanho, exceptuando o último, esta replicação pode ser definida na criação do ficheiro e pode ser alterada mais tarde.

As ferramentas descritas anteriormente irão ajudar a obter a informação necessária, de uma forma mais rápida, para a aplicação de alguns dos métodos e algoritmos descritos no capítulo 2.

1.4.5 Roteiro

Esta dissertação encontra-se organizada do seguinte modo: No capítulo 2 é feita uma descrição de vários métodos usados para a identificação de termos compostos. O capítulo 3 apresenta os passos para a implementação de identificação de compostos. No capítulo 4 faz-se a descrição dos critérios de avaliação e é apresentado os resultados obtidos, finalmente, no capítulo 5 apresentam-se as conclusões do estudo assim como o trabalho futuro.

Chapter 2

Trabalho Relacionado

Este capítulo faz uma descrição dos métodos estatísticos e algoritmos usados para a identificação automática de termos compostos, assim como também é feita uma comparação dos algoritmos apresentados.

Também é apresentado uma descrição dos critérios sintáticos que compõem as estruturas *Nome Adjetivo* e *Nome de Nome*.

2.1 Métodos estatísticos

Esta secção descreve os métodos estatísticos usados pelos vários algoritmos e sistemas de identificação de termos compostos, que serão descritos na secção 2.2.

2.1.1 Dice coefficient

O coeficiente de Dice (Smadja, McKeown, & Hatzivassiloglou 1996) (Dice 1945) consiste em medir o grau de coesão/fixidez que existe entre duas palavras de um bi-grama $[w_1 p_{12} w_2]$, sendo definido pela equação (1).

$$Dice([w_1 p_{12} w_2]) = \frac{2 \times f([w_1 p_{12} w_2])}{f([w_1]) + f([w_2])} \tag{1}$$

em que $f([w_1 p_{12} w_2])$, $f([w_1])$ e $f([w_2])$ representam respectivamente as frequências do bigrama $[w_1 p_{12} w_2]$ e dos unigramas $[w_1]$ e $[w_2]$, p_{12} representa a distância entre as palavras w_1 e w_2.

2.1.2 Specific Mutual Information

O método *Specific Mutual Information* (Church & Hanks 1990) é usado para medir a sobreposição entre duas ocorrências, contribuindo assim para uma medição do grau de coesão entre duas palavras de um bigrama, e sendo definido pela equação (2).

$$SMI([w_1 p_{12} w_2]) = \log_2 \frac{N \times f([w_1 p_{12} w_2])}{f([w_1]) \times f([w_2])} \tag{2}$$

em que $f([w_1 p_{12} w_2])$, $f([w_1])$ e $f([w_2])$ representam, respectivamente, as frequências do bigrama $[w_1 p_{12} w_2]$ e dos unigramas $[w_1]$ e $[w_2]$, N representa o número total de palavras no *corpus* e p_{12} representa a distância entre as palavras w_1 e w_2. Esta medida é particularmente propensa a sobreestimar dados com frequências baixas.

2.1.3 Pearson's χ^2

O método de *Pearson's* χ^2 (Hull & Grefenstette 1996) testa a hipótese nula, baseando-se na comparação das frequências observadas com as frequências esperadas. Frequência esperada é a frequência "justa" para as saídas possíveis num evento. Por exemplo, uma amostra com 100 bolas em que existem igual número de bolas pretas e vermelhas, a frequência esperada é 50% bolas pretas e 50% bolas vermelhas. Os eventos considerados têm de ser mutuamente exclusivos e ter uma probabilidade total de 1. Este método é definido pela equação (3).

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{3}$$

em que O_i representa a frequência observada, E_i a frequência esperada e n o número de saídas possíveis para cada evento.

2.1.4 ϕ^2

O método ϕ^2 (Gale & Church 1991) é baseado no Pearson's χ^2 para tabelas de contingência 2 x 2, testando a hipótese nula de que duas variáveis são independentes. A hipótese nula é normalmente representado por $H_0 : p(w_i p_{ij} w_j) = p(w_i) \times p(w_j)$. Se ϕ^2 for mínimo, a hipótese nula H_0 verifica-se e considera-se que as duas variáveis, isto é, as palavras de uma combinatória, são independentes. Caso contrário, considera-se que as duas variáveis estão relacionadas entre si, ou seja, neste caso, a combinatória apresenta um certo grau de fixidez. Este método é definido pela equação (4).

$$\phi^2([w_1 p_{12} w_2]) = \frac{(N \times f([w_1 p_{12} w_2]) - f([w_1]) \times f([w_2]))^2}{f([w_1]) \times (N - f([w_1])) \times f([w_2]) \times (N - f([w_2]))} \tag{4}$$

em que $f([w_1 p_{12} w_2])$, $f([w_1])$ e $f([w_2])$ representam, respectivamente, as frequências do bi-grama $[w_1 p_{12} w_2]$ e dos uni-gramas $[w_1]$ e $[w_2]$, N representa o número total de palavras no *corpus* e p_{12} representa a distância entre as palavras w_1 e w_2.

2.1.5 Log-likelihood Ratio

O método de *Log-likelihood Ratio* (Dunning 1993), tal como o método ϕ^2, testa a hipótese nula de que duas variáveis são independentes. A hipótese nula de independência estatística de duas variáveis é representado por $H_0 : p(w_i p_{ij}|w_j) = p(w_i p_{ij}|\overline{w_j})$ colocando o paradigma de independência entre duas linhas da tabela de contingência. Este método pode ser definido pela equação (5).

$$\begin{aligned} Loglike([w_1 p_{12} w_2]) = -2\log\lambda = \\ 2\mathsf{x}(\log\theta_1^{s_1}(1-\theta_1)^{n_1-s_1} + \log\theta_2^{n_2-s_2} \\ -\log\theta^{s_1}(1-\theta)^{n_1-s_1} - \log\theta^{s_2}(1-\theta)^{n_2-s_2} \end{aligned} \tag{5}$$

em que:

- $s_1 = f([w_1 p_{12} w_2])$
- $s_2 = f([w_2]) - f([w_1 p_{12} w_2])$
- $n_1 = f([w_1])$
- $n_2 = N - f([w_2])$
- $\theta_1 = \frac{s_1}{n_1}$
- $\theta_2 = \frac{s_2}{n_2}$
- $\theta = \frac{f([w_2])}{N}$
- $f([w_1 p_{12} w_2])$, $f([w_1])$ e $f([w_2])$ representam, respectivamente, as frequências do bigrama $[w_1 p_{12} w_2]$ e dos unigramas $[w_1]$ e $[w_2]$
- N representa o número total de palavras no *corpus*
- p_{12} representa a distância entre as palavras w_1 e w_2

2.1.6 Mutual Expectation

O método *Mutual Expectation* (Daille 1996; Dias, Guilloré, & Lopes 1999) é baseado no conceito de *Normalized Expectation (NE)*, cuja ideia é avaliar o custo, em termos de coesão, da perda de uma palavra num n-grama, ou seja, a probabilidade de uma palavra w_i ocorrer numa dada posição, sabendo a ocorrência das outras w_{i-1} palavras e as suas posições. Sabendo que um critério eficiente para a identificação de termos compostos é a frequência, pode-se deduzir com isto que entre dois n-gramas com o mesmo *NE*, o n-grama mais frequente é mais provavelmente um termo composto. O método é definido pela equação (6):

$$\begin{aligned} ME([w_1 \cdots p_{1i} w_i \cdots p_{1n} w_n]) = p([w_1 \cdots p_{1i} w_i \cdots p_{1n} w_n]) \\ \times NE([w_1 \cdots p_{1i} w_i \cdots p_{1n} w_n]) \end{aligned} \tag{6}$$

em que um n-grama é definido algebricamente pelo vector de palavras $[w_i \cdot p_{1i} w_i \cdot p_{1n} w_n]$, w_i uma palavra no n-grama, p_{1i} representa a distância que separa a palavra w_1 da palavra w_i, $p()$ indica a frequência e $NE()$ o cálculo da *Normalized Expectation*.

2.1.7 Simpson Similarity Coefficient

O método *Simpson Similarity Coefficient* (Martínez-Santiago, Díaz-Galiano, Martín-Valdivia, Rivas-Santos, & na Lopez 2002) avalia a associação entre duas palavras calculando a divisão da intersecção de duas palavras com o mais pequeno dos dois, de forma a não subvalorizar conjuntos em que uma das palavras possuiu uma frequência muito mais alta relativamente à palavra que se combina, o que daria uma valor muito baixo para este conjunto. Este método pode ser definido pela equação (7).

$$SIMPSON([w_1 p_{12} w_2]) = \frac{2 \times f([w_1 p_{12} w_2])}{\min(f([w_1]), f([w_2]))} \tag{7}$$

em que $f([w_1 p_{12} w_2])$, $f([w_1])$ e $f([w_2])$ representam, respectivamente, as frequências do bigrama $[w_1 p_{12} w_2]$ e dos unigramas $[w_1]$ e $[w_2]$ e p_{12} representa a distância entre as palavras w_1 e w_2.

2.1.8 Symmetrical Conditional Probability

O método *Symmetrical Conditional Probability* (Lopes & Silva 1999) mede a coesão de duas palavras num bi-grama pela equação (8):

$$SCP([x, y]) = p(x|y) \cdot p(y|x) = \frac{p([x, y])^2}{p([x]) \cdot p([y])} \tag{8}$$

em que p(x,y), p(x) e p(y) são, respectivamente, a probabilidade de ocorrência do bigrama [x,y] e dos unigramas [x] e [y] no *corpus*, $p(x|y)$ é a probabilidade condicional de x ocorrer na primeira posição do bigrama dado que y aparece na segunda posição. De forma similar $p(y|x)$ é a probabilidade condicional de y ocorrer na primeira posição do bigrama dado que x aparece na segunda posição do bigrama.

2.2 Algoritmos e Sistemas

Esta secção faz uma descrição dos algoritmos e sistemas estudados para a identificação de termos compostos.

2.2.1 Kohonen's Learning Vector Quantization

A Kohonen's Learning Vector Quantization (LVQ) (Kohonen 1989; Kohonen, Kangas, Laaksonen, & Torkkola 1992), para determinar se certos pares de palavras podem ser considerados como termos compostos ou não. As entradas para a rede são os valores gerados por um conjunto de estimadores estatísticos, e a saída da rede é uma classe, que determina se o valor corresponde a um termo composto ou não. A aprendizagem da rede é feita através dos valores gerados pelos estimadores quando estes são aplicados a pares de palavras previamente identificados como compostos e a outros pares de que não são compostos. Foram usados os seguintes estimadores estatísticos sendo alguns deles referidos atrás em 2.1:

- *Dice coefficient* (Adriani & Rijsbergen 1999);
- *Pearson's* χ^2 (Hull & Grefenstette 1996);
- *Simpson Similarity coeficient* (Martínez-Santiago, Díaz-Galiano, Martín-Valdivia, Rivas-Santos, & na Lopez 2002);
- Métrica *em* (Ballesteros & Croft 1998);
- *Mutual information ratio* (Johansson 1996).

O algoritmo LVQ é um método de classificação baseado em aprendizagem neural competitiva, que permite definir um grupo de categorias no espaço de dados de entrada por uma aprendizagem de reforço, ou seja, por reforço positivo (prémio) ou por reforço negativo (castigo). A equação (9) define o processo de aprendizagem básico para o algoritmo LVQ:

$$w_c(t+1) = w_c(t) + s \cdot \alpha(t) \cdot \lfloor x_i(t) - w_c(t) \rfloor \tag{9}$$

em que $x_i(t)$ é o vector de entrada no tempo t, e $w_k(t)$ representa o vector de peso para a classe k no tempo t; $\alpha(t)$ é o rácio de aprendizagem, sendo $0 < \alpha(t) < 1$, uma função monotonamente decrescente do tempo; $s = 0$, se $k \neq c$; $s = 1$, se $x_i(t)$ e $w_c(t)$ pertence à mesma classe; e $s = -1$ se não pertencerem.

O algoritmo LVQ funciona da seguinte forma: para cada classe k, associa-se um vector de peso w_k, em cada repetição, o algoritmo escolhe um vector de entrada x_i, e compara com o peso de cada vector w_k, usando a distância euclidiana $\|x_i - w_k\|$, para que o vencedor seja o vector de peso w_c mais perto de x_i, sendo c o seu índice:

$$\|x_i - w_i\| = \min_k\{\|x_i - w_k\|\} \tag{10}$$

As classes competem entre elas para encontrar o vector mais similar com o vector de entrada, para que o vencedor seja o que tenha a menor distância euclidiana tendo em consideração o vector de entrada. Só a classe vencedora irá modificar os seus pesos usando o algoritmo de aprendizagem por reforço, descrito anteriormente (9), dando reforço positivo ou reforço negativo, dependendo da classificação estar correcta ou errada. Assim, se a classe vencedora pertence à mesma classe que o vector de entrada (a classificação está correcta) os seus pesos são incrementados, aproximando-se do vector de entrada (prémio) ou fazendo o contrário, caso a classe ganhadora seja diferente da classe do vector de entrada.

De forma a treinar e testar a rede neuronal, foi criada uma lista com pares entrada-saída. Cada linha corresponde a um par de palavras, os valores de entrada foram obtidos aplicando os estimadores referidos anteriormente. Os valores de saída consistem num número que classifica o par de palavras como sendo um composto ou não. Nesta experiência só foram usados palavras compostas com duas palavras.

O autor deste trabalho para avaliar a rede, obteve uma lista de termos compostos usando a WordNet (Miller 1995) e o dicionário electrónico Encarta. Para a lista de termos não compostos (necessário para treinar a rede) retirou de um corpus usado no CLEF 2000 e comparou com a lista de termos compostos para verificar se não existiam pares iguais nas duas listas. Após obter o ficheiro com os pares entrada-saída, este foi dividido: 75% das amostras foram usadas para treinar a rede e as restantes para a validar.

Para testar, o autor fez *queries* para retirar informação do corpus CLEF 2000, da qual resultou uma melhoria da precisão em 4% com o uso de identificação de termos compostos através deste método (precisão de 41%), relativamente a um levantamento de informação sem a identificação de termos compostos (precisão de 37%).

2.2.2 HELAS

Foi proposto um sistema híbrido chamado HELAS (Dias 2003), que extrai candidatos a termos compostos de um corpus com as classificações gramaticais. Este sistema conjuga a medida Mutual Expectation (ME), acima apresentada, com um processo de aquisição chamado GenLocalMaxs, de forma a poder avaliar o grau de coesão de uma sequência de palavras, através da combinação do grau de coesão das palavras com o grau de coesão das suas classificações sintácticas do universo de discurso.

O primeiro passo deste sistema consiste em dividir o corpus em dois sub-corpus, um subcorpus das palavras e outro com as classificações gramaticais. Depois cada sub-corpus é segmentado num conjunto de n-gramas posicionais. Em paralelo cada n-grama posicional do subcorpus das palavras é associado à sua classificação do subcorpus de classificações, de forma a poder avaliar a coesão global de uma sequência de palavras e as suas classificações respectivas.

A ideia deste sistema é avaliar a coesão das associações palavra-classificação, ou seja, quanto mais coesão existir numa sequência de palavras e quanto mais coesão existir nas suas classificações no universo de discurso, mais provável é que essa sequência de palavras seja um termo composto. Assim o grau de coesão global pode ser avaliado através da combinação da ME das palavras e da ME das suas classificações gramaticais. Isto é avaliado pelo Combined Association Messure (CAM), definida na equação (11), em que α é um parâmetro que define o focos de maior relevância, ou seja, se a sequência de palavras tem maior peso que a sequência das suas classificações ou vice-versa.

$$\begin{aligned} CAM([p_{11}u_1t_1 \ldots p_{1i}u_it_i \ldots p_{1n}u_nt_n]) = \\ ME([p_{11}u_1 \ldots p_{1i}u_i \ldots p_{1n}u_n])^{\alpha} \\ \times \\ ME([p_{11}t_1 \ldots p_{1i}t_i \ldots p_{1n}t_n])^{1-\alpha} \end{aligned} \tag{11}$$

O processo de selecção dos termos compostos é feito através do algoritmo GenLocalMaxs que se concentra em identificar o máximo local dos valores das CAM's. Assim pode deduzir-se que um n-grama posicional palavra-classificação é um termo composto se o valor da sua CAM é igual ou maior do que os valores da CAM dos seus subgrupos de $(n-1)$ palavras e se é estritamente maior que o valor da CAM dos seus supergrupos de $(n+1)$ palavras. Este processo é definido pela equação (12).

$$\begin{gathered}\forall x \in \Omega_{n-1}, \forall y \in \Omega_{n+1}, W \text{ é uma palavra composta se} \\ (sizeof(W) = 2 \wedge CAM(W) > CAM(y)) \\ \vee \\ (sizeof(W) \neq 2 \wedge CAM(W) \geq CAM(x) \wedge CAM(W) > CAM(y))\end{gathered} \tag{12}$$

em que W é um n-grama posicional palavra-classificação, Ω_{n-1} o conjunto de todos os (n-1)-gramas posicionais contidos em W, Ω_{n+1} o conjunto de todos os (n+1)-gramas posicionais contidos em W e *sizeof()* uma função que devolve o número de palavras de um n-grama posicional palavra-classificação.

Os testes realizados pelo autor foram feitos sobre uma parte do *Brown Corpus* contendo 249.578 palavras e usando 11 valores diferentes para α, ou seja, $\alpha \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$, sendo o focus total nas palavras para $\alpha = 1$ e focus total nas classificações para $\alpha = 0$.

Os resultados obtidos mostraram que tanto a dependência de palavras como as dependências das classificações gramaticais têm uma tarefa importante na identificação de termos compostos, pois os melhores resultados foram obtidos para valores de α igual a 0,4 e 0,5. As sequências mais identificadas foram bigramas e trigramas, que atingiram uma precisão de 60% e 80%, respectivamente, para α igual a 0,5. Para os trigramas, verificou-se que a estrutura sintáctica tem um papel muito importante na identificação, pois a precisão caiu drasticamente quando o foco passou a dar mais relevância às dependências de palavras. No caso dos bigramas, demasiado focos nas dependências de palavras ou nas dependências das classificações levou a resultados insatisfatórios, sendo os melhores resultados obtidos através do equilíbrio entre os dois tipos de dependências. No entanto, a identificação de sequências de duas palavras continua a ser um problema para este sistema.

2.2.3 GALEMU

Foi proposto um algoritmo genético chamado GALEMU (Genetic Algorithm for the Extraction of Multiword Units) (Dias & Nunes 2004), que, como primeiro passo, vai segmentar o *corpus* numa lista de n-gramas posicionais. Depois, cada n-grama posicional é associado a uma lista de atributos com valores (por exemplo, frequência, tamanho, grau de coesão), que representa um cromossoma específico de toda a população. Depois da população estar definida, a *fitness function* providencia o melhor genótipo que é o máximo global. Finalmente para extrair os termos compostos, aplica-se uma medida de similaridade entre o n-grama posicional que está a ser analisado com o melhor genótipo escolhido anteriormente.

Para a identificação de sequências de palavras com um grau de coesão elevado foram definidas sete variáveis que correspondem às heurísticas da procura.

Heurística x_0: Quanto mais coesa for uma sequência de palavras, mais provável será ela constituir um termo composto. Assim a primeira heurística será definida como a medida de associação Mutual Expectation de um dado n-grama.

Heurística x_1: A frequência também é considerada como um critério forte para a identificação de palavras compostas, assim esta heurística é definida como a frequência de um dado n-grama.

Heurística x_2: É um facto que, se um n-grama aparecer dentro de outro n-grama mais longo (i.e. super-grupo), tal é um factor negativo para a sua relevância, a sequência das palavras aumenta em probabilidade de importância com o aumento do número destes n-gramas mais longos. Este número é considerado como uma heurística.

Heurística x_3: Quanto mais um n-grama contiver palavras simples com uma frequência elevada, menos relevante será esse N-grama. Como quarta heurística é medida a frequência de todos os elementos constituintes do n-grama de forma a medir a sua relevância, a que se chama frequência marginal.

A partir destas heurísticas pode-se definir a *fitness function* (13). No entanto, em problemas de optimização existem constrangimentos que são definidos nas restantes heurísticas.

$$g(X) = x_0 + x_1 + x_2 - x_3 \tag{13}$$

Heurísticas x_4 e x_5: Um n-grama posicional é um termo composto se o seu valor de associação é maior ou igual do que os valores de associação dos seus subgrupos de palavras e se for estritamente maior que os valores das medidas de associação dos seus super-grupos de palavras. Assim estas heurísticas são respectivamente o valor mais alto da Mutual Expectation dos subgrupos do genótipo escolhido e o valor mais alto da Mutual Expectation dos seus super-grupos. Estas heurísticas podem ser definidas pelas inequações (14) e (15).

$$x_0 \geq x_4 \tag{14}$$

$$x_0 > x_5 \tag{15}$$

Heurística x_6: Se a frequência de um dado n-grama é igual à frequência de um n-grama maior do que aquele que o contém, então o n-grama mais curto não deve ser tomado em conta como uma associação de palavras relevante (16).

$$x_6 < x_1 \tag{16}$$

No entanto, puderam ser formulados novos constrangimentos ao problema que introduziram novo conhecimento. Foi formulado que a frequência marginal de um n-grama tem de ser superior ou igual à sua frequência relativa (17), da mesma maneira que o número de super-grupos diferentes de um dado n-grama não pode ser superior à sua frequência relativa (18).

$$x_3 \geq x_1 \tag{17}$$

$$x_2 \leq x_1 \tag{18}$$

Depois de ter sido escolhido o melhor genótipo, é usada uma medida de similaridade para avaliar o relacionamento de cada n-grama com o genótipo. Quanto mais distante esses dois pares, menos similares eles serão. Foram usadas quatro medidas de similaridade: a medida euclidiana (19), a medida de divergência (20), a medida de *Bray/Curtis* (21) e a medida de *Soergel* (22).

$$D_{ij}^1 = \frac{1}{p}\sum_{k=1}^{p}(X_{ik} - X_{jk})^2 \ \textit{Euclidean} \tag{19}$$

$$D_{ij}^2 = \frac{1}{p}\sum_{k=1}^{p}\frac{(X_{ik} - X_{jk})^2}{(X_{ik} + X_{jk})^2} \ \textit{Divergence} \tag{20}$$

$$D_{ij}^3 = \frac{\sum_{k=1}^{p}|X_{ik} - X_{jk}|}{\sum_{k=1}^{p}(X_{ik} + X_{jk})} \ \textit{Bray/Curtis} \tag{21}$$

$$D_{ij}^4 = \frac{\sum_{k=1}^{p}|X_{ik} - X_{jk}|}{\sum_{k=1}^{p}\max(X_{ik}, X_{jk})} \ \textit{Soergel} \tag{22}$$

A distância entre duas unidades i e j é definida como $D_{ij} = f(X_i, X_j)$, onde f é uma função de medida, X_j é o genótipo e X_i o n-grama.

Para testar foi usado um manual de Linux em inglês com aproximadamente 54.000 palavras. Os melhores resultados obtidos foram 71% e 70% de precisão respectivamente para as medidas *Bray/Curtis* e *Soergel*, no entanto esta última medida só extraiu 40% dos termos seleccionados pela *Bray/Curtis*. As medidas *Euclidean* e *Divergence* atingiram, respectivamente, a precisão de 64% e 62%.

2.2.4 Algoritmo *LocalMaxs*

O algoritmo *LocalMaxs* (Silva, Dias, Guilloré, & Lopes 1999) é um algoritmo que identifica termos compostos a partir de uma lista de n-gramas baseando-se em dois pressupostos: Primeiro, as medidas de associação mostram que, quanto mais coeso for um grupo de palavras, mais alto será o valor da medida de associação para a sua identificação. Segundo, termos compostos são

grupos de palavras que estão bastante associadas, como consequência, um n-grama W é um termo composto se o seu valor de associação *g(W)* for um máximo local. O algoritmo pode ser definido pela equação (23).

$$\begin{gathered}\forall x \in \Omega_{n-1}, \forall y \in \Omega_{n+1} W \text{ é uma palavra composta se} \\ (length(W) = 2 \wedge g(W) > g(y)) \\ \vee \\ (length(W) > 2 \wedge g(x) \leq g(W) \wedge g(W) > g(y))\end{gathered} \tag{23}$$

Sendo Ω_{n-1} o conjunto de dos valores de associação de todos os (n-1)-gramas contidos no n-grama W, e Ω_{n+1} o conjunto dos valores de associação de todos os (n+1)-gramas contidos no n-grama W; um n-grama será um termo composto se o seu valor *g()* da medida de associação corresponder a um máximo local.

Este algoritmo não usa medidas de limiar e concentra-se na identificação das variações locais dos valores das medidas de associação. Foram usadas várias medidas estatísticas, sendo estas normalizadas pelo *Fair Dispersion Point Normalization*(Silva, Dias, Guilloré, & Lopes 1999) para a identificação de termos compostos formados por palavras contínuas. Os estimadores que foram usados são os seguintes:

- Dice coefficient;
- Specific Mutual Information(SMI);
- ϕ^2;
- Log-likelihood Ratio;
- Symmetric Conditional Probability(SCP).

Foi usado um *corpus* com 919.253 palavras para testar a precisão do algoritmo, tendo sido atingido o valor mais alto de 81% para a Symmetric Conditional Probability.

Para os termos compostos formados por palavras não contínuas, foram usadas várias medidas estatísticas, sendo estas normalizadas pelo *Normalized Expectation Measure* (Silva, Dias, Guilloré, & Lopes 1999) e o *Fair Point of Expectation* (Silva, Dias, Guilloré, & Lopes 1999). Os estimadores que foram usados são os seguintes:

- Dice coefficient;
- Specific Mutual Information(SMI);

- ϕ^2;
- Log-likelihood Ratio;
- Mutual Expectation(ME).

Somente a medida estatística Mutual Expectation não foi normalizada, por ser a única das referidas que está preparada para calcular o grau de coesão para sequências com mais de duas palavras. Para testar a precisão deste algoritmo, foi usado um *corpus* de debates políticos com aproximadamente 300.000 palavras e só se realizaram os testes para termos compostos não contínuos com exactamente uma palavra de intervalo, tendo sido atingido o valor de 90% para a Mutual Expectation. Devido ao uso das medidas estatísticas, este método continua com o mesmo problema dessas medidas, que é as palavras que possuem uma frequência muito elevada relativamente às outras palavras da mesma combinação, pois estas medidas sobreestimam o grau de coesão quando uma a probabilidade marginal de uma das palavras é demasiado elevado.

2.2.5 C-value/NC-value

Este método (Frantzi, Ananiadou, & Mima 2000) combina dois tipos de informação para extrair termos compostos de um *corpus*, a informação linguística e a estatística. Primeiro, o método *C-value* extrai os termos compostos e depois o método *NC-value* introduz informação de contexto ao resultado do método anterior para melhorar a extracção de termos compostos.

A informação linguística é obtida em três passos: Primeiro, é efectuada uma classificação gramatical a cada palavra do corpus. Segundo é colocado um filtro linguístico de forma a extrair os termos que obedecem a uma estrutura gramatical já definida. Finalmente, é usado uma *stop-list*, que é uma listagem de palavras que não são palavras compostas, para evitar a extracção de sequências de palavras que aparecem frequentemente mas que não são termos compostos.

A informação estatística consiste em atribuir um valor às sequências de palavras candidatas. Esta medida é feita tendo em conta os seguintes valores:

- a frequência total de ocorrências da sequência de palavras candidata;
- a frequência total da sequência de palavras candidatas como parte de outras sequências de palavras candidatas mais longas;
- o número dessas sequências de palavras candidatas mais longas;
- o número de palavras que compõe a sequência de palavras candidata.

Assim, a medida *C-value*, é dada pela função (24).

$$C-value(a) = \begin{cases} \log_2 |a| f(a) & a \text{ não está inserida noutra} \\ & \text{palavra composta} \\ \log_2 |a| (f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b)) & \text{caso contrário} \end{cases} \quad (24)$$

Onde a é a sequência de palavras candidata, *f()* é a frequência total de ocorrências no *corpus*, T_a é uma lista de sequências de palavras que contém a e $P(T_a)$ é o número dessas sequências de palavras candidatas.

Este método começa por calcular o *C-value* para as sequências de palavras mais longas, terminando depois na sequência mais pequena, depois de calcular o *C-value* para uma dada sequência de palavras, caso esse valor seja maior que o valor de limiar (valor previamente definido que indica se uma sequência é um termo composto), então esse termo composto é introduzido na listagem de saída. Essa listagem apresenta todos os termos compostos extraídos pelo processo.

O próximo passo consiste em introduzir informação de contexto. Para isso têm de ser extraídas palavras que aparecem próximas de termos compostos. Estas palavras são dadas um valor dependendo da sua importância quando aparecem perto desses termos compostos. O critério usado é o número de termos compostos que aparece junto, ou seja, quanto maior o número, maior será a probabilidade de essa palavra estar relacionada com termos compostos. Este critério pode ser expresso na formula (25).

$$weight(w) = \frac{t(w)}{n} \quad (25)$$

em que w é a palavra de contexto, *weight(w)* é o valor que será atribuído a essa palavra, *t(w)* é o número de termos compostos que aparece junto à palavra w e n o número total de termos compostos considerados.

Para calcular o *NC-value* a informação referida anteriormente terá de ser incorporada na listagem que foi retirada pelo *C-value*. Resumindo, este método ordena a listagem retirada pelo método *C-value*, colocando no topo da listagem os termos compostos cuja certeza é maior. O método *NC-value* pode ser formalmente descrito pela fórmula (26).

$$NC-value(a) = 0.8C-value(a) + 0.2 \sum_{b \in C_a} f_a(b) weight(b) \quad (26)$$

em que a é o termo composto candidato, C_a é a listagem de palavras de contexto de a, $f_a(b)$ é a frequência de b como palavra de contexto de a e $weight(b)$ é o valor de b como palavra de contexto.

Para testar estes métodos o autor, utilizou um *corpus* de registos médicos com patologias

relacionadas com os olhos, contendo somente o diagnóstico e a descrição da doença, com um total de 810.719 palavras. A lista obtida possuía 2956 termos compostos diferentes e foi ordenada de forma decrescente do valor de *NC-Value*. A precisão obtida foi de 75% para o grupo de termos compostos do topo até ao quadragésimo termo, sendo que desta até à décima palavra foi de 36%, da décima até à quarta de 26% e as restantes de 25%, a precisão média foi de 31%.

2.3 Comparação de métodos

Como se pode verificar pelas várias medidas estatísticas, algoritmos e métodos descritos na secção anterior, existem várias formas de abordar o problema da identificação automática de termos compostos. Apresenta-se na tabela 2.1 um resumo das características principais identificadas nos sistemas e a precisão obtida. Os algoritmos apresentados, com exceção do C-value/NC-value, apresentam como principal catacterística a independência da língua e a independência de uma valor de limiar.

Sistemas que usam mais informação do que simplesmente a frequência pura, como por exemplo o uso de contexto ou o uso das categorias gramaticais dos termos compostos, melhoram a precisão de recolha de termos compostos.

O uso de estimadores estatísticos, como suporte a outros sistemas ou algoritmos, influencia a precisão destes sistemas, pois herdam as desvantagens inerentes dos estimadores estatísticos.

Considerando que, neste trabalho, se pretende extrair nomes compostos cuja estrutura sintáctica já se encontra pré-definida, pois existe uma clara predominância de um certo tipo de estruturas, o uso de filtros para a extracção dessas estruturas sintácticas deverá vir a apresentar melhores resultados.

As estruturas que se pretendem retirar são:

- Nome Adjetivo (*buraco negro*)
- Nome *de* (Determinante) Nome (*lua de mel*)

Os compostos deste último tipo podem apresentar um artigo a determinar o segundo nome (*rosa dos ventos*).

O algoritmo *LocalMaxs* apresentou melhores resultados. No entanto, é de referir que estes valores não podem ser directamente comparados porque as avaliações dos sistemas variam não só no método usado como também no *corpus* usado.

Visto de uma forma aparente, o algoritmo *LocalMaxs* foi o que apresentou melhores resultados, este é o algoritmo que foi escolhido para implementação na identificação automática de nomes compostos, como o algoritmo HELAS é semelhante ao *LocalMaxs* e como também apresentou resultados aparentemente bastante positivos, este algoritmo também será implementado.

Table 2.1: Características dos métodos e algoritmos

Método	Informação de contexto	informação sintáctica	independente da língua	independente de limiar	Precisão
LVQ			x	x	41%
HELAS		x	x	x	60%(2gramas) 80%(3gramas)
GALEMU			x	x	71%
LocalMaxs			x	x	90%(ME)
C-value/NC-value	x				31%

2.4 Critérios Sintácticos

Um nome composto apresenta restrições nas suas propriedades sintácticas, demonstrando uma certa fixidez na combinatória desse conjunto de elementos lexicais. Pelo facto de existirem diferentes tipos de nomes compostos com diferentes estruturas sintáticas, estas não respondem todas aos mesmos critérios de fixidez. As duas estruturas sintácticas aqui estudadas permitem a afloração dos seguintes critérios(Baptista 1994) de identificação do seu grau de fixidez da combinatória:

Classe *Nome Adjetivo*

- Perda de predicatividade do adjetivo;
- Variação do adjetivo em grau;
- Coordenação do adjetivo com outro adjetivo;
- Elisão do adjetivo;
- Ruptura paradigmática;
- Variação em número;

Classe *Nome de Nome*

- Inserção de elementos no grupo nominal;
- Coordenação de grupos nominais;
- Variação do determinante de N2;
- Elisão de elementos do grupo nominal;
- Ruptura paradigmática;
- Variação em número.

Se só se verificar um dos critérios, tal não é suficiente classificar uma dada combinação como nome composto. Pelo contrário é na intersecção dos vários critérios que é possível definir a sua fixidez, ou seja, quanto mais restrições forem observadas mais fixa será essa sequência de elementos lexicais.

Falamos, pois, da composição não como uma classificação binária mas sim como um fenómeno linguístico intrinsecamente escalar: A composição é uma questão de grau de fixidez (Gross 1988). Contudo, para efeitos práticos, é necessário decidir, deforma binária, sobre a inclusão ou não uma combinatória no léxico dos sistemas de PLN. Nesse sentido, a determinação destes índices de fixidez pode contribuir de forma significativa.

2.4.1 Perda de predicatividade do adjetivo

Adjetivos predicativos são adjetivos que aceitam o contexto pós-verbo copulativo, ou seja, quando um adjetivo em posição pós-nominal é um atributo do substantivo que modifica, a predicação que exerce sobre o substantivo pode ser parafraseada por uma frase com verbo copulativo (*ser* e/ou *estar*). Quando um adjetivo, que é predicativo, é combinado com certos nomes:

O Zé tomou um xarope amargo.

O Zé tomou um xarope que (era + estava) amargo.

deixa de aceitar o contexto predicativo, quando combinado com outros nomes:

O Zé tomou uma amêndoa amarga.

*O Zé tomou uma amêndoa que (era + estava) amarga.

diz-se, então, que o adjetivo perdeu a sua predicatividade, o que é um sinal de fixidez sintáctica dessa construção.

2.4.2 Variação do adjetivo em grau

Num grupo nominal livre, em que o adjetivo é predicativo, é geralmente possível fazê-lo variar em grau, mas quando o adjetivo forma com o substantivo um nome composto observam-se restrições quanto à sua variação em grau.

O Zé esqueceu-se de pôr o acento (grave + *muito grave + *gravíssimo).

Estas restrições constituem um sinal claro de fixidez existente entre os elementos da combinação nome adjetivo. Porém, existem adjetivos que não admitem qualquer tipo de variação em grau, o que faz com que este critério não seja pertinente para determinar a fixidez das combinações em que entram estes adjetivos.

2.4.3 Coordenação do adjetivo com outro adjetivo

Nos compostos com a estrutura *Nome Adjetivo*, o adjetivo forma com o substantivo uma nova unidade lexical, pelo que não é possível coordená-lo com um adjetivo livre.

A Ana organizou uma mesa redonda (e + mas) alta.

A relação entre o nome e o adjetivo, não é da mesma natureza sintáctica da que liga um adjetivo predicativo ao nome que modifica num grupo nominal livre.

A Ana comprou uma mesa redonda (e + mas) alta.

Só em condições experimentais devidamente controladas é que se pode verificar se a coordenação de dois adjetivos, modificadores do mesmo nome, é ou não possível, e, assim determinar se há ou não fixidez sintáctica na combinação.

2.4.4 Elisão do adjetivo

Em muitos nomes compostos não é possível omitir o adjetivo, sob pena de alterar a interpretação da frase em que o composto se encontra ou mesmo de a tornar inaceitável:

O Zé é a ovelha negra da família.

*O Zé é a ovelha da família.

A Ana é o braço direito do Zé.

*A Ana é o braço do Zé.

Como se pode observar, a impossibilidade de omitir o adjetivo, revela a fixidez sintáctica da sequência *Nome Adjetivo*.

2.4.5 Ruptura paradigmática

Na Classe *Nome Adjetivo*, o adjetivo pode comutar com outros adjetivos, desde que sejam respeitadas as restrições distribucionais impostas pelo substantivo.

Esta mesa é (alta + baixa + ...), (redonda + circular + quadrada + ...), (feia + estética + bonita + ...).

Cada série de adjetivos do exemplo anterior formam aquilo que habitualmente se designa por paradigma distribuicional. Num nome composto, o substantivo só se combina com um ou alguns adjetivos do paradigma distribucional em que estes se podem integrar. Esta restrição é recíproca, outros substantivos susceptíveis de pertencerem ao mesmo paradigma distribucional de *mesa* não podem comutar com este nome na combinação fixa *mesa redonda*:

O Zé e a Ana participaram em (uma secretária redonda + uma escrivaninha redonda + um estirador redondo).

Num grupo nominal livre *Nome de Nome* os elementos ficam bloqueados quando um dado elemento faz parte de um nome composto:

O Zé comprou um livro de (bolso + *algibeira).

A Ana manteve o seu (nome + *substantivo) de solteira.

Esta característica revela a fixidez sintáctica e lexical da combinação.

2.4.6 Variação em número

Em muitos nomes compostos, não se observa qualquer variação em número, pelo que o composto é ou obrigatoriamente singular, ou obrigatoriamente plural:

O povo português está neste momento a passar por (tempos difíceis +*tempo difícil).

O Zé foi condenado (à pena capital + *às penas capitais).

Alguns compostos *Nome de Nome* apresentam ou não variação em número consoante a sua construção sintática:

Os funcionários estão em (greve + *greves) de zelo.

Os funcionários fizeram (uma greve de zelo + várias greves de zelo).

Esta restrição depende, pois, em grande parte, da construção sintáctica em que o composto se encontra. Estas restrições foram consideradas como sinais de fixidez sintáctica nestas combinações.

2.4.7 Inserção de elementos no grupo nominal

Quando a combinação *Nome de Nome* forma um nome composto, não é possível inserir facultativamente um modificador específico de cada um dos substantivos, mas apenas elementos que modifiquem o nome composto na sua globalidade:

A Ana leu um livro de bolso.

*A Ana leu um livro do Zé de bolso.

A Ana leu um livro de bolso (novo + do Zé).

Quando uma dada combinação *Nome de Nome* não permite que cada um dos nomes tenha um modificador facultativo, isso é um sinal claro de fixidez sintáctica da combinação.

2.4.8 Coordenação de grupos nominais

Se o nome que está à cabeça dos dois grupos nominais for o mesmo, é possível pronominalizar a sua segunda ocorrência:

O Zé leu o livro da Ana e o livro do Pedro.

O Zé leu o livro da Ana e o do Pedro.

Mas, se um dos grupos nominais contituir um nome composto, essa pronominalização é bloqueada:

O Zé leu o livro de bolso e o livro do Pedro.

*O Zé leu o livro de bolso e o do Pedro.

Quando os dois grupos nominais são ambos nomes compostos e o primeiro nome de cada um é o mesmo, a pronominalização do substantivo repetido também é bloqueado:

A Ana discutiu com o juiz de direito e com o juiz de fora.

*A Ana discutiu com o juiz de direito e com o de fora.

2.4.9 Variação do determinante de N2

A maioria dos compostos *Nome de Nome* apresenta uma elevada fixidez quanto ao preenchimento da posição de determinante do segundo nome. Este determinante é quase sempre ou o artigo definido, ou o determinante zero (ausência de determinante).

A Ana colecciona estrelas de (*E + o) mar.

*A Ana colecciona estrelas de (um + este + esse + aquele + o seu) mar.

O Pedro tem uma estrela de (E + *o) David.

*O Pedro tem uma estrela de (um + este + esse + aquele + o seu) David.

2.4.10 Elisão de elementos do grupo nominal

Os nomes compostos *Nome de Nome* não admitem a omissão do primeiro nome, no entanto um reduzido número permite que não só o primeiro nome seja omitido, mas também a preposição *de* e o eventual determinante do segundo nome:

O Zé tomou um vinho do Porto.

O Zé tomou um Porto.

O determinante do composto mantém-se na variante elíptica. Se se tratar de um numeral, o segundo nome passa a plural:

Bebemos dois vinhos (do Porto + da Madeira) diferentes.

Bebemos dois (Portos + Madeiras) diferentes.

Em muitos compostos *Nome de Nome* não é possível o apagamento do complemento de *N2*, já que este forma com *N1* uma unidade lexical composta:

O Zé e a Ana estão em lua de mel.

*O Zé e a Ana estão em lua.

Este trabalho procura usar a rica informação linguística disponibilizada pela cadeia de processamento STRING por forma a tentar validar estes critérios linguísticos dentro das limitações da informação disponível. Tal será descrito nas secções 3.1.1 e 3.2.1.

Chapter 3

Estratégia e Implementação

Este capítulo descreve a construção dos filtros necessários para a procura das estruturas pretendidas: *Nome Adjetivo* e *Nome de Nome*.

Também é apresentado as soluções implementadas para a identificação dos cirtérios sintáticos destas mesmas estruturas.

3.1 Estrutura *Nome Adjetivo*

O processo para a identificação de termos compostos com a estrutura *nome adjetivo* é dividido em duas fases. Primeiro, é construído e aplicado um filtro que percorre as árvores xml resultantes do processamento do *corpus* CETEMPúblico pela cadeia de processamento STRING. Este filtro é um programa feito na linguagem de programação Java, fazendo uso do paradigma MapReduce. O programa funciona da seguinte forma:

- Verifica se o nó READING tem o atributo "pos" igual a NOUN, ou seja, se um dado lema é um nome;
- Para o nó identificado anteriormente, verifica-se se existe um nó FEATURE com o atributo "attribute" igual a PROPER. Este atributo indica se um nome é um nome próprio. Um termo composto não é formado, de um modo geral, por nomes próprios, sendo estes então descartados caso se verifiquem.
- Partindo do nó READING anteriormente encontrado, verifica-se se o nó READING adjacente a este possui o atributo "pos" igual a ADJ ou PASTPART, ou seja, se a palavra é um adjetivo ou um particípio passado. Os particípios passados comportam-se de forma semelhante a adjetivos, ocorrendo como modificadores adnominais e concordando com o nome que modificam em género e número; por simplicidade, tratamo-los como adjetivos;

- Para este adjetivo, verifica-se se existe um nó FEATURE com o atributo "attribute" igual a GENT. Este atributo indica se um adjetivo é um adjetivo gentílico, ou seja, um adjetivo que designa um indivíduo em função do seu local de nascimento ou residência (*asiático*, *londrino*, *cipriota*, *português*, etc...). De modo geral, um termo composto também não é formado por este tipo de adjetivos, pelo que estes candidatos serão descartados. A determinação de compostos com este tipo de adjetivos deve ser feito de um modo autónomo (*pastor alemão*, *chave inglesa*, *tortura chinesa*, *pontualidade britânica*, *calçada portuguesa*, etc...), noutro momento, pois verificou-se que introduzem demasiado ruído no processo de recolha de candidatos;

- Finalmente, o padrão encontrado é enviado para o REDUCER e o processo é repetido para as outras árvores do *corpus* processado.

O resultado obtido é uma lista com todos os candidatos encontrados no *corpus*, seguido do seu número de ocorrências. A cadeia de processamento STRING já identifica cerca de 22.000 nomes compostos diferentes com esta estrutura. Estes termos são identificados como um único token NOUN, pelo que não serão capturados pelo filtro aqui proposto.

Em segundo lugar, são aplicados os métodos e algoritmos descritos no capítulo anterior. No entanto, é necessário retirar do corpus outro tipo de informação, nomeadamente os n-gramas de palavras que contêm o candidato identificado, assim como os n-gramas das categorias gramaticais.

Os métodos estatísticos apresentados no capítulo anterior são usados para calcular bigramas, com a exceção do Mutal Expectation. Para os dados poderem ser processados pelos algoritmos é necessário o cálculo de trigramas, pelo que alguns dos métodos foram normalizados, nomeadamente o coeficiente de Dice, Specific Mutual Information, ϕ^2, Symmetric Conditional Probability, e o Loglikelihood Ratio.

Somente os candidatos *nome adjetivo* identificados pelo filtro que tivessem um número de ocorrências superior a cinco foram processados, pois os métodos utilizados neste trabalho tornam-se incertos quando lidam com eventos raros (Pecina & Schlesinger 2006). Para os candidatos *nome de nome* isto não se aplicou, de forma a poder-se verificar esta afirmação, visto que todos os algoritmos descritos na capítulo 2.2 não têm qualquer descrição quanto a este factor.

3.1.1 Critérios Sintácticos

Esta secção descreve a implementação usada para a aplicação de cada um dos critérios sintáticos para a estrutura *Nome Adjetivo* descrito na secção 2.4.

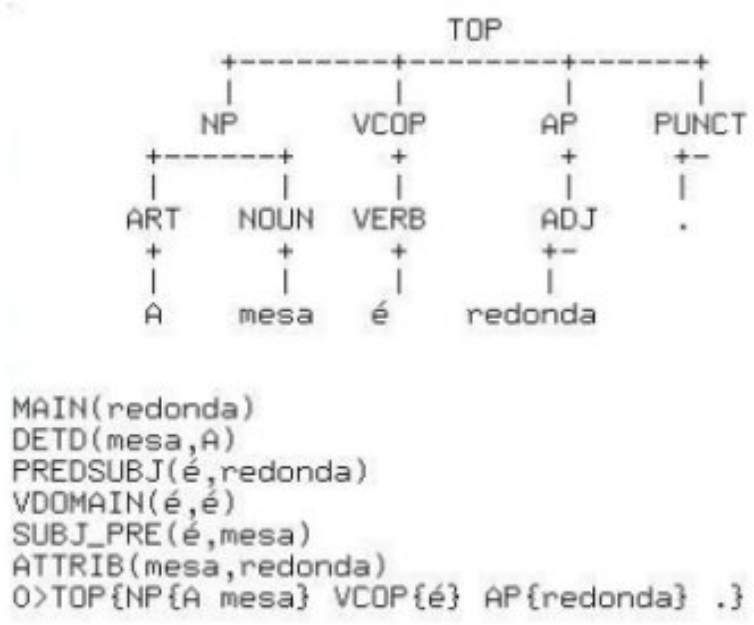


Figure 3.1: A frase "*A mesa é redonda.*", processada pelo XIP.

3.1.1.1 Perda de predicatividade do adjetivo

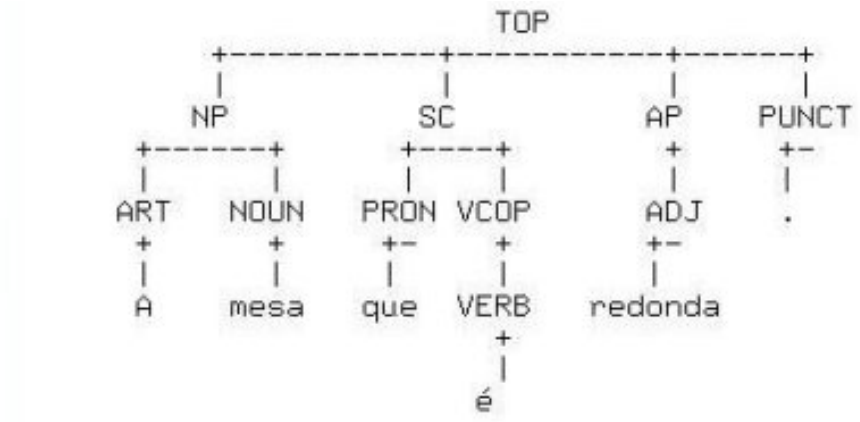
Para a aplicação deste critério foi usado a informação resultante do processamento do corpus CETEMPúblico pela cadeia de processamento STRING, nomeadamente a rede de dependências produzidas pelo XIP.

Para a tentativa de determinação desta propriedade, a ideia é tentar verificar se o adjetivo aparece no *corpus* em contexto predicativo tendo como sujeito o mesmo nome. Tal ocorre em frases simples (ou atributivas) como ilustrado na figura 3.1 ou no quadro de uma oração relativa, como se mostra na figura 3.2.

Na figura 3.1 apresenta-se um exemplo em que foi extraída pela cadeia uma dependência ATTRIB entre o nome e o adjetivo enquanto que na figura 3.2 esta dependência se estabelece entre o pronome relativo *que* e o adjetivo. Para esta última situação é necessário verificar a existência de outras duas dependências: a dependência PREDSUBJ, que relaciona o verbo copulativo com o adjetivo, e a dependência ANTECEDENT_RELAT, que relaciona o nome antecedente com o pronome relativo.

O programa para reconhecer estes padrões funciona da seguinte forma:

- Ao percorrer a árvore xml da frase, são colocados numa lista todas as palavras da frase com as suas respectivas categorias gramaticais;
- Procuram-se as dependências ATTRIB, PREDSUBJ e ANTECEDENT_RELAT e




```
MAIN(mesa)
DETD(mesa,A)
PREDSUBJ(é,redonda)
VDOMAIN(é,é)
MOD_POST_RELAT(mesa,é)
SUBJ_PRE(é,que)
INTROD-AUX_RELAT(que,é)
INTROD_RELAT(que,é)
ANTECEDENT_RELAT(mesa,que)
QBOUNDARY_RELAT(que,que,redonda)
ATTRIB(que,redonda)
0>TOP{NP{A mesa} SC{que VCOP{é}} AP{redonda} .}
```

Figure 3.2: A frase "*A mesa que é redonda.*", processada pelo XIP.

guardam-se os seus pares de palavras;

- Verifica-se se nos pares ATTRIB a primeira palavra é um NOUN e a segunda um ADJ ou PASTPART; se existir, este resultado é enviado para o REDUCER;

- Verifica-se se a primeira palavra do par ANTECEDENT_RELAT é um NOUN; se isto ocorrer, verifica-se se primeira palavra dos ATTRIB é igual à segunda palavra de um par ANTECEDENT_RELAT; verifica-se ainda se a segunda palavra do ATTRIB é um ADJ ou PASTPART e se esta palavra é igual à segunda palavra de um par PREDSUBJ; se se encontrar estas relações, o par nome adjetivo é então enviado para o REDUCER;

- O processo é repetido para cada árvore xml do ficheiro de entrada.

O resultado deste processo é uma lista de pares *nome-adjetivo* que não perdem a predicatividade. No entanto, o que se pretende é uma lista de pares cujos adjetivos perdem de facto a predicatividade. Assim, este resultado é cruzado com a lista de pares que foi retirada pelo filtro produzido e descrito na secção 3.1, sendo retirado dessa lista todos os pares *nome-adjetivo* encontrados pelo processo aqui descrito, o resultado final é uma lista de pares *nome adjetivo* cujos adjetivos não aparecem no *corpus* em contexto predicativo.

Este resultado é apenas uma aproximação ao que se pretende, visto estes critérios terem sido construídos para identificação manual de termos compostos e requererem conhecimento empírico. Com este resultado não podemos afirmar que, para estes candidatos a nomes compostos, o adjetivo perde a sua predicatividade quando combinado com aquele nome, mas apenas se pode dizer que não foi encontrado no *corpus* nenhum exemplo em que o adjetivo ocorresse em contexto predicativo.

3.1.1.2 Variação do adjetivo em grau

O processo para aplicação deste critério é semelhante ao processo utilizado para o critério anterior. São procuradas no *corpus* situações em que, num par *nome-adjetivo*, o adjetivo varie em grau, cruzando estes resultados com os resultados do filtro para nomes adjetivos, resultando numa lista de pares *nome-adjetivo* em que não foi encontrado no *corpus* casos em que o adjetivo tenha apresentado variação em grau.

O programa usado para reconhecer estes padrões funciona da seguinte forma:

- Tal como descrito na secção 3.1, o programa identifica primeiro os candidatos constituídos por um par *nome-adjetivo* (ou particípio passado), descartando os casos dos nomes próprios e dos adjetivos gentílicos

- No nó correspondente ao adjetivo, verifica-se ainda se este tem um nó FEATURE com o atributo "attribute" igual a SINT; este atributo indica que o adjetivo se encontra no grau superlativo absoluto sintético;
- Se o nó adjacente ao nome for igual a ADV, ou seja, um advérbio, verifica-se se o nó READING seguinte é então um ADJ ou PASTPART;
- Em qualquer um dos casos, é enviado para o REDUCER o par *nome-adjetivo* assim encontrado;
- O processo é repetido para cada árvore do ficheiro de entrada.

O resultado deste processo é uma lista de pares *nome-adjetivo* cujo adjetivo tenha apresentado variação em grau no *corpus*. Este resultado é então cruzado com a lista de pares *nome-adjetivo* produzido pelo programa descrito na secçaõ 3.1, sendo retirados dessa lista todos os pares encontrados pelo processo descrito aqui. O resultado final é uma lista de pares *nome-adjetivo* cujo adjetivo não apresentou no *corpus* qualquer variação em grau.

Como foi referido para o critério anterior, este resultado é apenas uma aproximação ao que se pretende pelas mesmas razões enunciadas, não podemos afirmar com total certeza que o adjetivo nunca varie em grau para o nome que modifica, simplesmente podemos dizer que não foi encontrado um exemplo em contrário.

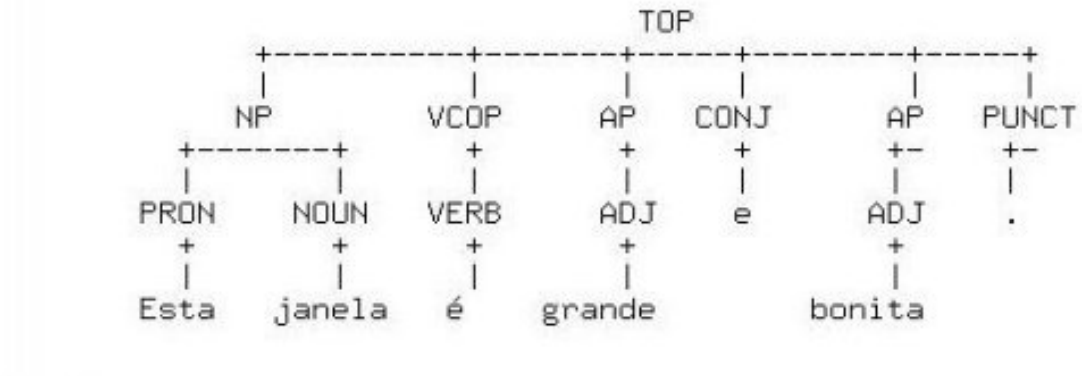
3.1.1.3 Coordenação do adjetivo com outro adjetivo

Os padrões que se pretende encontrar para a validação deste critério são situações em que, para um dado par *nome-adjetivo*, o adjetivo se encontre coordenado com outro adjetivo. As figuras 3.3 e 3.4 ilustram exemplos destas situações.

Em ambas as figuras são produzidas duas relações de coordenação (COORD) entre a conjunção coordenativa e os adjetivos. No entanto, as relações destes adjetivos com o nome são diferentes: No primeiro, caso são feitas duas relações de atributo (ATTRIB e ATTRIB_ANAPH0); no segundo caso, são obtidas duas relações de modificador (MOD_POST). É necessário reconhecer estes padrões.

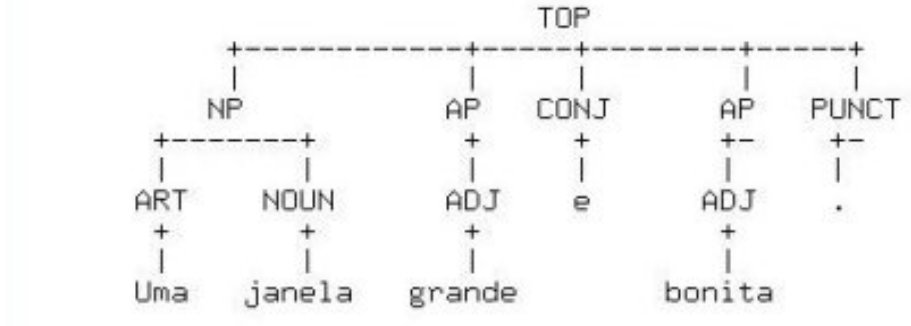
O programa para reconhecer estes padrões funciona da seguinte forma:

- Ao percorrer a árvore xml da frase, são colocadas numa lista todas as palavras da frase com as suas respectivas categorias gramaticais;
- Procuram-se as dependências COORD, ATTRIB, ATTRIB_ANAPH0 e MOD_POST e guardam-se os seus pares de palavras;




```
MAIN(e)
DETD(janela,Esta)
COORD(e,grande)
COORD(e,bonita)
PREDSUBJ(é,grande)
VDOMAIN(é,é)
SUBJ_PRE(é,janela)
ATTRIB(janela,grande)
ATTRIB_ANAPHO(janela,bonita)
0>TOP{NP{Esta janela} VCOP{é} AP{grande} e AP{bonita} .}
```

Figure 3.3: A frase "*Esta janela é grande e bonita.*", processada pelo XIP.




```
MAIN(janela)
QUANTD(janela,Uma)
COORD(e,grande)
COORD(e,bonita)
MOD_POST(janela,grande)
MOD_POST(janela,bonita)
0>TOP{NP{Uma janela} AP{grande} e AP{bonita} .}
```

Figure 3.4: A frase "*Uma janela grande e bonita.*", processada pelo XIP.

- Verifica-se se nos pares COORD a segunda palavra é um ADJ ou PASTPART; se forem, verifica-se se existe nos pares ATTRIB e MOD_POST uma segunda palavra igual à segunda palavra da dependência COORD e se nos pares ATTRIB_ANAPH0 e MOD_POST existe uma segunda palavra igual à segunda palavra da outra dependência COORD;
- Se as relações forem encontradas, verifica-se se a primeira palavra dos pares ATTRIB e ATTRIB_ANAPH0 encontrados são NOUN; se assim for, é enviado para o REDUCER o par nome-adjetivo capturado pelas dependências ATTRIB ou MOD_POST;
- O processo é repetido para cada árvore do ficheiro de entrada.

A aplicação do critério é semelhante à dos critérios descritos anteriormente: é feito um cruzamento com a lista de pares *nome-adjetivo* produzido pelo filtro, resultando numa lista de pares, para os quais não se encontrou no *corpus* uma situação em que o adjetivo estivesse coordenado com outro adjetivo como modificadores daquele nome.

Como foi referido para os critérios anteriores, este resultado é apenas uma aproximação ao que se pretende pelas mesmas razões enunciadas, isto é, não podemos afirmar com total certeza que o adjetivo nunca se poderá coordenar com outro adjetivo quando é modificador de um certo nome; simplesmente podemos dizer que não foi encontrado qualquer exemplo em tal acontecia.

3.1.1.4 Elisão do adjetivo

A abordagem principal a este critério centra-se na comparação de contextos, ou seja se, para um par *nome-adjetivo* com um certo contexto é encontrado um nome com precisamente o mesmo contexto, então podemos concluir que é possível omitir o adjetivo para esse nome. O objectivo é identificar os casos em que não é possível esta omissão. Para a identificação do "contexto" utilizam-se as dependências sintácticas extraídas pelo STRING embora várias dependências pudessem ser aqui utilizadas, decidiu-se utilizar as 3 mais gerais: SUBJ (sujeito), CDIR (complemento directo) e MOD (modificador).

O processo para a identificação funciona da seguinte forma:

- Retira-se do *corpus* os pares nome-adjetivo e para esses pares procura-se nos nós DEPENDENCY as tags CDIR, SUBJ e MOD cuja segunda palavra corresponda ao nome do par encontrado, esta informação é enviada para o REDUCER;
- Retira-se do *corpus* os nomes que não se encontrem seguidos de um adjetivo e procura-se nos nós DEPENDENCY as tags CDIR, SUBJ e MOD cuja segunda palavra seja igual ao nome encontrado, esta informação é enviada para o REDUCER;
- As duas listas produzidas pelos dois passos anteriores são processados por um programa que verifica quais os pares *nome-adjetivo* que ocorrem num contexto igual em que esse nome ocorre isolado.

3.1.1.5 Ruptura paradigmática

Para a aplicação deste critério (secção 2.4.5), pretende-se verificar no *corpus* se, para o nome de um par *nome-adjetivo*, se verifica uma ruptura distribucional quanto ao adjetivo que com ele se combina, isto é, sendo dado o paradigma distribucional do adjetivo, se se verifica que o adjetivo da combinação é o único dentro do seu paradigma distribucional que ocorre com aquele nome. Inversamente, o mesmo critério também se aplica ao nome do mesmo par. Se se tratar de um composto, o nome não deverá variar com outros nomes do mesmo paradigma distribucional na combinação com aquele adjetivo do par candidato.

Ora, a cadeia de processamento STRING não identifica o paradigma ou paradigmas distribucionais em que se poderia integrar cada palavra. Por essa razão, na aplicação deste critério, avalia-se apenas, e de forma aproximativa, a coocorrência de nomes e adjetivos, considerando o par candidato em relação ao conjunto de todos os nomes e adjetivos com que aparecem combinados no *corpus*.

Assim, numa primeira abordagem, calculou-se a ruptura distribucional relativamente ao adjetivo dividindo o número de ocorrências do par candidato pelo número total de pares *nome-adjetivo* em que ocorre o nome do par candidato. Inversamente, para o substantivo, calculou-se o rácio do par candidato sobre todos os pares *nome-adjetivo* em que o adjetivo é o mesmo do par candidato. Considerou-se que havia ruptura distribucional se um dos dois rácios fosse igual ou superior a 0,75. Para estes cálculos usaram-se os dados obtidos pelo filtro de identificação das estruturas *nome-adjetivo* descritas na secção 3.1.

Numa segunda abordagem, aplicaram-se outras medidas estatísticas, habitualmente usadas para avaliar o grau de coesão de um diagrama isto é, que indicasse a probabilidade de um certo par nome-adjetivo ocorrer no *corpus* usando somente a informação relativamente aos pares previamente encontrados. As medidas usadas foram o Pearson's χ^2 e o Student t test (Manning & Schütze 1999).

3.1.1.6 Variação em número

A identificação deste critério é feita de uma forma semelhante ao critério descrito anteriormente. É necessário comparar as ocorrências plural/singular dos candidatos com as ocorrências plural/singular dos seus nomes. Antes de se proceder ao cálculo dos rácios foi necessário retirar informação acerca destas ocorrências.

Foi utilizado o programa descrito na secção 3.1 com uma ligeira alteração. Nos tokens identificados como NOUN e ADJ ou PASTPART foi verificado se a tag FEATURE tem um atributo "attribute" igual a SG ou PL, que indica se está no singular ou plural, respectivamente. É então enviado para o REDUCER o par nome-adjetivo e o respectivo valor em número, obtendo

no final uma lista de pares nome adjetivo com o respectivo número de ocorrências no plural e no singular.

Para proceder à contagem do valor em número dos nomes, foi somente necessário produzir um programa que verifica a FEATURE referente ao número da palavra, para todos o nomes do *corpus*, obtendo no final uma lista de nomes com o respectivo número de ocorrências no singular e no plural.

Após obtermos estas informações, os dados são submetidos a um programa que calcula:

- o rácio do número de ocorrências no singular (ou no plural) do par candidato sobre o total de ocorrências do par: $num(NA) = \frac{sg(NA)}{f(NA)}$
- o rácio do número de ocorrências no singular (ou no plural) do nome do par candidato sobre o total de ocorrências desse nome no *corpus*: $num(N) = \frac{sg(N)}{f(N)}$

Apenas foram considerados os pares candidatos em que $num(NA) \geq 0,9$. De seguida, verificou-se a diferença entre os dois rácios: $num(NA) - num(N)$. Se essa diferença for reduzida, isso quer dizer que não se observaram alterações na propriedade de variação em número do nome quando este se encontra numa dada combinação *nome-adjetivo*. Foram testados vários valores para esta diferença, tendo a melhor performance sido atingida com um valor de 0,2.

3.2 Estrutura *Nome de Nome*

A identificação dos nomes compostos com a estrutura *nome-de-nome* é semelhante à apresentada para a estrutura *nome adjetivo*. Foi construído um filtro que foi aplicado às árvores de xml resultantes do processamento do *corpus* CETEMPúblico pela cadeia de processamento STRING. O programa aplicado funciona da seguinte forma:

- Verifica-se se o nó READING tem o atributo "'pos"' igual a NOUN, ou seja, se um dado lema é um nome;
- Para o nó identificado anteriormente verifica-se se existe um nó FEATURE com o atributo "attribute" igual a PROPER (nomes próprios), se for encontrado o atributo, esse nó é então descartado;
- Partindo do nó READING anteriormente encontrado, verifica-se se o nó READING adjacente a este possui o atributo "'lemma"' igual à palavra *de*;
- Depois, verifica-se se o nó READING adjacente ao identificado no passo anterior possui o atributo "pos" igual a NOUN;

- Para o nó identificado anteriormente verifica-se se existe um nó FEATURE com o atributo "attribute" igual a PROPER; se for encontrado este atributo é então descartado. A determinação de compostos com este tipo de nomes deve ser feito de um modo autónomo (constante de Planck, teorema de Pitágoras, tinta da China, etc...), noutro momento, pois verificou-se que introduzem demasiado ruído no processo de recolha de candidatos;
- As estruturas encontradas são enviadas para o REDUCER;
- O processo é repetido para cada árvore do ficheiro de entrada.

O resultado obtido é uma lista com todos os candidatos encontrados no *corpus*, seguido do respectivo número de ocorrências. A cadeia de processamento STRING também já identifica à partida alguns termos com esta estrutura, sendo estes termos identificados como um único token NOUN. Estes casos não serão, pois, identificados pelo filtro aqui apresentado.

Também é necessário retirar mais informação para se poder aplicar os métodos e algoritmos estudados, nomeadamente os unigramas, bigramas, trigramas e os N+1 gramas que contêm o candidato identificado das palavras e das categorias gramaticais.

Foram usados os métodos que tinham sido normalizados para a identificação da estrutura nome adjetivo assim como os algoritmos LocalMaxs e HELAS (secções 2.2.2 e 2.2.4 respectivamente).

3.2.1 Critérios Sintácticos

Esta secção descreve a implementação usada para a identificação de cada um dos critérios sintácticos para a estrutura *Nome de Nome*.

3.2.1.1 Inserção de elementos no grupo nominal

Os modificadores de nomes aqui considerados são os adjetivos que podem ser inseridos dentro da estrutura *nome de nome*, o que, no caso do termo composto, não sucede. Para determinar este tipo de restrição, será necessário proceder à identificação no *corpus* de estruturas com os seguintes padrões:

- N1 ADJ de N2;
- N1 de ADJ N2;
- N1 ADJ de ADJ N2.

O programa que procede à identificação destes padrões é uma variante do programa descrito na secção 3.2 e funciona da seguinte forma:

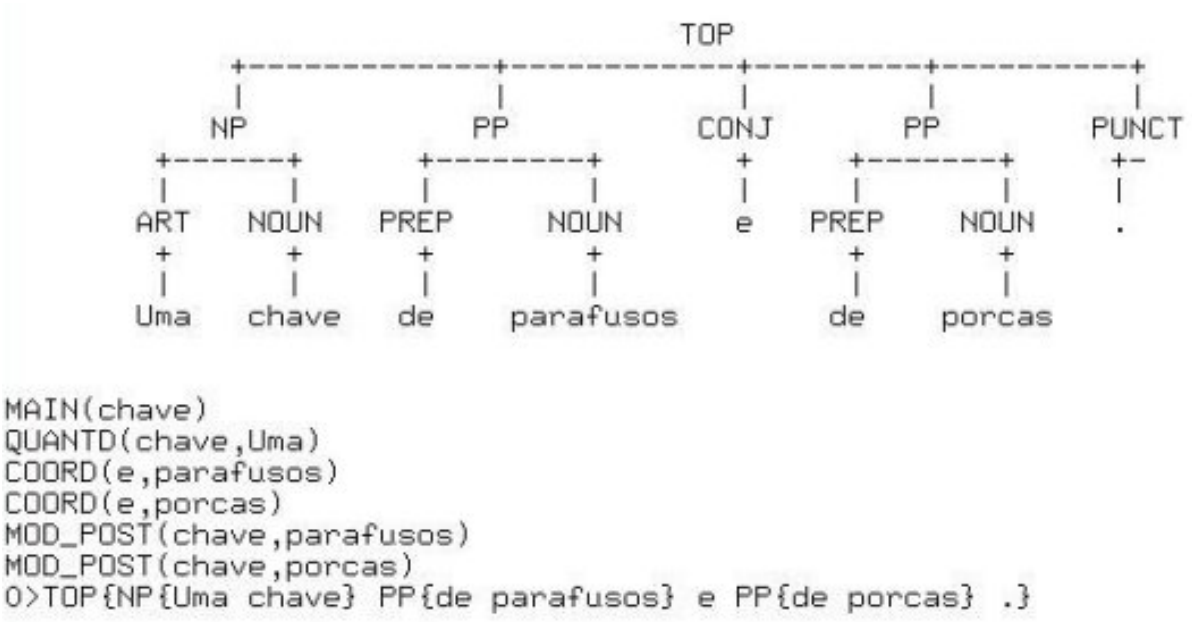


Figure 3.5: A expressão "*Uma chave de parafusos e de porcas.*", processada pelo XIP.

- Tal como descrito na secção anterior, o programa identifica os candidatos *nome de nome* que não sejam nomes próprios;
- Verifica-se se existe um nó ADJ ou PASTPART entre os nomes e a preposição;
- É enviado para o REDUCER os candidatos *nome de nome* encontrados;
- Este processo é repetido para todas as árvores do ficheiro de entrada.

O resultado é uma lista de candidatos com a estrutura *nome de nome* em que se observam modificadores adjectivais inseridos nas posições sintácticas acima referidas. Esta lista é depois processada por um programa que cruza estes resultados com a lista de estruturas encontradas pelo filtro descrito na secção 3.2, produzindo uma lista de candidatos *nome de nome* que não se encontrem na lista produzida pelo programa descrito acima mas que foram encontrados pelo filtro.

3.2.1.2 Coordenação de grupos nominais

Os padrões que se pretende encontrar para este critério são situações em que, para uma dada estrutura *nome de nome*, o segundo nome se encontra coordenado com outro nome. As figuras 3.5 e 3.6 ilustram estas situações.

No exemplo 3.5 são extraídas as relações de coordenação (COORD) entre a conjunção *e* e os dois nomes dos complementos *de N*. Tal permite, então, a obtenção das dependências de

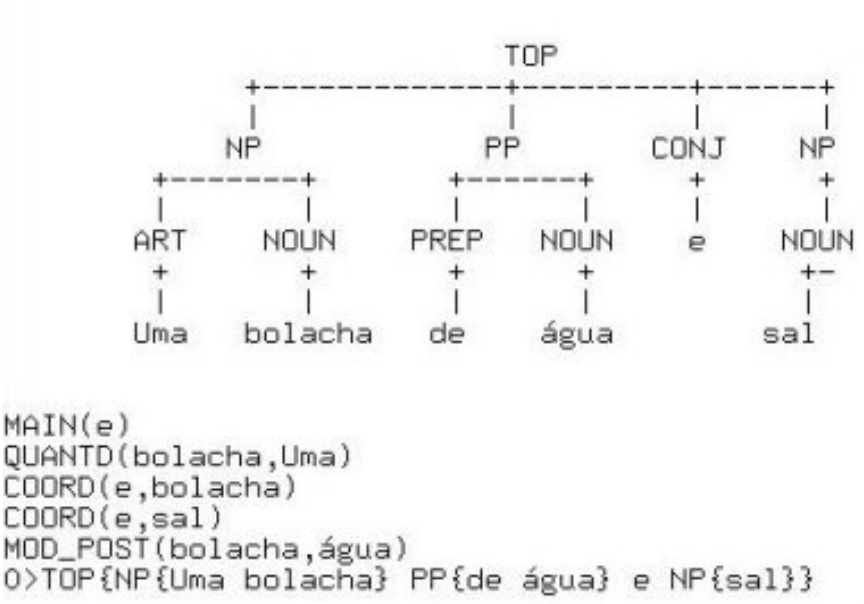


Figure 3.6: A expressão "*Uma bolacha de água e sal.*", processada pelo XIP.

modificador (MOD) entre estes nomes e o nome que é cabeça do grupo nominal. No exemplo 3.6, obtêm-se as mesmas dependências de coordenação mas apenas se extrai a dependência de MOD. Tal deve-se ao facto de se ter considerado que sem informação adicional não era possível determinar o escopo da conjunção, tendo a decisão sido adiada para mais tarde.

O programa para reconhecer estes padrões funciona da seguinte forma:

- Ao percorrer a árvore xml da frase, são colocadas numa lista todas as palavras da frase com as suas respectivas categorias gramaticais;
- São colocadas numa lista todas as estruturas *nome de nome* encontrados na frase;
- Nos nós DEPENDENCY são procuradas as tags COORD e MOD_POST e guardados os seus pares de palavras;
- Verifica-se se nos pares COORD a segunda palavra é um NOUN; se forem, verifica-se se existem duas situações em que um MOD_POST e um COORD tenham a segunda palavra igual e que desses MOD_POST a primeira palavra pertença a uma das estruturas *nome de nome* encontradas na frase;
- Também é verificado se existe um COORD em que a segunda palavra pertença a uma das estruturas *nome de nome* encontradas; se for encontrado, procura-se por um MOD_POST cuja primeira palavra seja igual a esta palavra e que a sua segunda palavra pertença também à estrutura *nome de nome* encontrada;

- Nas situações verificadas, a respectiva estrutura é enviada para o REDUCER;
- O processo é repetido para cada árvore do ficheiro de entrada.

Esta lista é depois processada por um programa que cruza estes resultados com a lista de estruturas encontradas pelo filtro descrito na secção 3.2, daí resultando uma lista de estruturas *nome de nome* que não se encontram na lista produzida pelo programa descrito acima mas que foram encontrados pelo filtro.

3.2.1.3 Variação do determinante de N2

Um nome composto *nome de nome* apresenta geralmente ou determinante zero (ausência de determinate) ou o artigo definido a determinar o segundo nome da combinatória. Para a verificação deste critério a estratégia seguida consistiu em determinar, para cada par candidato o rácio do número de ocorrências da combinatória (com artigo ou sem determinante) sobre todas as ocorrências da mesma estrutura com quaisquer outros determinantes.

O programa é uma variação do programa descrito na secção 3.2 e funciona da seguinte forma:

- Tal como descrito na secção 3.2, o programa identifica os candidatos *nome de nome* que não sejam nomes próprios;
- Verifica se o nó adjacente ao lema *de* é igual a ART, ou seja, o token é um artigo;
- É enviado para o REDUCER a estrutura *nome de nome* encontrada;
- Este processo é repetido para todas as árvores do ficheiro de entrada.

A lista resultante deste programa são todas as estruturas *nome de nome* em que o segundo nome aparece determinado por um artigo definido.

É também necessário recolher as situações em que o segundo nome se encontra determinado por todos os outros casos, o programa também é uma variação do programa descrito na secção 3.2 e funciona da seguinte forma:

- Tal como descrito na secção 3.2, o programa identifica os candidatos nome de nome que não sejam nomes próprios;
- Verifica se o nó adjacente ao lemma *de* é diferente de ART ou PUNCT, ou seja, verifica se não é um artigo definido ou uma pontuação;
- É enviado para o REDUCER a estrutura nome de nome encontrada;

- Este processo é repetido para todas as árvores do ficheiro de entrada.

A lista resultante deste programa são todas as estruturas *nome de nome* em que o segundo nome aparece determinado por um determinante que não é o artigo definido. Estes dados são então processados por um programa que calcula o rácio entre o número de ocorrências da combinatória candidata e o total de ocorrências da expressão envolvendo os mesmos nomes mas com outros determinantes, ou seja, todas as instâncias *nome de* (*det*) *nome* em que *det* é diferente do determinante do candidato. Se este valor for superior a um dado limiar, considera-se que o critério se aplica. O limiar utilizado foi de 0,75. Se no *corpus* não se tiver observado qualquer variação do determinante, o candidato é imediatamente classificado como verificando este critério.

3.2.1.4 Elisão de elementos do grupo nominal

A solução produzida para este critério é idêntica à estratégia usada para a elisão do adjetivo na estrutura nome adjetivo. É necessário procurar por contextos com a estrutura pretendida e compará-los com os contextos dos nomes que não se encontram nesta estrutura, ou seja em que o primeiro nome não apresenta o complemento *de N*. As dependências usadas para comparação foram também as dependências de sujeito (SUBJ), complemento directo (CDIR) e modificador (MOD).

O processo para a identificação funciona da seguinte forma:

- Retira-se do *corpus* as estruturas *nome de nome* e para essas estruturas procura-se nos nós DEPENDENCY as tags CDIR, SUBJ e MOD cuja segunda palavra corresponda ao primeiro nome da estrutura encontrada, sendo depois esta informação enviada para o REDUCER;
- Retira-se do *corpus* os nomes que não se encontrem numa estrutura *nome de nome* e procura-se nos nós DEPENDENCY as tags CDIR, SUBJ e MOD cuja segunda palavra seja igual ao nome encontrado, esta informação é enviada para o REDUCER;
- As duas listas produzidas pelos dois passos anteriores são então processados por um programa que verifica quais as estruturas *nome de nome* para cujo contexto, definido em termos do conjunto de dependências acima referidas, foi possível encontrar ocorrências do primeiro nome da combinatória sem a presença do complemento *de N*. O resultado é uma lista de termos *nome de nome* para os quais não foram encontrados contextos iguais.

3.2.1.5 Ruptura paradigmática

Este critério segue uma solução semelhante à proposta para o mesmo critério na estrutura *nome adjetivo* (secção 3.1.1.5). Pretende-se verificar no *corpus* se, sendo dado o primeiro nome

da estrutura *nome de nome*, não existe grande variação do segundo nome dentro do mesmo paradigma distribucional, ou se, sendo dado o segundo nome da mesma estrutura, não existe grande variação do primeiro nome dentro do mesmo paradigma distribucional.

Como já tinha sido referido para a estrutura *nome adjetivo*, a cadeia de processamento STRING não retira qualquer tipo de informação relativamente ao paradigma distribucional de uma palavra. Assim, todos os nomes foram considerados como tendo o mesmo paradigma distribucional.

Para a classificação deste critério, usou-se as mesmas abordagens propostas para a estrutura nome-adjetivo, é calculado o rácio de um dos nomes relativamente ao outro nome, usando o mesmo valor de limiar de 0,75. Os dados usados foram os resultantes da lista de *nome de nome* encontrados pelo filtro de identificação desta estrutura.

Também foram usadas as mesmas medidas estatísticas para avaliar o grau de coesão entre os nomes dos candidatos que tinham sido usadas para a estrutura nome adjetivo, nomeadamente Pearson's χ^2 e o Student t test.

3.2.1.6 Variação em número

O processo de determinação deste critério para a estrutura *nome de nome* é um pouco diferente do que foi proposto para a estrutura *nome adjetivo*. Na estrutura *nome de nome*, a variação em número pode ocorrer mas, de um modo geral, apenas o primeiro nome flexiona em número mantendo-se o segundo nome invariável, como por exemplo:

O Pedro comprou um livro de bolso.

O Pedro comprou vários livros de bolso.

*O Pedro comprou vários livros de bolsos.

*O Pedro comprou um livro de bolsos.

Assim, apenas é necessário verificar se, para os candidatos encontrados existem casos em que o primeiro nome varia em número.

Foi utilizado o programa descrito na secção 3.2 com uma ligeira alteração. Nos tokens identificados como NOUN é verificado se a tag FEATURE tem um atributo "attribute" igual a SG ou PL, que indica se o nome está no singular ou plural respectivamente. É depois enviado para o REDUCER o candidato com o valor em número de cada nome que o compõe, obtendo-se no final uma lista de candidatos com o número de ocorrências dos valores em número de cada nome.

Foi usado o programa descrito para o mesmo critério na estrutura nome adjetivo que faz as contagens dos valores em número dos nomes no *corpus*.

Com estas informações, estes dados são submetidos a um programa que verifica se um candidato tem ocorrências em que o primeiro nome se encontra no singular ou plural e que o segundo nome encontra-se sempre no singular; se isto ocorrer, então o candidato é classificado como apresentando este critério.

Chapter 4

Avaliação e Resultados

4.1 Avaliação

Esta secção descreve os procedimentos adoptados para verificar se os filtros apresentados nas secções 3.1 e 3.2 funcionam correctamente e da forma pretendida. Apresenta-se também o conjunto de procedimentos utilizados para verificar e validar os métodos e programas de aplicação dos critérios sintáticos, descritos nas secções 3.1.1 e 3.2.1.

4.1.1 Filtros *Nome Adjetivo* e *Nome de Nome*

Para a validação dos filtros constituiu-se um texto de input que foi depois verificado manualmente tendo em vista a identificação dos padrões *nome adjetivo* e *nome de nome* pretendidos. O texto é constituído por 100 frases extraídas aleatoriamente do *corpus* CETEMPúblico. Nele se observaram 101 padrões *nome adjetivo* (99 padrões diferentes) e 62 padrões *nome de nome* (todos diferentes.)

O texto foi então processado pela cadeia de processamento STRING e ao resultado foram aplicados os programas de extracção dos padrões pretendidos. Os resultados foram comparados com a verificação manual e confirmou-se que eram equivalentes, confirmando igualmente o correcto funcionamento dos filtros.

4.1.2 Métodos e algoritmos

Para a validação dos métodos e algoritmos, foi produzido manualmente uma lista de unigramas, bigramas, termos compostos candidatos, (N+1)-gramas e as suas respectivas categorias gramaticais, sendo depois calculados manualmente os valores das medidas estatísticas e os resultados dos algoritmos.

Calcular manualmente uma quantidade grande de dados é um processo bastante moroso e de uma extrema dificuldade. Assim as listas produzidas compõem-se de um total de 11 instâncias de padrões, sendo 6 desses diferentes. Estas listas foram processadas pelos métodos e algoritmos apresentados e foram comparados com os que tinham sido obtidos manualmente. Os resultados foram equivalentes.

É necessário também uma avaliação para a classificação de nomes compostos no *corpus* CETEMPúblico. Como se trata de um *corpus* de dimensões muito grandes, medir os resultados em termos de *recall* é impossível, simplesmente porque retirar manualmente todos os nomes compostos com as estruturas pretendidas de um *corpus* constituído por cerca de 190 milhões de palavras não é exequível.

Assim, o método principal de avaliação é o da *precisão*, que é medido da seguinte forma:

$$\text{Precisão} = \frac{\text{número de candidatos classificados correctamente como nome composto}}{\text{número de candidatos classificados como nome composto}} \quad (27)$$

No entanto, devido ao número demasiado elevado de candidatos, a avaliação de precisão dos métodos de classificação usados foi limitada a uma amostra aleatória estratificada com base na frequência de 1000 candidatos.

4.1.3 Critérios Sintácticos

Como foi referido na secção anterior, o conjunto total de combinatórias candidatos classificadas pelos métodos aqui utilizados é demasiado grande para poder ser verificado manualmente. Assim a avaliação foi limitada à amostra aleatória referida na secção anterior. No entanto, o que se pretende verificar é, o número de candidatos que são nomes compostos em que foram classificados como presentes os critérios mais o número de candidatos que não são nomes compostos em que o critério não foi identificado como presente. A tabela 4.1 ilustra a matriz dos possíveis resultados.

	critério presente	critério não presente
Nome Composto	C_1	C_2
Combinatória livre	C_3	C_4

Table 4.1: Matriz de resultados.

em que C_1 é o número total de nomes compostos em que se verificou o critério estudado; C_2 é o número total de nomes compostos em que não foi possível verificar esse critério; C_3 é o número total de combinatórias livres que apresentam o critério analisado; e C_4 é o número total de combinatórias livres que não apresentam o critério analisado.

A precisão de um critério na identificação dos nomes compostos é então calculada pela fórmula (28):

$$\text{Precisão do critério} = \frac{C_1 + C_4}{\text{Total de candidatos classificados}} \tag{28}$$

Os melhores resultados são aqueles cujas células C_1 e C_4 estejam maximizadas e as células C_2 e C_3 minimizadas.

4.2 Resultados

Esta secção apresenta os resultados mais relevantes para os métodos usados. Primeiro, são apresentados os resultados da aplicação dos filtros de procura das estruturas pretendidas. Seguidamente, apresentam-se os resultados da aplicação do algoritmo *HELAS* e do sistema *LocalMaxs*. Finalmente, são apresentados os resultados da aplicação dos programas para identificação de compostos baseado em critérios sintácticos.

4.2.1 Filtros *Nome Adjetivo* e *Nome de Nome*

A tabela 4.2 mostra o número total de padrões diferentes encontrados no *corpus* CETEMPúblico para as duas estruturas pretendidas e o número total de ocorrências de todos os padrões encontrados.

	Número de padrões diferentes	Total de ocorrências
Nome Adjetivo	1.032.733	6.002.836
Nome de Nome	529.497	2.834.893

Table 4.2: Resultados dos filtros.

Seria espectável um número maior de padrões diferentes encontrados, mas tal não ocorre devido às opções usadas na filtragem, nomeadamente a exclusão de casos de nomes próprios e adjetivos gentílicos, que compõem uma grande parte das sequências encontradas sem essas opções activadas.

4.2.2 Algoritmo *HELAS*

Nesta secção apresentamos os resultados obtidos na aplicação do algoritmo *HELAS* para as duas estruturas sintáticas. A tabela 4.3 mostra o número de padrões diferentes encontrados e o número total de instâncias, com a estrutura *nome adjetivo*, extraídos pelo algoritmo *HELAS* para os vários valores de α utilizados, usando a medida estatística SCP tal como foi descrito na secção 2.1.8.

Como se pode observar a informação gramatical tem um peso relevante na classificação de candidatos como nomes compostos. No entanto se se colocar demasiado peso nas categorias

Table 4.3: Resultados HELAS para o padrão *nome adjetivo* com a medida SCP.

α	Padrões diferentes	Número de ocorrências
0,0	37.557	924.880
0,1	79.826	2.777.930
0,2	127.189	4.385.502
0,3	132.350	4.570.680
0,4	129.849	4.575.417
0,5	117.720	4.475.638
0,6	99.857	4.271.949
0,7	82.413	4.007.052
0,8	68.754	3.727.891
0,9	58.371	3.463.785
1,0	50.357	3.203.238

gramaticais ou demasiado peso nas palavras que compõem a combinação, pode-se notar um rápido decréscimo de padrões identificados. Outro ponto de relevância é o facto de o número de ocorrências diminuir muito mais drasticamente quando só se tem em conta as categorias gramaticais

Este processamento do *HELAS* foi estendido também para a medida estatística ϕ^2 e os resultados apresentam-se na tabela 4.4.

Table 4.4: Resultados HELAS para o padrão *nome adjetivo* com a medida ϕ^2.

α	Padrões diferentes	Número de ocorrências
0,0	46.888	503.777
0,1	50.623	555.953
0,2	50.629	556.116
0,3	50.751	556.116
0,4	51.998	670.443
0,5	58.130	1.046.018
0,6	69.208	1.989.131
0,7	80.497	2.848.530
0,8	90.505	3.422.760
0,9	98.959	3.790.373
1,0	45.128	2.419.034

Como se pode observar, as categorias gramaticais são preponderantes na identificação de nomes compostos. No entanto, esta medida tem um comportamento ligeiramente diferente, do que se verifica com a SCP: o número de padrões encontrados e total de instâncias vai aumentando à medida que se dá peso ao valor de coesão das palavras, decrescendo bruscamente quando se dá o peso total à coesão das palavras. A figura 4.1 mostra estes resultados de uma forma sintetizada.

A tabela 4.5, mostra os resultados obtidos do mesmo tipo de processamento do *HELAS* mas

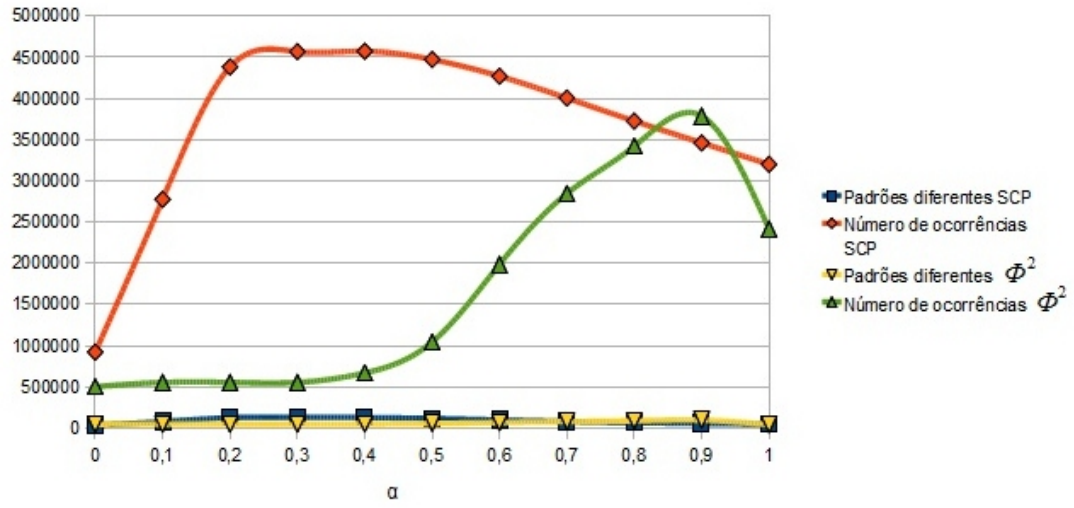


Figure 4.1: Resultados HELAS para o padrão *nome adjetivo*

para a estrutura *nome de nome* com a medida estatística SCP.

Table 4.5: Resultados HELAS para o padrão *nome de nome* com a medida SCP

α	Padrões diferentes	Número de ocorrências
0,0	441.089	2.726.813
0,1	441.080	2.726.804
0,2	309.487	2.466.601
0,3	169.026	2.015.802
0,4	104.520	1.698.244
0,5	73.060	1.476.595
0,6	54.940	1.316.955
0,7	43.763	1.189.686
0,8	36.357	1.086.166
0,9	31.291	998.021
1,0	24.537	894.231

Pelos resultados observados, para este tipo de estrutura as categorias gramaticais também são preponderantes na classificação de um candidato como nome composto. No entanto, ao contrário dos outros resultados mostrados anteriormente, estes vão decrescendo com quanto mais peso se dá à coesão das palavras.

Ainda para a mesma estrutura *nome de nome* o processamento do *HELAS* foi estendido também para a medida estatística ϕ^2 e os resultados apresentam-se na tabela 4.6.

Pode-se evidenciar resultados semelhantes aos resultados mostrados para o HELAS com o método SCP, os resultados vão decrescendo com quanto mais peso se vai dando à coesão das palavras. A figura 4.2 mostra os resultados de uma forma sintetizada.

Table 4.6: Resultados HELAS para o padrão *nome de nome* com a medida ϕ^2.

α	Padrões diferentes	Número de ocorrências
0,0	441.089	2.726.813
0,1	441.089	2.726.813
0,2	441.059	2.726.759
0,3	439.093	2.723.760
0,4	429.718	2.707.419
0,5	405.796	2.660.646
0,6	354.897	2.554.063
0,7	262.600	2.330.100
0,8	159.974	1.970.669
0,9	90.513	1.562.282
1,0	24.020	806.639

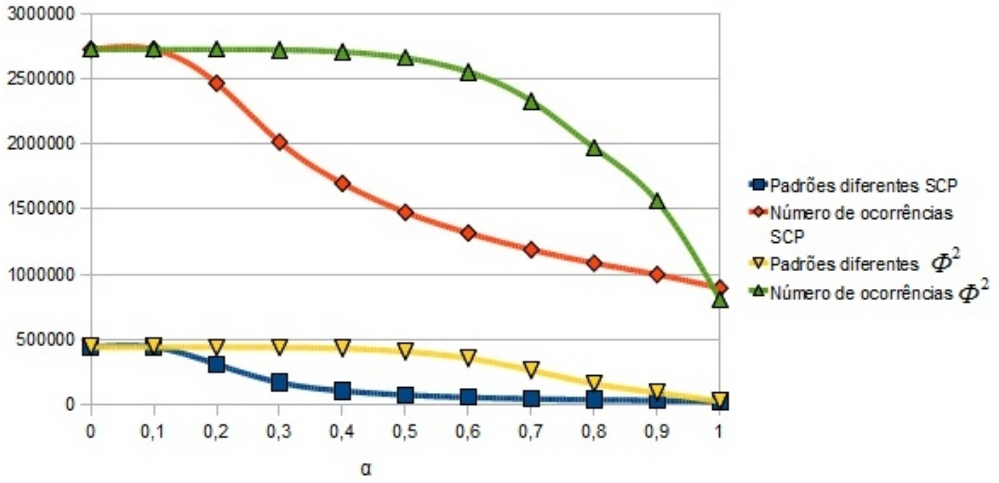


Figure 4.2: Resultados HELAS para o padrão *nome de nome*

Nas secções seguintes analisam-se os resultados do algoritmo *LocalMaxs*. Por uma questão de clareza, apresentam-se primeiro os resultados para os compostos *nome adjetivo* (secções 4.2.3 a 4.2.5) e seguidamente os dos *nome de nome* (secções 4.2.6 a 4.2.8).

4.2.3 Algoritmo *LocalMaxs* e os compostos *Nome Adjetivo*

Para avaliação do algoritmo *LocalMaxs* consideraram-se dois cenários. No primeiro, a cadeia de processamento STRING processou o *corpus* sem utilizar os consideráveis recursos lexicais já construídos e disponíveis no sistema, e que contem, à data de escrita deste documento, cerca de 35.000 palavras compostas. No segundo cenário, o algoritmo foi aplicado ao resultado da cadeia utilizando todos esses recursos. Os resultados de cada um destes cenários são apresentados nas tabelas 4.7 e 4.8 e sintetizadas no gráfico da figura 4.3.

Table 4.7: Resultados do *LocalMaxs* para a estrutura *nome adjetivo* quando a cadeia não identifica nomes compostos

Medida estatística	Nº de padrões diferentes	Nº de ocorrências
Dice coefficient	127.760	4.510.839
Specific Mutual Information(SMI)	28.040	1.466.816
ϕ^2	45.128	2.419.034
Log-likelihood Ratio	129.721	3.166.841
Mutual Expectation	140.161	4.723.158
Symmetric Conditional Probability(SCP)	50.357	3.203.238

Table 4.8: Resultados *LocalMaxs* para a estrutura *nome adjetivo* quando a cadeia identifica nomes compostos.

Medida estatística	Nº de padrões diferentes	Nº de ocorrências
Dice coefficient	116.565	2.981.983
Specific Mutual Information(SMI)	12.917	630.767
ϕ^2	21.319	1.251.948
Log-likelihood Ratio	116.036	1.829.301
Mutual Expectation	139.701	3.273.087
Symmetric Conditional Probability(SCP)	22.967	1.527.815

Ao comparar os resultados das tabelas, podemos ver o aumento significativo do número de padrões capturados pelos métodos estatísticos SMI, ϕ^2 e SCP, podemos concluir que estes métodos podem ser os melhores para identificar nomes compostos em *corpus* muito grandes. O número de padrões encontrados para as medidas Dice coefficient, Log-likelihood Ratio e Mutual Expectation também aumentaram mas tiveram um aumento abaixo dos 11.000 padrões, que é cerca de metade dos nomes compostos com a estrutura *nome adjetivo* que a cadeia de processamento já identificava. É de realçar que alguns dos novos padrões capturados, podem ser ruído introduzido pela nova informação.

O número de ocorrências também aumentou significativamente para todas as medidas. Isto indica que os novos padrões identificados possuem grande frequência no *corpus*. É, pois, possível concluir que a cadeia de processamento já faz a identificação dos nomes compostos mais comuns da língua portuguesa, com a estrutura *nome adjetivo*. Ainda se pode inferir que a frequência de um candidato tem um peso muito grande em todas as medidas estatísticas para a sua classificação.

4.2.4 Cruzamento das medidas estatísticas

Nas tabelas 4.9 e 4.10 apresentam-se os resultados do cruzamento das medidas estatísticas aqui utilizadas, tanto no primeiro cenário (sem os léxicos de palavras compostas) como no segundo cenário (com compostos), respetivamente. A primeira linha de cada tabela indica o número de padrões diferentes que são comuns a todas as medidas. As restantes linhas apresentam o

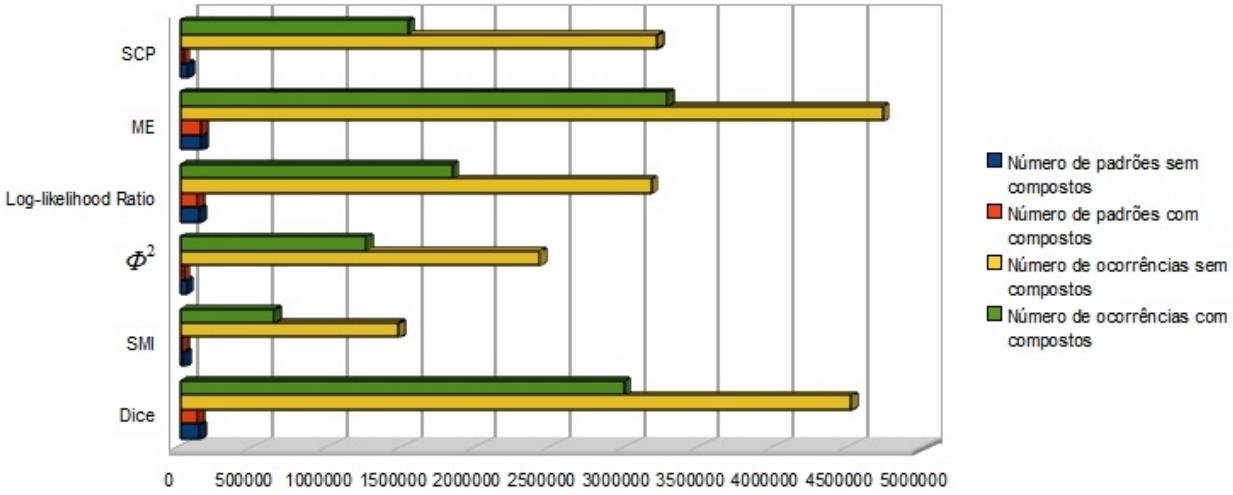


Figure 4.3: Resultados *LocalMaxs* para o padrão *nome adjetivo*

resultado do cruzamento de todas as medidas com exceção da medida indicada na coluna da esquerda.

Table 4.9: Resultados cruzados para a estrutura *nome adjetivo* quando a cadeia não identifica nomes compostos.

	Número de compostos	Número de ocorrências
Cruzamento de todas as medidas	17.354	401.232
sem Dice coefficient	17.357	401.264
sem SMI	39.244	1.492.635
sem ϕ^2	18.123	519.394
sem Log-likelihood Ratio	19.569	863.678
sem Mutual Expectation	17.354	401.232
sem SCP	17.810	409.190

Como se pode verificar, entre os dois cenários dá-se uma nítida diminuição, em cerca de 13.000, de candidatos a compostos capturados. Naturalmente, tal resultado é esperado, já que no segundo cenário os compostos já se encontram identificados sendo analisados pela cadeia como se de um nome simples se tratasse.

4.2.5 Validação manual por amostragem

Como já tinha sido referido na secção 4.1.2, medir a precisão de qualquer uma das medidas estatísticas aplicadas nos algoritmos, não é exequível. Assim para avaliação, os resultados das medidas estatísticas foram cruzados, obtendo uma lista de pares candidatos que são comuns a todas as medidas estatísticas. Dessa lista foram retirados aleatoriamente 1000 candidatos estratificados por frequência, ou seja, a lista original é organizada por grupos de frequências e

Table 4.10: Resultados cruzados para a estrutura *nome adjetivo* quando a cadeia identifica nomes compostos.

	Número de compostos	Número de ocorrências
Cruzamento de todas as medidas	4.368	91.788
sem Dice coefficient	4.374	91.840
sem SMI	14.577	498.115
sem ϕ^2	4.516	106.290
sem Log-likelihood Ratio	6.031	345.611
sem Mutual Expectation	4.368	91.788
sem SCP	4.439	94.498

de cada um desses grupos é aleatoriamente retirado um certo número de candidatos para formar a lista final de 1000 candidatos, os quais serão então classificados. Esta lista foi entregue a um linguista para ser validada manualmente.

Esta validação manual confirmou que 231 candidatos são efectivamente nomes compostos; 21 candidatos fazem parte de outros termos compostos mais longos, pelo que foram contabilizados como compostos; finalmente, 113 candidatos são colocações, isto é, combinações de palavras que se distinguem pela sua alta frequência de uso, por exemplo, *estilo inconfundível* ou *velocidade alucinante*. Este tipo de termos são interessantes para outro tipo de estudo mas não foram contabilizados para efeitos de precisão. Com estes resultados, obteve-se uma precisão global de 25,2%.

Esta lista de 1000 candidatos foi dividida em 4 grupos de 250 candidatos, organizados por ordem decrescente de frequência. Observou-se que para o grupo dos 250 candidatos mais frequentes (2277 a 11 ocorrências), se obteve uma precisão de 44,4%, os grupos seguintes possuem respectivamente as precisões de 27,2%, 21,6% e 7,6%. Tal confirma a ideia de que a frequência é um factor preponderante para avaliar a coesão interna de uma sequência candidata, sendo de descartar (ou, pelo menos, de atribuir tanta importância) as expressões que, num *corpus* com as dimensões como as do que aqui foi usado, apresentam frequências inferiores a 10 ocorrências.

4.2.6 Algoritmo *LocalMaxs* e os compostos *Nome de Nome*

O processo de avaliação dos resultados do algoritmo *LocalMaxs* com os compostos *nome de nome* é idêntico ao que foi apresentado para os *nome adjetivo*. Apresenta-se, em primeiro lugar, os resultados do *LocalMaxs* no cenário sem os recursos lexicais de palavras compostas (tabela 4.11) e, depois, usando esses recursos (tabela 4.12). O gráfico da figura 4.4 resume estes resultados.

Ao comparar estes resultados, em geral, verifica-se igualmente uma diminuição do número de padrões diferentes e do número de ocorrências quando se usam os recursos lexicais já disponíveis. Contudo, ao contrário do que sucede no caso dos compostos *nome adjetivo*, nestes compostos

Table 4.11: Resultados do *LocalMaxs* para a estrutura *nome de nome* quando cadeia não identifica nomes compostos

Medida estatística	Nº de padrões diferentes	Nº de ocorrências
Dice coefficient	153.787	1.861.528
Specific Mutual Information(SMI)	40.913	257.153
ϕ^2	24.020	806.639
Log-likelihood Ratio	439.168	2.724.720
Mutual Expectation	6.446	793.520
Symmetric Conditional Probability(SCP)	24.537	894.231

Table 4.12: Resultados *LocalMaxs* para a estrutura *nome de nome* quando a cadeia identifica nomes compostos.

Medida estatística	Nº de padrões diferentes	Nº de ocorrências
Dice coefficient	60.465	1.107.405
Specific Mutual Information(SMI)	12.710	326.183
ϕ^2	10.504	539.100
Log-likelihood Ratio	139.197	1.457.260
Mutual Expectation	16.030	941.267
Symmetric Conditional Probability(SCP)	10.740	526.499

essa diminuição não ocorre com a medida Mutual Expectation[1] verificando-se, pelo contrário, um aumento tanto dos padrões diferentes como do número de ocorrências. Também na medida SMI se verificou um aumento mas apenas do número de ocorrências quando se utiliza os lexicos de palavras compostas.

Estes resultados parecem indicar que estas medidas identificam muito ruído. Tal poderá dever-se ao facto de, para o processamento dos *nome de nome* se terem considerado todas as ocorrências e não apenas as que tinham uma frequência igual ou superior a cinco, como se fizera para os *nome adjetivo*. Neste sentido, estes métodos parecem particularmente sensíveis tornando-se incertos quando têm de lidar com eventos raros.

4.2.7 Cruzamento das medidas estatísticas

Em seguida, mostra-se os resultados do cruzamento das várias medidas estatísticas em cada um desses cenários - sem compostos (tabela 4.13) e com compostos (tabela 4.14), respetivamente.

Verificou-se um aumento na quantidade de termos em comum entre as duas fases de processamento, isto explica-se pelo facto de não se ter usado candidatos com ocorrências acima de 5. Os métodos estatísticos começam a ter comportamentos diferenciados quando lidam com eventos raros. É também de referir que a medida Mutual Expectation é a que identifica mais padrões

[1] Os valores inferiores da medida Mutual Expectation resultam de um erro de implementação que só foi detectado depois do processamento do corpus no cenário em que não se utilizavam os recursos lexicais. Assim, seria esperável que neste cenário, os valores desta medida fossem muito inferiores.

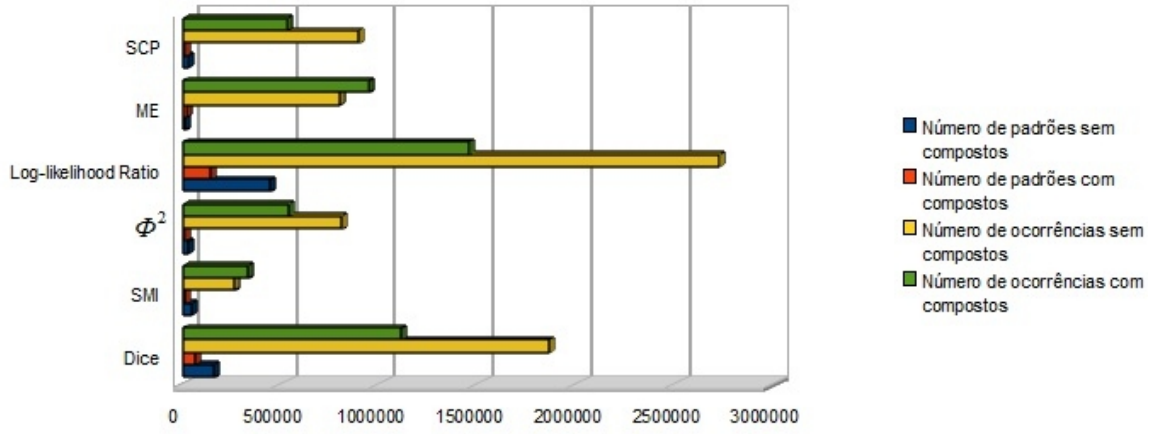


Figure 4.4: Resultados *LocalMaxs* para o padrão *nome de nome*

Table 4.13: Resultados cruzados para a estrutura *nome de nome* quando a cadeia não identifica nomes compostos.

	Número de compostos	Número de ocorrências
Cruzamento de todas as medidas	682	44055
sem Dice coefficient	682	44.055
sem SMI	2.892	505.517
sem ϕ^2	690	52.335
sem Log-likelihood Ratio	682	44.055
sem Mutual Expectation	18.097	95.977
sem SCP	682	44.055

que não são em comum com as outras medidas, como também se pode verificar na tabela 4.14. No entanto, nessa fase de processamento a SMI também é outra medida que identifica menos padrões em comum com as outras medidas.

4.2.8 Validação manual por amostragem

Foi também produzido para a estrutura *nome de nome* uma lista de 1000 candidatos selecionados aleatoriamente e estratificados por frequência. Esses candidatos foram retirados da lista resultante do cruzamento de todas as medidas estatísticas sem a medida Mutual Expectation. A lista resultante foi dada a um linguista para a validar manualmente. Esta validação manual identificou 93 candidatos como nomes compostos, 7 candidatos que fazem parte de outros termos compostos mais longos e 33 candidatos que formaram colocações. Com estes resultados, obteve-se uma precisão global de 10%.

Esta lista de 1000 candidatos foi dividida em 4 grupos de 250 candidatos organizados por ordem decrescente de frequência. Observou-se que o grupo dos 250 candidatos mais frequentes possuiu uma precisão de 30,8%, os grupos seguintes possuem respectivamente as precisões de

Table 4.14: Resultados cruzados para a estrutura *nome de nome* quando a cadeia identifica nomes compostos.

	Número de compostos	Número de ocorrências
Cruzamento de todas as medidas	2.433	179.843
sem Dice coefficient	2.433	179.843
sem SMI	5.950	505.321
sem ϕ^2	2.467	196.966
sem Log-likelihood Ratio	2.433	179.843
sem Mutual Expectation	5.739	192.603
sem SCP	2.433	179.843

	critério presente	critério não presente
Nome Composto	22,7%	2,5%
Combinatória livre	62,4%	12,4%

Table 4.15: Matriz de resultados do critério predicatividade na estrutura *nome adjetivo*.

2,8%, 1,2% e 5,2%. Como tinha sido verificado para a estrutura *nome de nome*, a frequência é um factor preponderante para avaliar a coesão de um candidato, observando-se, no entanto, que no caso dos *nome de nome*, a precisão é bastante inferior, mesmo no caso da classe de frequência mais alta. Tal deve ficar a dever-se às menores frequências consideradas nesta classe de composto.

4.3 Critérios Sintácticos

Nesta secção, apresentamos os resultados obtidos na aplicação dos critérios sintáticos para a identificação da estrutura *nome adjetivo* e para a estrutura *nome de nome*. Os programas de determinação dos critérios sintáticos da estrutura *nome adjetivo* foram aplicados à amostra aleatória apresentada nas secções 4.2.5 e 4.2.8.

As tabelas 4.15, 4.16, 4.17, 4.18, 4.19 e 4.20, mostram as percentagens correspondentes de cada célula da tabela enunciada na secção 4.1.3, relativamente a cada critério sintático. A tabela 4.21 mostra os valores de precisão medidos para cada um dos critérios sintáticos na estrutura *nome adjetivo*.

Uma análise superficial da tabela 4.21 poderia levar a concluir que o critério da ruptura paradigmática é aquele que apresenta os melhores resultados. No entanto, uma análise atenta

	critério presente	critério não presente
Nome Composto	25,1%	0,1%
Combinatória livre	74,6%	0,2%

Table 4.16: Matriz de resultados do critério coordenação na estrutura *nome adjetivo*.

	critério presente	critério não presente
Nome Composto	22,5%	2,7%
Combinatória livre	60,0%	14,8%

Table 4.17: Matriz de resultados do critério variação em grau na estrutura *nome adjetivo*.

	critério presente	critério não presente
Nome Composto	4,5%	20,7%
Combinatória livre	12,4%	62,4%

Table 4.18: Matriz de resultados do critério elisão do adjetivo na estrutura *nome adjetivo*.

da tabela 4.19 permite constatar que este valor de precisão do critério resulta de uma elevada percentagem de verdadeiros negativos (74,3%), isto é, expressões livres que, efectivamente não são capturados pelo critério. O que se pretende é um equilíbrio e maximização das células C_1 e C_4, tendo isto em conta, os critérios de variação em grau e variação em número aparentam ter os melhores resultados.

Os programas de determinação dos critérios sintáticos da estrutura *nome de nome*, foram aplicados à amostra aleatória desta estrutura enunciada na secção anterior.

As tabelas 4.22, 4.23, 4.24, 4.25, 4.26 e 4.27, mostram as percentagens correspondentes de cada célula da tabela enunciada na secção 4.1.3, relativamente a cada critério sintático. A tabela 4.28 mostra os valores de precisão medidos para cada um dos critérios sintáticos na estrutura *nome de nome*.

Os resultados obtidos de precisão (tabela 4.28) foram na sua generalidade maiores que a precisão obtida pelos métodos estatísticos. Pelos resultados observados no critério elisão do segundo nome, verificou-se que esta determinou como presente o critério em todos os candidatos. É possível que seja necessário retirar e comparar mais dependências de contexto, pois as que foram usadas podem não ser suficientes ou adequadas para este tipo de estrutura.

Pode-se também verificar que os critérios mais precisos são os que apresentam maior percentagem de verdadeiros negativos - muito maior que a percentagem de verdadeiros positivos. Por outro lado, os critérios que alcançaram uma precisão mais baixa são justamente os que apresentam maior percentagem de casos positivos. Por esta razão, é difícil determinar de forma clara qual o melhor critério para a classificação de candidatos com a estrutura *nome de nome*.

Os resultados obtidos pela exploração de critérios sintáticos são, na generalidade, positivos e promissores. No entanto, alguns destes critérios podem ainda ser melhorados. Em particu-

	critério presente	critério não presente
Nome Composto	0,9%	24,3%
Combinatória livre	0,5%	74,3%

Table 4.19: Matriz de resultados do critério ruptura paradigmática na estrutura *nome adjetivo*.

	critério presente	critério não presente
Nome Composto	11,4%	13,8%
Combinatória livre	29,8%	45,0%

Table 4.20: Matriz de resultados do critério variação em número na estrutura *nome adjetivo*.

Table 4.21: Precisão dos critérios sintáticos na estrutura *nome adjetivo*

Perda de predicatividade	35,1%
Coordenação	25,3%
Variação em grau	37,3%
Elisão do adjetivo	66,9%
Ruptura Paradigmática	75,2%
Variação em número	56,4%

	critério presente	critério não presente
Nome Composto	9,1%	0,9%
Combinatória livre	87,8%	2,2%

Table 4.22: Matriz de resultados do critério inserção de modificadores na estrutura *nome de nome*.

	critério presente	critério não presente
Nome Composto	7,4%	2,6%
Combinatória livre	79,4%	10,6%

Table 4.23: Matriz de resultados do critério variação do determinante na estrutura *nome de nome*.

	critério presente	critério não presente
Nome Composto	8,2%	1,8%
Combinatória livre	86,5%	3,5%

Table 4.24: Matriz de resultados do critério coordenacao na estrutura *nome de nome*.

	critério presente	critério não presente
Nome Composto	10%	0%
Combinatória livre	90%	0%

Table 4.25: Matriz de resultados do critério elisão do segundo nome na estrutura *nome de nome*.

	critério presente	critério não presente
Nome Composto	0,4%	9,6%
Combinatória livre	0,5%	89,5%

Table 4.26: Matriz de resultados do critério ruptura paradigmática na estrutura *nome de nome*.

	critério presente	critério não presente
Nome Composto	3,6%	6,4%
Combinatória livre	4,1%	85,9%

Table 4.27: Matriz de resultados do critério variação em número na estrutura *nome de nome*.

Table 4.28: Precisão dos critérios sintáticos na estrutura *nome de nome*

Inserção de modificadores	11,3%
Variação do determinante	18,0%
Coordenação	11,7%
Elisão do segundo nome	10%
Ruptura Paradigmática	89,9%
Variação em número	89,5%

lar, o critério da elisão do *adjetivo*, no caso dos *nome adjetivo*, e o da elisão do complemento *de nome*, nos *nome de nome* deverá apresentar melhores resultados se se vier a integrar outras dependências definitórias de contexto do primeiro nome isolado, aumentando o âmbito da comparação do contexto das sequências candidatas.

Também o critério da ruptura paradigmática poderá vir a apresentar melhroes resultados se se dispuser de informação distribucional que permita comparar a estrutura candidata dentro dos respetivos paradigmas em que cada elemento componente (cada um dos nomes ou o adjetivo) se pdoe inserir.

Chapter 5

Conclusão e Trabalho Futuro

5.1 Conclusão

Foi feita uma breve descrição das ferramentas usadas no processo de extrair candidatos a nomes compostos no *corpus* CETEMPúblico, nomeadamente a cadeia de processamento STRING para processar o *corpus* e retirar informação gramatical; o Condor que providencia uma calenderização e processamento de forma paralela na fase de processamento do corpus; e da ferramenta Hadoop, que facilita o acesso aos dados processados pela cadeia de processamento.

Foi feito também uma descrição dos sistemas que se usaram para a identificação dos candidatos, bem como das estratégias criadas para a determinação da presença de propriedades sintáticas nas expressões candidatas. Isto levou à criação de programas para atingir os objetivos pretendidos. Estes programas passaram por um processo de avaliação para determinar a sua precisão.

Com este trabalho, podemos verificar que nomes compostos apresentam, na sua grande maioria, frequências altas. Também se pode constatar que usar informação lexical na identificação automática influencia a avaliação que os sistemas fazem. Ao processar candidatos cujo número de ocorrências é inferior a 5, torna-se óbvio que o processo de extração dá origem a muitos candidatos espúrios, o que nos diz que os sistemas têm problemas ao lidar com eventos raros.

Os resultados dos critérios sintáticos parecem positivos e revelam que certos critérios sintáticos podem ser formalizados e aplicados de maneira relevante na identificação de nomes compostos, pelo que muitos sistemas poderão vir a ganhar com este tipo de informação.

5.2 Trabalho Futuro

Nesta secção final traçamos alguns pistas de trabalho futuro. Os aspectos principais que podem ser realizados são:

- Estender a procura a outras estruturas sintáticas, como por exemplo as estruturas adverbiais com a forma de sintagmas preposicionais;
- Para essas novas estruturas, estudar e aplicar automaticamente a determinação da presença dos critérios sintáticos que permitem a sua identificação;
- Usar a informação fornecida pela determinação de critérios sintáticos por sistemas que usam mais informação que a frequência de candidatos e frequência dos seus constituintes, como por exemplo o GALEMU (secção 2.2.3);
- Aumento da informação lexical disponibilizada pela cadeia de processamento STRING. De momento não existe qualquer tipo de informação relativamente a paradigmas distribucionais de palavras. Ao extrair este tipo de informação, a determinação do critério da ruptura paradigmática para as duas estruturas poderá vir a ter melhores resultados;
- Melhorar a procura dos critérios sintáticos, nomeadamente a elisão do adjetivo na estrutura *nome adjetivo* ou a elisão do nome na estrutura *nome de nome*. Estes critérios ainda têm muito espaço para desenvolvimento e seria desejável no futuro estender a mais informação de contexto;
- Usar os nomes compostos extraídos para enriquecer a cadeia de processamento STRING.

Dos aspectos enunciados, aumentar a informação lexical disponibilizada pela cadeia de processamento STRING é a mais importante, porque pode vir a melhorar outros projectos que usem a cadeia. Estender o trabalho para outras estruturas também é relevante porque a cadeia de processamento tende a melhorar com a identificação destas expressões compostas, permitindo uma mais precisa identificação das unidades de sentido num texto.

Bibliography

Adriani, M. & C. J. V. Rijsbergen (1999). Term similarity-based query expansion for cross-language information retrieval. In *Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries ECDL 99*, pp. 311–322.

Aït-Mokhtar, Salah; Jean-Pierre Chanod, and Claude Roux (2002). Robustness beyond shallowness: Incremental deep parsing. Natural Language Engineering, 8. Cambridge University Press, New York, pp. 121–144.

Azuaga, L., I. Faria, E. Ribeiro, I. Duarte, & C. Gouveia (1996). Introdução à linguística geral e portuguesa. Lisboa: Caminho, pp. 215–244.

Ballesteros, L. & W. B. Croft (1998). Resolving ambiguity for cross-language retrieval. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, pp. 64–71. ACM.

Baptista, J. (1994). Estabelecimento e formalização de classes de nomes compostos. Master's thesis, Faculdade de Letras da Universidade de Lisboa, Lisboa.

Church, K. W. & P. Hanks (1990). Word association norms, mutual information, and lexicography. *Comput. Linguist. 16*(1), 22–29.

Daille, B. (1996). Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. In J. Klavans & P. Resnik (Eds.), *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, pp. 49–66. Cambridge, Massachusetts: The MIT Press.

Dias, G. (2003). Multiword unit hybrid extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions*, Morristown, NJ, USA, pp. 41–48. Association for Computational Linguistics.

Dias, G., S. Guilloré, & J. Lopes (1999). Language Independent Automatic Acquisition of Rigid Multiword Units from Unrestricted Text corpora. In *Proceedings of 6éme Conferérence Annuelle sur le Traitement Automatique des Langues Naturelles*, Cargése.

Dias, G. & S. Nunes (2004). Evaluation of Different Similarity Measures for the Extraction of Multiword Units in a Reinforcement Learning Environment. In *Proceedings of the 4th International Conference on Languages Resources and Evaluation*, pp. 1717–1721.

Dice, L. (1945). Measures of the Amount of Ecologic Association Between Species. *Journal of Ecology*.

Diniz, C. F. P. (2010). Um conversor baseado em regras de transformação declarativas. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics 19*, 61–74.

Frantzi, K., S. Ananiadou, & H. Mima (2000). Automatic recognition of multi-word terms: the C-value/NC-value method. *International Journal on Digital Libraries V3*(2), 115–130.

Gale, W. & K. Church (1991). Concordances for Parallel Texts. *Proceedings of the Seventh Annual Conference of the UW Center for the New OED and Text Research, Using Corpora*.

Gross, G. (1988). Degré de figement des noms composés. In *Languages 90*, Paris: Larousse, pp. 57–72.

Hull, D. & G. Grefenstette (1996). Experiments in Multilingual Information Retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 6–9.

Johansson, C. (1996). Good bigrams. In *Proceedings of the 16th conference on Computational linguistics*, Morristown, NJ, USA, pp. 592–597. Association for Computational Linguistics.

Kohonen, T. (1989). *Self-organization and Associative Memory* (3rd edition ed.). New York, NY, USA: Springer-Verlag New York, Inc.

Kohonen, T., J. Kangas, J. Laaksonen, & K. Torkkola (1992). LVQ PAK: A program package for the correct application of Learning Vector Quantization algorithms. pp. 725–730.

Lopes, G. & J. Silva (1999). A local Maxima Method and a Fair Dispersion Normalization for Extracting Multiword Units. In *Proceedings of the* 6^{th} *Meeting on the Mathematics of Language*, pp. 369–381.

Luís, T. (2008). Parallelization of Natural Language Processing Algorithms on Distributed Systems. Master's thesis, Universidade Técnica de Lisboa, Portugal.

Mamede, N. (2011). STRING - A Cadeia de Processamento de Língua Natural do L^2F em Fevereiro de 2011 (Technical Report). L^2F - Laboratório de Sistemas de Língua Falada, INESC-ID Lisboa, Lisboa.

Manning, C. & H. Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachussets.

Martínez-Santiago, F., M. Díaz-Galiano, M. Martín-Valdivia, V. Rivas-Santos, & L. U. na Lopez (2002). Using Neural Networks for Multiword Recognition in IR. In *Proceedings of Conference of International Society of Knowledge Organization (ISKO-02)*, Granada, Espanha, pp. 559–564.

Miller, G. (1995). Wordnet: A lexical database for english. *Communications of the ACM 38*, 39–41.

Pardal, J. P. (2007). Manual do Utilizador do RuDriCo. L^2F - Laboratório de Sistemas de Língua Falada, INESC-ID Lisboa, Lisboa.

Pecina, P. & P. Schlesinger (2006). Combining Association Measures for Collocation Extraction. In *ACL'06*, pp. 652.

Ribeiro, R., L. Oliveira, & I. Trancoso (2003). Using morphossyntactic information in tts. In *In Computational Processing of the Portuguese Language: 6th International Workshop, PROPOR 2003*, pp. 26–27. Springer.

Santos, D. & P. Rocha (2001). Evaluating CETEMPúblico, a free resource for Portuguese. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, pp. 442–449.

Silva, J., G. Dias, S. Guilloré, & J. Lopes (1999). Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. In *EPIA '99: Proceedings of the 9th Portuguese Conference on Artificial Intelligence*, London, UK, pp. 113–132. Springer-Verlag.

Smadja, F., K. R. McKeown, & V. Hatzivassiloglou (1996). Translating collocations for bilingual lexicons: a statistical approach. *Comput. Linguist. 22*(1), 1–38.

Tannenbaum, T., D. Wright, K. Miller, & M. Livny (2001). Condor – A Distributed Job Scheduler. In T. Sterling (Ed.), *Beowulf Cluster Computing with Linux*, Chapter 15. MIT Press.

Appendix A

Lista de *nome adjetivo* classificados como compostos e respectivas ocorrências

impacte ambiental 2277

junta metropolitano 820

abuso sexual 680

problema técnico 618

défice democrático 528

estabilidade cambial 487

parlamento nacional 457

imprensa internacional 429

barreira psicológico 376

engenho explosivo 354

sigilo bancário 336

inovação tecnológico 317

proteção ambiental 306

sentido contrário 293

propriedade intelectual 278

roda motriz 264

auxílio humanitário 242

lua cheio 231

elefante branco 218

grupo estrangeiro 211

potência administrante 204

conversa informal 197

medida económico 187

segurança pessoal 183

interlocutor privilegiar 168

monopólio estatal 162

mão atar 158

modu operandi 151

catástrofe ecológico 139

postal ilustrar 134

marca próprio 130

diagnóstico precoce 118

contenção salarial 114

zona frontal 111

ensino recorrente 103

forma categórico 100

nível financeiro 90

melting pot 87

atividade piscatório 75

atividade cinegético 70

área construir 68

acordo amigável 66

exterminador implacável 64

instituto hidrográfico 61

fora-de-jogo posicional 18

ex-director desportivo 18

pluralismo informativo 18

cartão canelar 18

jogo viciar 17

latitude médio 17

requalificação urbanístico 16

boneca insuflável 16

revolução bolchevista 16

futebol aéreo 16

feira tradicional 15

cor diverso 15

poesia erótico 15

violino barroco 14

grau superlativo 14

civilização burguês 14

fim caritativo 14

campo raso 13

pai desconhecer 13

ar despreocupar 13

centro oceanográfico 13

indústria hollywoodiano 12

centro lúdico 12

alimentador automático 12

gasto corrente 12

pessoal militarizar 12

comportamento negligente 12

kung fu 11

menino feio 11

cabeça tapar 11

aleitamento materno 11

dança sagrar 11

cara chapar 11

zona urbanizável 11

cabimentação orçamental 11

futuro longínquo 11

sinalização informativo 11

soberania territorial 11

jantar informal 11

resíduo reciclável 10

funcionamento experimental 10

diarreia hemorrágico 10

gás asfixiante 10

dissençõe interno 10

pescoço esticar 10

heterónimo pessoano 10

coração cheio 10

carapau frigir 10

continente latino-americano 10

abuso verbal 9

termo afetivo 9

exploração sustentável 9

custa alheio 9

vídeo experimental 9

levantamento arquitetónico 9

automobilismo internacional 9

separação amigável 9

semáforo verde 9

drama romântico 9

unidade anti-terrorista 9

crónica radiofónico 9

utilização sustentável 9

intervenção florestal 9

purga estalinista 9

balanceamento atacante 8

pesticida químico 8

coma superficial 8

seminário conciliar 8

polimorfismo humano 8

modo fasear 8

versão suave 8

parede externo 8

via descendente 8

norma imperativo 8

soma positivo 8

aprofundamento institucional 8

título vitalício 8

alma matar 8

exame escolar 8

desenvolvimento emocional 8

economia clandestino 8

feijão encarnar 8

passaporte comunitário 8

convivência diário 7

humor brejeiro 7

lugar idílico 7

plataforma elevar 7

greve ilegal 7

correio interno 7

custo processual 7

meio costeiro 7

curandeiro tradicional 7

aviação geral 7

magister dixit 7

alga verde 7

despesa consolidar 7

satisfação estampar 7

mira apontar 7

rocha escarpar 7

laço fraternal 7

complicação pós-operatório 7

execução coercivo 7

cenário envolvente 7

ar suspeito 7

língua dominante 7

cara visível 7

estratégia ganhador 7

falsificação agravar 7

princípio vital 7

descanso forçar 7

germe patogénico 7

descanso dominical 7

margem tangencial 7

hemorragia nasal 7

silêncio conivente 7

tranquilidade social 6

coração apertar 6

papo cheio 6

população anónimo 6

via extra-judicial 6

travagem direcional 6

céu baixo 6

atividade químico 6

porto bacalhoeiro 6

gesto comedir 6

tratamento dentário 6

canto fúnebre 6

ganho direto 6

igualdade religioso 6

lugares- comum 6

folk songs 6

malha tecer 6

morada oficial 6

milícia antidroga 6

iniciação carnal 6

ordem salesiano 6

bastião rebelde 6

despejo sumário 6

retinopatia diabético 6

fogo proibir 6

comida fresco 6

ala histórico 6

parque radical 6

padrão comportamental 6

traumatismo abdominal 6

humor melancólico 6

fracasso estrondoso 6

micro-organismo primitivo 6

rive gauche 6

centro terciário 6

videogravador estéreo 6

perceção social 6

solvente orgânico 6

registo paródico 6

tiragem reduzir 5

paixão desencontrar 5

sismicidade induzir 5

ordem equestre 5

execução vocal 5

islão radical 5

fabricação artesanal 5

ala populista 5

resistência khmer 5

gasto sumptuoso 5

oceanografia biológico 5

recinto polidesportivo 5

letra imprimir 5

bula pontifício 5

hi-fi stereo 5

alfaia litúrgico 5

cólica abdominal 5

oceano primitivo 5

envergadura moral 5

cordeiro inocente 5

força letal 5

princípio estatutário 5

memória profundo 5

metal branco 5

escalão competitivo 5

delinquência infantil 5

Appendix B

Lista de *nome de nome* classificados como nomes compostos e respectivas ocorrências

posto de trabalho 4648

carteira de encomenda 460

pré-aviso de greve 297

unidade de diálise 163

dispensa de OPA 132

embarcação de recreio 115

tese de mestrado 95

jantar de gala 82

circulação de peão 74

poste de iluminação 67

baile de máscara 61

testa de ferro 56

segredo de polichinelo 38

enfarte de miocárdio 33

cláusula de isenção 31

largada de toiro 28

rito de iniciação 26

europeu de esperança 25

mar de dúvida 21

clínica de aborto 20

calço de travão 19

poço de recarga 16

locutor de continuidade 15

cassete de video 13

leão de bronze 13

manga de camisa 11

carbonato de cálcio 10

diálogo de bateria 10

caderneta de racionamento 10

cavalo de toiro 10

ajudante de eletricista 9

hijo de puta 9

pega de cernelha 8

sequência de tecla 8

reencaminhamento de chamada 8

cêntimo de euro 7

choque de mentalidade 7

júri de doutoramento 7

torre de refrigeração 7

sapato de bico 6

detetor de mina 6

tampa de panela 6

gaiola de pássaro 6

cesta de vime 5

cantiga de roda 5

espingarda de cana 5

chaminé de ventilação 5

ninho de lacrau 5

eixo de simetria 5

largura de ombro 5

gás de combustão 5

certificado de equivalência 5

troco de quê 5

acetato de ciproterona 4

hino de estádio 4

agulha de pinheiro 4

flor de sabugueiro 4

ideal de cavalaria 4

magistrado de turno 4

pensionista de invalidez 4

perturbação de sono 4

fracionamento de plasma 4

recuperador de calor 4

cordão de duna 4

cerveja de barril 4

prancha de windsurf 4

abaixamento de padrão 4

açorda de coentrada 3

coleira de telemetria 3

bebedeira de caixão 3

doce de coco 3

gozo de folga 3

regente de cadeira 3

cabaz de natal 3

cana de açucar 2

seio de silicone 2

cesto de gávea 2

recheio de espinafre 2

paté de fígado 2

estojo de lápis 2

diabete de tipo_i 2

filtro de chaminé 1

sprays de pimenta 1

cana de soprador 1

manto de púrpura 1

comunhão de leito 1

bar de striptease 1

gel de sílica 1

hidróxido de bário 1

louça de forno 1

risca de colarinho 1

recetor de telex 1

miga de feijão 1

pássaro de gaiola 1

contrato-promessa de cessão 1

boneco de luva 1

botija de camping-gá 1

bateria de PB 1

varredor de ruas 1

mola de impulsão 1