Resolução de anáfora pronominal em português utilizando o algoritmo de Lappin e Leass

Thiago Thomes Coelho

Dissertação de Mestrado

Instituto de Computação
Universidade Estadual de Campinas

Resolução de anáfora pronominal em português utilizando o algoritmo de Lappin e Leass

Thiago Thomes Coelho[1]

16 de Novembro de 2005

Banca Examinadora:

- Profa. Dra. Ariadne Maria Brito Rizzoni Carvalho
  Instituto de Computação, Unicamp (Orientadora)
- Profa. Dra. Renata Vieira
  Centro de Ciências Exatas e Tecnológicas, Unisinos
- Prof. Dr. Jacques Wainer
  Instituto de Computação, Unicamp
- Prof. Dr. Tomasz Kowaltowski
  Instituto de Computação, Unicamp (Suplente)

[1]**Auxílio financeiro parcial do CNPq**

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA CENTRAL DA UNICAMP

C65r

Coelho, Thiago Thomes.
Resolução de anáfora pronominal em português utilizando o algoritmo de Lappin e Leass / \c Thiago Thomes Coelho. -- Campinas, SP : \b [s.n.], 2006.

Orientador: Ariadne Maria Brito Rizzoni Carvalho.
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Processamento de línguagem natural (Computação). 2.Anáfora (Linguística). 3. Algoritmos de computador. I.Carvalho, Ariadne Maria Brito Rizzoni. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Lappin and Leass' algorithm for pronominal anaphora resolution in portuguese.

Palavras -chave em inglês (Keywords): Natural language processing (Computer science), Anaphora (Linguistics), Computer algorithms.

Área de Concentração: Processamento de Línguas Naturais.

Titulação: Mestre em Ciência da Computação.

Banca examinadora: Renata Vieira, Jacques Wainer, Tomasz Kowaltowski.

Data da Defesa: 16/12/2005.

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 16 de dezembro de 2005, pela Banca examinadora composta pelos Professores Doutores:

Profa. Dra. Renata Vieira
UNISINOS

Prof. Dr. Jacques Wainer
IC / UNICAMP

Profa. Dra. Ariadne Maria Brito Rizzoni Carvalho
IC / UNICAMP

Resolução de anáfora pronominal em português utilizando o algoritmo de Lappin e Leass

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Thiago Thomes Coelho e aprovada pela Banca Examinadora.

Campinas, 16 de Dezembro de 2005.

Profa. Dra. Ariadne Maria Brito Rizzoni Carvalho
Instituto de Computação, Unicamp
(Orientadora)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Resumo

Um dos problemas do processamento de língua natural é a resolução de anáforas, fenômeno que ocorre quando duas ou mais expressões de um texto se referem a uma mesma entidade do discurso. Diversos algoritmos foram propostos para fazer a identificação do antecedente anafórico de pronomes, como o algoritmo de Lappin e Leass (1994), Hobbs (1978) e Grosz et al. (1995). A resolução de anáforas pode melhorar consideravelmente a qualidade do resultado em diversas aplicações de processamento de língua natural, como por exemplo a recuperação e extração de informações, geração automática de resumos, traduções automáticas, entre outros.

A pesquisa envolvendo o processamento automático do português ainda é limitada, se comparada a outras línguas, como inglês, francês e espanhol. Este trabalho visa implementar e avaliar o algoritmo de Lappin e Leass para a resolução de anáforas pronominais em terceira pessoa e pronomes reflexivos e recíprocos, em português. O algoritmo é baseado em um sistema de pesos, atribuídos de acordo com a estrutura sintática da sentença, e utiliza apenas conhecimento sintático na resolução das anáforas.

Abstract

One of the most challenging problems in natural language processing is anaphora resolution. This phenomenon occurs when one or more expressions in a text refer to the same entity previously mentioned in the discourse. Several approaches to anaphora resolution have been proposed, such as Lappin e Leass' algorithm (1994), Hobbs (1978) and Grosz et al. (1995). Anaphora resolution can improve significantly the performance of several natural language processing applications, such as automatic translation and summarisation, among others.

Research in automatic processing of Portuguese is still incipient, when compared with other languages such as English, Spanish or French. This work aims at developing and evaluating the Lappin and Leass' algorithm for third person, as well as reflexive and reciprocal pronoun resolution, in Portuguese. The algorithm relies on an weighting scheme assigned according to the syntactic structure of the sentence, and on syntactic knowledge to perform anaphora resolution.

Agradecimentos

À minha família, e especialmente aos meus pais, pelo carinho, amor, incentivo e por tudo que fizeram para garantir meu crescimento, amadurecimento e aprendizado.

À minha orientadora Profa. Ariadne, pela orientação, paciência, confiança, apoio e pelos valiosos direcionamentos, os quais tornaram possível a realização deste trabalho.

Agradeço também aos membros da banca, em especial à Profa. Renata Vieira e à Profa. Ariadne, cujas relevantes sugestões melhoraram este trabalho.

Aos professores Paulo Quaresma e Charlotte Marie Chambelland Galves, pelas fundamentais e preciosas contribuições.

À Sandra Collovini, Sheila Morais de Almeida e Ana Margarida Aires pela paciência e colaboração ao longo deste trabalho.

A todos os amigos que conquistei neste período, pelo companheirismo, apoio e compreensão.

A todos que participaram na minha vida e de alguma forma contribuiram para eu estar onde estou hoje.

Agradeço ao CNPq pelo apoio financeiro.

E, finalmente, a todos os colegas, professores e funcionários do Instituto de Computação.

Sumário

Lista de Abreviações

ADVP: locução adverbial
ADV: advérbio
PP: locução preposicionada
P: preposição
PRON: pronome
S: sentença
SN: sintagma nominal
SV: sintagma verbal
V: verbo

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

Lingüística Computacional é o estudo de sistemas computacionais visando entender e gerar língua natural. Ao entender os processos da linguagem em termos procedurais, podemos dar aos sistemas computacionais a habilidade de gerar e interpretar língua natural. Isso torna possível que computadores executem tarefas lingüísticas, como tradução, e processem dados textuais, como livros, jornais e revistas [Gri92].

Este trabalho propõe a utilização do algoritmo de Lappin e Leass para resolver o fenômeno lingüístico denominado anáfora pronominal, que ocorre, por exemplo, no seguinte texto[1]:

(1) Maria$_i$ gosta de Pedro$_j$ mas ela$_i$ resiste em não falar com ele$_j$.
Eles brigaram pois Pedro é um cabeça-dura.

Nesse texto, "ela" e "ele" são anáforas pronominais, que se referem à "Maria" e "Pedro", respectivamente. Entretanto, o pronome "Eles" se refere a "Maria" e "Pedro" conjuntamente. Portanto, esse fenômeno consiste em referenciar uma entidade do discurso, mencionada previamente, através de alguma abreviação ou forma lingüística alternativa. A determinação das entidades mencionadas pela expressão anafórica é denominada resolução.

A resolução de anáforas é um dos problemas mais difíceis da lingüística computacional [Smi91], e é uma etapa vital para a maioria dos sistemas de processamento de língua natural [Mit01]. Além disso, a determinação do antecedente anafórico tem papel fundamental na coesão de um texto ou diálogo, como no texto (1), acima.

A dificuldade no tratamento de anáforas reside na identificação do elemento co-referenciado nos casos em que múltiplos antecedentes são possíveis para uma certa expressão anafórica. Nesses casos, a determinação do antecedente apropriado é uma tarefa desafiadora e pode requerer o uso de conhecimentos variados, como o emprego de informações

[1] Os constituintes com mesmo índice são co-referentes, ou seja, se referem a mesma entidade do discurso.

morfológicas, léxicas e restrições semânticas e pragmáticas [Mit02]. O texto (2), a seguir, ilustra esse problema:

(2) O deputado mandará matar o senador caso ele ameace denunciar o esquema de corrupção. Ele exige uma quantia maior do dinheiro que está sendo desviado para não denunciá-lo.

Observa-se que as entidades referenciadas pelos pronomes "ele", "Ele" e "lo" podem ser tanto "senador" quanto "deputado".

1.1 Contexto e Objetivo

A pesquisa envolvendo o processamento automático do português ainda é limitada, se comparada a outras línguas, como inglês, francês e espanhol. Diversos algoritmos foram propostos para fazer a identificação do antecedente anafórico de pronomes, como o algoritmo de Lappin e Leass [LL94], o algoritmo de Hobbs [Hob78] e o algoritmo de *Centering*, proposto em [GWJ95, BFP87].

Este trabalho visa implementar e avaliar o algoritmo de Lappin e Leass na língua portuguesa. O algoritmo resolve anáforas pronominais em terceira pessoa e pronomes reflexivos e recíprocos. Ele é baseado em um sistema de pesos, atribuídos de acordo com a estrutura sintática da sentença, e em uma representação simples do modelo do discurso[2]. A escolha desse algoritmo foi devido ao seu bom desempenho obtido com a língua inglesa.

1.2 Metodologia

A metodologia utilizada para desenvolver este trabalho envolveu uma série de etapas, iniciadas com uma revisão bibliográfica, através da leitura de artigos científicos, trabalhos acadêmicos e livros relacionados ao processamento de língua natural e à resolução de anáforas.

Após a compreensão do contexto onde o trabalho está inserido, foi feita uma análise do algoritmo de Lappin e Leass e das ferramentas que foram utilizadas na sua implementação. Na última etapa do trabalho, o algoritmo foi implementado e avaliado com três corpora distintos.

[2] O modelo do discurso contém todas as entidades mencionadas no discurso e as relações das quais elas participam [JHM00].

1.3 Organização

Neste trabalho, além do presente capítulo, há outros seis capítulos que expõem de forma detalhada cada tópico abordado:

- *Capítulo 2 - Anáforas:* são considerados aspectos importantes sobre anáforas e seu processo de resolução. Também são abordados alguns dos principais algoritmos que visam resolver esse fenômeno lingüístico;
- *Capítulo 3 - Algoritmo de Lappin e Leass:* é apresentado o algoritmo de Lappin e Leass, assim como sua adaptação para o português;
- *Capítulo 4 - Ferramentas utilizadas:* são apresentadas as ferramentas utilizadas, como o analisador sintático PALAVRAS e a ferramenta Xtractor, que visa facilitar a extração das informações necessárias geradas pelo analisador;
- *Capítulo 5 - Implementação:* são apresentadas a plataforma escolhida para implementação e a modelagem do sistema desenvolvido, que implementa o algoritmo proposto neste trabalho;
- *Capítulo 6 - Avaliação do algoritmo:* são apresentados os corpora utilizados e os resultados da avaliação do algoritmo;
- *Capítulo 7 - Conclusão:* são feitas as considerações finais e apresentadas as perspectivas futuras para a continuidade do trabalho.

Capítulo 2

Anáforas

As referências são os ítens da língua que remetem a outros ítens do discurso, implícitos ou explícitos, necessários a sua interpretação [Koc89]. As anáforas são um tipo de referência na qual o item do discurso referenciado, presente no texto ou diálogo, precede a expressão referencial.

Neste capítulo, inicialmente é apresentada uma visão geral do fenômeno lingüístico da referência. Em seguida, são apresentados alguns problemas na resolução de anáforas, em particular anáforas pronominais, que são o objeto de estudo deste trabalho. Também são apresentados alguns fatores geralmente empregados no processo de determinação do antecedente anafórico, como as preferências e as restrições sintáticas. Finalmente, são apresentados os trabalhos relacionados.

2.1 Referência

O fenômeno da referência consiste no mecanismo do discurso no qual é feita uma referência abreviada à alguma entidade (ou entidades), na expectativa de que o ouvinte do discurso seja capaz de determinar a identidade da entidade abreviada [Hir81]. A entidade abreviada é levada ao conhecimento do ouvinte através de uma instrução lingüística, que varia de acordo com o conhecimento sobre a referida entidade que o locutor possui, e acredita que o ouvinte também possua [MBDF89]. A entidade referenciada chama-se referente ou antecedente.

O antecedente geralmente é representado como um sintagma nominal[1], ou na forma de uma estrutura mais complexa, como uma sentença[2], ou um conjunto de sentenças. Em

[1] Um sintagma é constituído por um grupo de elementos lingüísticos que formam uma unidade numa organização hierarquizada. Ele é seguido por um qualitativo que define sua categoria gramatical. No caso do sintagma nominal, o termo qualitativo geralmente é um nome ou substantivo [CJ81, DGG+04].

[2] Sentença é um termo também usado neste trabalho como sinônimo de frase ou oração.

alguns casos, o antecedente pode ser representado por um termo implícito, ou mesmo pelo discurso como um todo [MBDF89].

Há casos onde o referente está no contexto situacional, ou seja, o antecedente encontra-se no mundo externo, e não no texto. Esse tipo de referência é denominado exófora, exofórica, situacional ou extratextual [HH76, Hir81, Fáv91]. Neste tipo de relação referencial, geralmente a entidade em questão só tem identidade no espaço cognitivo determinado pelo discurso e pela situação [MBDF89]. Na sentença (3), abaixo, a interpretação correta do termo "isto" depende do contexto situacional, como uma indicação gestual do objeto referido.

(3) Pedro, pegue isto e guarde no baú agora!

Há situações onde o antecedente é um conceito ou entidade mencionado explicitamente ou implicitamente pelo texto ou discurso. Esse tipo de referência é denominada endófora, endofórica, textual ou co-referência [HH76, Hir81, Fáv91]. O processo de determinação do antecedente da expressão de co-referência é chamado resolução [Hir81]. A expressão de co-referência e seu antecedente são ditos co-referentes. As referências textuais podem ser elípticas, catafóricas ou anafóricas.

A referência elíptica (ou elipse) ocorre quando há omissão de algum constituinte da sentença, recuperável pelo contexto [Fáv91], como na sentença (4), abaixo, onde o termo elíptico, isto é $\emptyset$, se refere ao sintagma nominal "Pedro".

(4) Pedro$_i$ sofreu um acidente e $\emptyset_i$ foi para o hospital.

Já a referência catafórica, ou catáfora, ocorre quando o antecedente é posterior a sua referência, como na sentença (5), abaixo, onde o antecedente, isto é, "Maria", ocorre após o pronome "a".

(5) Quando Pedro a$_i$ encontrou, Maria$_i$ estava morta na banheira com os pulsos cortados.

Finalmente, a referência anafórica ou anáfora, ocorre quando o antecedente precede a referência, como no texto (6), abaixo, onde as anáforas "Ela" e "o" co-referem com os sintagmas nominais "Maria" e "um giz de cera", respectivamente. Neste tipo de referência, a expressão de co-referência é denominada anáfora.

(6) Pedro deu um giz de cera$_j$ para Maria$_i$.
Ela$_i$ o$_j$ usou para fazer um desenho.

O antecedente anafórico também pode estar implícito no texto, como no texto (7), abaixo [Fáv91]

(7) Minha filha vai se casar.
Ele é médico.

onde nosso conhecimento do mundo permite inferir que o antecedente do pronome "Ele" é o "o noivo". A figura 2.1 sintetiza a classificação apresentada.

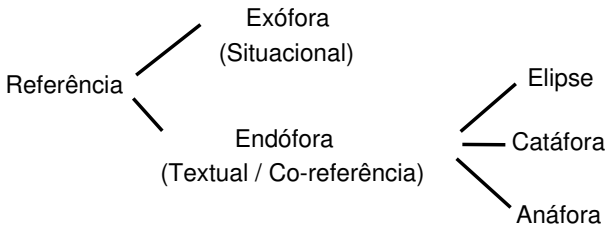


Figura 2.1: Classificação das Referências [HH76].

Dentre os três tipos de referência endofórica apresentados, serão abordadas em mais detalhes as referências anafóricas, tratadas pelo algoritmo proposto neste trabalho.

O fenômeno da anáfora é muito freqüente na produção discursiva, pois ele é fundamental para a coesão de um texto. Portanto, esse fenômeno é essencial também para o entendimento global do texto, ou seja, para sua coerência. A referência anafórica é um fenômeno habitualmente associado aos pronomes, especialmente aos pronomes pessoais. As anáforas, cuja expressão anafórica é um pronome, são denominadas anáforas pronominais.

As expressões de co-referência também podem ser classificadas como intra-sentenciais ou inter-sentenciais. As anáforas intra-sentenciais ocorrem quando o referente está na mesma sentença da anáfora [All95], como na sentença (8), abaixo, onde a anáfora "se" e seu antecedente, ou seja, o sintagma nominal "Pedro", estão na mesma sentença.

(8) Pedro$_i$ feriu-se$_i$ com uma faca.

Já as anáforas inter-sentenciais ocorrem quando a anáfora e seu antecedente estão em sentenças distintas [All95], como no texto (9), a seguir, onde a anáfora "Ele" e seu antecedente, o sintagma nominal "Pedro", estão em sentenças diferentes.

(9) Pedro$_i$ bebeu demais na festa ontem.
Ele$_i$ ficou tão bêbado que foi carregado para casa pelos amigos.

Quando várias anáforas se referem a um mesmo antecedente, elas constituem uma cadeia anafórica [MBDF89]. Assim, o texto (10), a seguir, apresenta uma cadeia anafórica composta pelo antecedente "Pedro" e pelas anáforas "se" e "lo", respectivamente.

(10) Pedro$_i$ se$_i$ suicidou ontem ao chegar em casa depois do trabalho.
Chamaram uma ambulância, mas quando ela chegou ao hospital era tarde demais para salvá-lo$_i$.

Na seção seguinte, serão apresentados alguns problemas referentes à resolução de anáforas, em particular as anáforas pronominais. Além disso, serão apresentados alguns fatores empregados na maioria das abordagens existentes para a resolução de anáforas, como as restrições sintáticas que foram implementadas neste trabalho.

2.2 Resolução de anáforas

A resolução de anáforas envolve um processo no qual o leitor ou ouvinte, interpretando o texto ou discurso, constrói um modelo mental, sintetizando as informações do texto ou discurso com o conhecimento que ele possui a respeito do mundo [Smi91]. Assim, ele emprega estratégias distintas na resolução de cada anáfora, envolvendo conhecimentos variados, como informações sintáticas, semânticas ou pragmáticas.

Em muitos casos, o antecedente anafórico pode ser determinado a partir de características morfossintáticas, como na sentença (11), abaixo:

(11) Pedro$_i$ e Maria morreram, pois ele$_i$ não conseguiu parar o carro antes do penhasco.

O sintagma nominal "Pedro" é o antecedente do pronome "ele", pois é o único candidato anterior ao pronome que concorda em gênero e número com a anáfora.

A estrutura sintática da sentença também impõe restrições na escolha do antecedente. Na sentença (12), a seguir, o emprego de certas restrições sintáticas, que serão apresentadas na seção 2.2.1, permite a escolha do antecedente correto para o pronome "si".

(12) Pedro gostou da foto que Manuel$_i$ tirou de si$_i$ mesmo.

Há casos onde é necessário o emprego de conhecimento semântico na resolução da anáfora, como nas sentenças (13) e (14), abaixo. Apesar dessas sentenças possuírem estruturas sintáticas idênticas, elas possuem cadeias anafóricas distintas.

(13) Pedro tirou o cadáver$_i$ do caixão e o$_i$ embalsamou.

(14) Pedro tirou o cadáver do caixão$_i$ e o$_i$ envernizou.

Em sentenças mais complexas, o emprego de conhecimento pragmático torna-se necessário para a determinação do antecedente, como na sentença (15), a seguir. Nessa sentença, a interpretação correta do pronome "ele" somente é possível pois sabe-se que um assalto envolve a participação de um ladrão.

(15) Fui assaltado à caminho do trabalho e chamei a polícia, mas ele conseguiu escapar.

Portanto, o conhecimento sintático é indispensável na escolha do antecedente anafórico, mas há casos onde o emprego isolado desse tipo de conhecimento não é suficiente [Mit03].

Existem diversas abordagens computacionais que visam determinar o antecedente anafórico, usando diferentes tipos de conhecimento e estratégias lingüísticas, usualmente denominadas fatores de resolução anafórica [Mit03]. Essas abordagens geralmente constróem uma lista de possíveis referentes para a anáfora, a partir da qual é escolhido o melhor candidato, com o emprego de fatores de resolução, como restrições e preferências.

As restrições definem propriedades que devem ser satisfeitas pelo candidato para que ele seja considerado um possível antecedente. Restrições tipicamente usadas na resolução das anáforas são concordância de gênero e número, restrições de c-comando e restrições semânticas. Portanto, elas são utilizadas como um filtro, eliminando candidatos que não as satisfazem [Mit03].

Já as preferências favorecem candidatos que possuam determinadas características. Elas geralmente são empregadas através de regras heurísticas, e visam ordenar a lista dos candidatos favorecendo aqueles que satisfaçam as preferências escolhidas [Mit03]. As preferências usualmente empregadas são o paralelismo sintático (candidatos com a mesma função sintática que a anáfora são favorecidos), preferência pela entidade central do discurso (o candidato que está no centro do discurso tem preferência sobre os demais) e preferência por candidatos mencionados na sentença mais próxima à anáfora [Mit03].

Na próxima seção, serão apresentadas as restrições sintáticas utilizadas neste trabalho.

2.2.1 Restrições de Reinhart

Existem várias restrições sintáticas para a resolução de anáforas, que visam resolver o seguinte problema: a relação de co-referência não ocorre de maneira arbitrária. Assim, não é suficiente considerar apenas o contexto ou a semântica da sentença para permitir a relação de co-referência. Algumas propriedades estruturais da sentença impõem restrições à relação de co-referência [Car89]. As restrições de Reinhart, utilizadas neste trabalho, são fundamentadas em torno do conceito de c-comando (*constituent-command*), que é assim definido [Rei83]:

- *O nó A c-comanda o nó B se e somente se o primeiro nó com ramificação, α, que domina*[3] *o nó A, também domina o nó B, ou é imediatamente dominado por um nó β, que possui a mesma categoria sintática do nó α.*

As restrições relevantes para a resolução de anáforas pronominais que foram implementadas neste trabalho são as seguintes [Rei83]:

1. A co-referência é proibida caso o pronome c-comande o sintagma nominal;
2. A co-referência é permitida caso o pronome seja reflexivo ou recíproco, o sintagma nominal c-comande o pronome e esteja dentro da Categoria de Governo Mínima (*Minimal Governing Category* - MGC)[4] do pronome;
3. A co-referência é proibida caso o pronome não seja reflexivo ou recíproco, o sintagma nominal c-comande o pronome e esteja dentro da Categoria de Governo Mínima do pronome.

Considere as sentenças abaixo:

(16) Ele sentou perto de Pedro.

(17) Maria gosta de si.

(18) Maria gosta dela.

As arvores sintáticas das sentenças (16), (17) e (18) são apresentadas nas figuras 2.2, 2.3 e 2.4, respectivamente.

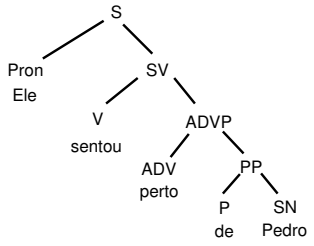


Figura 2.2: Árvore sintática da sentença (16).

Na sentença (16), de acordo com a restrição 1, a co-referência entre o sintagma nominal "Pedro" e o pronome "Ele" é proibida, pois o pronome c-comanda o sintagma nominal, ou seja, o primeiro nó que se ramifica e que domina "Ele", o nó *S*, também domina "Pedro".

[3]Um nó α é dominado por todos os seus ancestrais [All95].

[4]A categoria de governo mínima do nó α é definida como o nó raiz da sentença, ou sintagma nominal ancestral do nó α, que domina o sujeito da sentença.

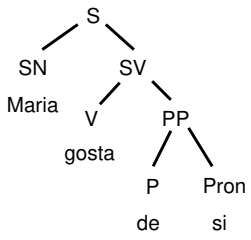


Figura 2.3: Árvore sintática da sentença (17).

Já na sentença (17), a restrição 2 permite que a co-referência ocorra entre o sintagma nominal "Maria" e o pronome "si", pois o sintagma nominal c-comanda o pronome, o pronome é reflexivo, e o sintagma nominal se encontra dentro da MGC do pronome.

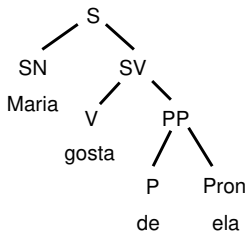


Figura 2.4: Árvore sintática da sentença (18).

Finalmente, na sentença (18) a restrição 3 não permite a co-referência entre o sintagma nominal "Maria" e o pronome "ela", pois o sintagma nominal c-comanda o pronome, o pronome não é reflexivo, e o sintagma nominal se encontra dentro da MGC do pronome.

2.3 Trabalhos relacionados

Abordagens que utilizam restrições sintáticas, como o algoritmo de Lappin e Leass [LL94] e o algoritmo de Hobbs [Hob78], apresentam bons resultados para a língua inglesa devido a eficácia dessas restrições na seleção dos candidatos intra-sentenciais [Mit03].

Na busca por trabalhos relacionados à resolução de referências anafóricas pronominais, constatou-se a escassez de estudos específicos sobre resolução de pronomes na língua portuguesa.

Nesta seção, são apresentados alguns trabalhos relacionados à resolução de anáforas. Dado que o algoritmo proposto neste trabalho baseia-se principalmente em conhecimentos sintáticos, serão abordados os trabalhos que empregam essencialmente esse tipo de conhecimento lingüístico.

Nas seções 2.3.1 e 2.3.2 são apresentados os algoritmos de *Centering* [GWJ95, BFP87] e Hobbs [Hob78] para resolução de anáforas na língua inglesa. Por último, na seção 2.3.3, será apresentado o algoritmo proposto em [PMP+01] para a resolução de anáforas em espanhol.

2.3.1 Algoritmo de *Centering*

O algoritmo de *Centering*, proposto em [BFP87], é uma extensão da Teoria de *Centering* [GWJ95] para a resolução de anáforas pronominais. A Teoria de *Centering* apresenta regras e restrições que regem a conexão entre as sentenças de um discurso. Essa teoria propõe a existência de uma entidade central, denominada centro do discurso, que é o assunto principal do discurso.

O fluxo de idéias entre as sentenças que compõem um discurso deve ser coerente. Um discurso deve apresentar coerência global, entre os seus diversos segmentos, e coerência local, interna a um segmento.

Um segmento de discurso consiste de uma seqüência de sentenças U_1, U_2,...,U_n. O centro retrospectivo (*Backward-looking center*), denominado $C_b(U_n)$, representa a entidade que está no centro do discurso após a interpretação da sentença U_n. Os centros prospectivos (*Forward-looking centers*), denominados $C_f(U_n)$, são uma lista ordenada contendo todas as entidades mencionadas em U_n, que são candidatas a centro da próxima sentença. O algoritmo de *Centering* propõe que os centros prospectivos sejam ordenados de acordo com o papel gramatical de cada entidade. O primeiro elemento de $C_f(U_n)$ é chamado próximo centro preferencial (*Preferred center*), ou $C_p(U_n)$.

A Teoria de *Centering* define um conjunto de transições, supondo que em um mesmo segmento do discurso o centro tenderá a permanecer inalterado. Além disso, existe um conjunto de restrições e regras, apresentadas em [GWJ95, BFP87], que permitem a escolha do antecedente das anáforas pronominais.

As principais restrições afirmam que uma sentença possui um único centro ($C_b(U_n)$), e o centro da próxima sentença ($C_b(U_{n+1})$) é definido como o elemento melhor posicionado na lista de centros prospectivos da sentença anterior ($C_f(U_n)$). O algoritmo emprega duas regras no processo de resolução. A primeira afirma que se algum elemento da lista de centros prospectivos da sentença anterior ($C_f(U_n)$) for o referente de um pronome da próxima sentença (U_{n+1}), então o centro da próxima sentença ($C_b(U_{n+1})$) também será antecedente de algum pronome. A segunda regra impõe uma ordem de preferência nas transições definidas, retratando a preferência em manter o centro dentro de um mesmo segmento do discurso.

O processo de resolução do algoritmo primeiramente gera uma lista dos C_b–C_f contendo todas as combinações dos possíveis antecedentes à anáfora. Essa lista é então

filtrada por restrições sintáticas, semânticas e pelas regras e restrições da Teoria de *Centering*, apresentadas anteriormente. Após essa etapa, as possíveis transições do centro são ordenadas, de acordo com a segunda regra descrita anteriormente. Assim, o antecedente da anáfora será o candidato que provoca a transição de centro preferencial, de acordo com a segunda regra apresentada anteriormente, desde que a primeira regra e as restrições de co-referência empregadas (concordância de gênero, número, restrições semânticas) não sejam violadas [JHM00].

Uma avaliação manual do algoritmo *Centering* é apresentada em [Wal89]. Essa avaliação foi realizada em um corpus composto por 281 sentenças extraídas de um texto jornalístico, um capítulo de uma obra literária e diálogos orientados à tarefa. O algoritmo de *Centering* apresentou taxa de acerto 77,6%.

2.3.2 Algoritmo de Hobbs

O algoritmo de Hobbs caminha na árvore sintática da sentença à procura de um sintagma nominal com mesmo gênero e numero que o pronome [Hob78]. Logo, ele depende da estrutura sintática correta das sentenças do texto [JHM00].

A busca pelo antecedente da anáfora se inicia na sentença onde ela ocorre, a partir do nó da árvore sintática que a representa. O algoritmo faz uma busca em largura (*breadth-first*), da esquerda para direita, subindo em direção à raiz da árvore. Caso não seja encontrado nenhum antecedente intra-sentencial, a busca continuará na sentença anterior a partir da raiz, em largura, da esquerda para direita. Algumas das restrições sintáticas e das preferências na resolução de anáforas são implicitamente empregadas pelo algoritmo através da forma que a busca pelo antecedente na árvore é realizada [JHM00].

O algoritmo foi avaliado por Hobbs em 300 sentenças extraídas de três textos diferentes: um texto literário, um texto técnico de arqueologia e outro jornalístico. O percentual de acerto reportado foi de 88,3%. Esse algoritmo também foi avaliado em [Wal89], juntamente com o algoritmo de *Centering* e usando o mesmo corpus, apresentando uma taxa de acerto de 81,1%.

2.3.3 Algoritmo de resolução anáforas em espanhol

Um algoritmo que resolve anáforas pronominais em espanhol é apresentado em [PMP+01]. Esse algoritmo visa identificar o antecedente de pronomes em terceira pessoa, demonstrativos, reflexivos e nulos[5]. Ele utiliza a estrutura sintática gerada pelo analisador sintático *Slot Unification Parser* [FPM98], que produz uma análise parcial das sentenças.

[5]Pronomes nulos são pronomes elípticos com função de sujeito [PMP+01].

As preferências e restrições sintáticas utilizadas pelo algoritmo e apresentadas em [PMP+01], empregam diferentes tipos de conhecimento (léxico, morfológico e sintático). Foi definido um conjunto distinto de preferências para cada tipo de anáfora, através do estudo do corpus usado na avaliação do algoritmo. As restrições sintáticas empregadas são fundamentadas nas restrições de Reinhart [Rei83], e nas restrições propostas por Lappin e Leass [LL94].

O processo de resolução consiste primeiramente em identificar o tipo da anáfora, ou seja, se ela é um pronome pessoal, demonstrativo, reflexivo ou nulo. Em seguida, é gerada a lista dos possíveis candidatos; para os pronomes reflexivos somente são considerados os sintagmas nominais que ocorrem na mesma sentença da anáfora. Já para os demais pronomes, são considerados todos os sintagmas nominais, até quatro sentenças anteriores à sentença que contém a anáfora. Os candidatos então são filtrados, com o emprego das restrições sintáticas e da verificação da concordância morfológica. Após a filtragem dos candidatos, caso exista apenas um candidato remanescente, ele será escolhido como antecedente da anáfora; caso contrário, o conjunto de preferências definido para o tipo da anáfora sendo resolvida será aplicado aos demais candidatos. Assim, restando apenas um candidato, ele será o antecedente da anáfora; caso contrário serão aplicadas mais três preferências comuns a todos os tipos de anáforas. São elas (em ordem de prioridade): sintagma nominal mencionado com mais freqüência no texto; candidatos que ocorrem repetidas vezes com o verbo da anáfora; e o sintagma nominal mais próximo à anáfora. Finalmente, após o emprego dessas três últimas preferências, o antecedente é escolhido.

Esse algoritmo foi avaliado em um corpus composto por manuais técnicos e textos literários, com 1677 ocorrências de pronomes, alcançando uma taxa de sucesso de 76,8%.

2.4 Considerações

Neste capítulo foi apresentada uma visão geral do fenômeno lingüístico da referência. Foram abordados alguns tipos de referência, como as endofóricas e as exofóricas, com ênfase nas referências anafóricas pronominais. Também foram apresentados alguns problemas na resolução de anáforas pronominais, e os fatores de resolução mais comumente empregados para determinar o antecedente anafórico, como as restrições e as preferências.

Além disso, foram apresentados três algoritmos para resolução de referências anafóricas: o algoritmo de Hobbs e o algoritmo proposto em [PMP+01], que são baseados em restrições sintáticas, e o algoritmo de *Centering*, que é fundamentado na Teoria de *Centering*.

No próximo capítulo, serão apresentados o algoritmo original de Lappin e Leass e sua adaptação para o português. O algoritmo visa resolver anáforas pronominais de terceira pessoa e pronomes reflexivos e recíprocos.

Capítulo 3

Algoritmo de Lappin e Leass

O algoritmo de Lappin e Leass, ou RAP (*Resolution of Anaphora Procedure*), como é mais conhecido, visa resolver anáforas inter-sentenciais e intra-sentenciais pronominais, em terceira pessoa, na língua inglesa. O algoritmo é baseado em um sistema de pesos, atribuídos de acordo com a estrutura sintática da sentença, e em uma representação simples do modelo do discurso.

Neste capítulo são apresentados o algoritmo de Lappin e Leass original e sua adaptação para o português. Além disso, é dado um exemplo ilustrando o funcionamento do algoritmo adaptado.

3.1 O algoritmo original

Os componentes essenciais do algoritmo de Lappin e Leass são [LL94]:

- Um filtro intra-sentencial sintático [LM90b, LM90a] para descartar dependências anafóricas de um pronome a um sintagma nominal, com base em fundamentos sintáticos;
- Um filtro morfológico para excluir dependências anafóricas de um pronome a um sintagma nominal, devido a discordâncias de gênero, número ou pessoa;
- Um algoritmo de ligação [LM90a] para identificar os possíveis antecedentes de um pronome reflexivo na mesma sentença;
- Um procedimento para identificar pronomes semanticamente vazios;
- Uma função para atribuir os fatores de saliência adequados a cada sintagma nominal, como paralelismo sintático e sujeito, dentre outros;

- Uma função de decisão para selecionar o antecedente de um pronome proveniente de uma lista de candidatos.

O algoritmo de ligação identifica os candidatos intra-sentenciais a antecedente dos pronomes reflexivos/recíprocos; já o filtro sintático elimina os candidatos intra-sentenciais a antecedentes de pronomes de terceira pessoa não reflexivos/recíprocos, com os quais co-referência não é permitida. Para os candidatos remanescentes são calculados os valores de saliência, como descrito na seção 3.1.1. Assim, o candidato escolhido como antecedente do pronome será aquele que possuir maior valor de saliência e, em caso de empate, será escolhido o candidato mais próximo ao pronome. Tanto o filtro sintático como o algoritmo de ligação analisam a estrutura sintática da sentença onde ocorre o pronome para determinar se a co-referência é permitida.

A identificação de pronomes pleonásticos, que é realizada após a extração das anáforas pronominas, engloba somente o emprego do pronome *it*. O uso desse pronome é classificado como pleonástico caso a estrutura da sentença onde ele ocorre coincida com um dos padrões de estruturas sentenciais descritas em [LL94]. Por exemplo, na sentença (19), a seguir, o emprego do pronome *it* é considerado semanticamente vazio, pois a estrutura da sentença coincide com o padrão *It is thanks to SN that S*. O algoritmo descarta todos os pronomes identificados como pleonásticos, pois eles não possuem antecedente.

(19) *It is thanks to your dedication that our team won the race.*
(Graças a nossa dedicação, nosso time ganhou a corrida.)

O RAP utiliza a representação gramatical gerada pelo analisador sintático desenvolvido por McCord [McC90, McC93].

3.1.1 Fatores de saliência

O RAP utiliza um sistema de pesos baseado em características sintáticas – nenhuma informação semântica é usada no processo de resolução. O algoritmo possui basicamente dois tipos de operação: atualização do modelo de discurso e resolução do pronome. Quando um sintagma nominal, que introduz uma nova entidade no discurso, é encontrado, uma representação para ele é criada e seu valor de saliência é calculado. O valor de saliência é dado pela soma de todos os fatores de saliência que se aplicam ao sintagma nominal. Os valores iniciais dos fatores de saliência são apresentados na tabela 3.1.

Os fatores de saliência retratam a preferência na escolha do antecedente, de acordo com seu papel gramatical, refletida no valor do peso de cada fator.

Tabela 3.1: Valores dos fatores de saliência iniciais [LL94].

Fatores de Saliência	**Valores**
Sentença atual	100,0
Sujeito	80,0
Construção existencial	70,0
Objeto direto	50,0
Objeto indireto	40,0
Ênfase não adverbial	50,0
Sintagma nominal não contido	80,0

As sentenças a seguir ilustram a atribuição dos fatores de saliência às expressões em negrito:

- *Sujeito:*
  (20) **Os fiscais** procederam à prova com atraso.
- *Construção existencial:*
  (21) Havia **uma casa azul** ao lado da padaria.
- *Objeto direto:*
  (22) Mariana transformava **a minha vida**.
- *Objeto indireto:*
  (23) Duvidava **da riqueza da terra**.
- *Ênfase não adverbial:*
  (24) Não saímos por causa **da chuva**.
- *Sintagma nominal não contido:*
  (25) **Pedro** comprou **um carro**.

A atribuição dos quatro primeiros fatores é bastante clara, mas a atribuição dos dois últimos fatores merece uma explicação adicional. O fator de saliência *Ênfase não adverbial* é atribuído à entidades do discurso que não estejam contidas numa locução adverbial preposicionada demarcada. Assim, na sentença (24), o sintagma nominal "chuva" recebe esse fator pois, apesar de estar contido numa locução adverbial preposicionada, ela não é demarcada. Por outro lado, na sentença (26), a seguir, o fator de saliência *Ênfase não adverbial* não é atribuído ao sintagma nominal "da padaria", pois ele está contido na locução adverbial preposicionada demarcada "Ao lado da padaria".

(26) Ao lado da padaria, Pedro foi roubado.

O fator de saliência *Sintagma nominal não contido* é atribuído à entidades do discurso que não estejam contidas em outro sintagma nominal. Assim, na sentença (25), os sintagmas nominais "Pedro" e "um carro" recebem esse fator. Por outro lado, na sentença (27), a seguir, os sintagmas nominais "usuário do carro" e "carro" não recebem o fator de saliência *Sintagma nominal não contido*, pois estão contidos nos sintagmas nominais "O manual de usuário do carro" e "usuário do carro", respectivamente. Entretanto, esse fator é atribuído ao sintagma nominal "O manual de usuário do carro" como um todo, pois ele não está contido em nenhum outro sintagma nominal.

(27) **O manual de usuário do carro** está no porta-luvas.

O fator de saliência *Sentença atual* é atribuído a todos os sintagmas nominais presentes na sentença que está sendo processada. O valor de saliência atribuído a toda entidade do modelo do discurso é dividido pela metade no início do processamento de cada sentença. O mecanismo de degradação dos valores de saliência e o fator de saliência *Sentença atual* visam priorizar candidatos à co-referência mencionados recentemente, já que os candidatos que ocorrem na sentença atual e nas sentenças mais próximas a ela tendem a possuir valores de saliência maiores.

No processo de escolha do antecedente, são considerados mais dois fatores de saliência: o fator de saliência *Paralelismo sintático* e *Catáfora*. O primeiro prioriza candidatos que apresentam paralelismo sintático com o pronome que está sendo resolvido, como no texto (28), abaixo:

(28) **Pedro** foi a concessionária com Bruno.
Ele comprou um carro.

O sintagma nominal "Pedro" receberá o fator de saliência *Paralelismo sintático* na resolução do pronome "Ele", pois ambos exercem a função sintática de sujeito.
O segundo fator penaliza catáforas, como na sentença (29), abaixo:

(29) Ela estava preparando o almoço quando **Maria** chegou.

O sintagma nominal "Maria" será penalizado na resolução do pronome "Ela" pois ocorre após o pronome. A atribuição desses fatores de saliência adicionais depende do pronome que está sendo resolvido e é temporária, já que seu peso é adicionado ao valor de saliência da entidade do discurso somente durante a resolução do pronome. Os valores desses fatores de saliência são apresentados na tabela 3.2.

Tabela 3.2: Fatores de saliência adicionais [LL94].

Fatores de Saliência	Valores
Paralelismo sintático	35,0
Catáfora	-175,0

Os valores de todos os fatores de saliência foram obtidos empiricamente por Lappin e Leass, visando otimizar o desempenho do algoritmo no corpus utilizado para avaliá-lo. O corpus era composto por manuais de computadores, na língua inglesa.

É importante ressaltar que uma entidade do discurso poderá receber mais de um fator de saliência, atribuídos de acordo com a estrutura da sentença e com sua função sintática. Detalhes sobre essas atribuições serão ilustrados através de um exemplo, apresentado na seção 3.2.2.

3.1.2 Classes de equivalência

As classes de equivalência agrupam os pronomes que possuem um mesmo referente; assim, o referente e todos os pronomes que se referem a ele são agrupados na mesma classe. O fator de saliência de uma classe de equivalência é dado pela soma dos fatores de todos os seus membros. Considere, por exemplo, o texto (30), a seguir:

(30) Pedro$_i$ comprou uma casa.
Ele$_i$ fez o pagamento à vista.
A nova casa dele$_i$ é azul.

Nesse texto, os pronomes "Ele" e "dele" pertencem à classe de equivalência do sintagma nominal "Pedro", pois ambos possuem esse sintagma nominal como antecedente.

3.2 O algoritmo de Lappin e Leass adaptado para o português

O algoritmo desenvolvido e avaliado neste trabalho é baseado no RAP. O algoritmo aqui proposto possui todos os principais componentes do algoritmo original, com as seguintes diferenças [CC05a, CC05b]:

- O filtro sintático e o algoritmo de ligação foram substituídos pelas restrições de co-referência propostas por Reinhart [Rei83]. Uma análise dos exemplos apresentados

em [LL94, LM90b, LM90a] mostrou que as restrições de Reinhart seriam eficientes para resolver os casos apresentados pelos autores do algoritmo original;

- O analisador sintático utilizado foi o PALAVRAS [Bic00];
- A ferramenta Xtractor [GVGQ03] foi empregada para converter a saída do analisador sintático em XML (*Extensible Markup Language*[1]). Essa ferramenta foi utilizada visando facilitar a extração das informações lingüísticas do corpus analisado pelo PALAVRAS;
- Não foi implementada a identificação do uso pleonástico do pronome *it*, já que esse fenômeno não ocorre no português;
- Não foi implementado um tratamento para catáforas.

3.2.1 Processo de resolução

Os passos abaixo descrevem o processo de resolução de co-referência, de acordo com o algoritmo desenvolvido [CC05a, CC05b]:

- No início do processamento de uma nova sentença, degradar o fator de saliência de todas as classes de equivalência, ou seja, dividir o fator de saliência de todas as classes por dois;
- Extrair todos os possíveis candidatos na sentença;
- Criar classes de equivalência para todos os candidatos que não pertencem a nenhuma das classes existentes. Nesses casos, os fatores de saliência apropriados serão atribuídos ao candidato segundo a tabela 3.1. O valor de saliência do candidato já incluído numa classe de equivalência será o fator de equivalência da classe à qual ele pertence;
- Para cada pronome da sentença:
  - Calcular o fator de saliência do pronome, utilizando os valores da tabela 3.1;
  - Gerar lista com possíveis candidatos para pronomes de terceira pessoa:
    - Extrair candidatos, utilizando uma janela de 4 sentenças[2], que concordem em gênero e número com o pronome;

[1] *Disponível em http://www.w3.org/XML/*

[2] Estudos empíricos mostram que a distância entre a anáfora e seu antecedente geralmente é de 2 ou 3 sentenças [MTB97, Hob78]

- Gerar lista com possíveis candidatos para pronomes reflexivos/recíprocos:
  - Extrair somente candidatos intra-sentenciais que concordem em gênero e número com o pronome;
- Aplicar as restrições de Reinhart aos candidatos intra-sentenciais remanescentes. Caso o candidato seja rejeitado pelas restrições, todas as entidades da sua classe de equivalência serão excluídas da lista de candidatos;
- Atribuir pesos adicionais aos candidatos, de acordo com a tabela 3.2;
- Selecionar o candidato com o maior fator de saliência. Se ocorrer empate, escolher o candidato mais próximo ao pronome;
- Incluir o pronome na classe de equivalência do candidato selecionado.

3.2.2 Exemplo de execução do algoritmo

A execução do algoritmo é ilustrada a seguir. Considere as seguintes sentenças:

(31) Pedro viu um lindo carro na concessionária.

(32) Ele mostrou-o a Rafael.

(33) Ele comprou-o.

As árvores sintáticas simplificadas das sentenças (31), (32) e (33), geradas pelo PALAVRAS [Bic00], são apresentadas nas figuras 3.1, 3.2 e 3.3, respectivamente.

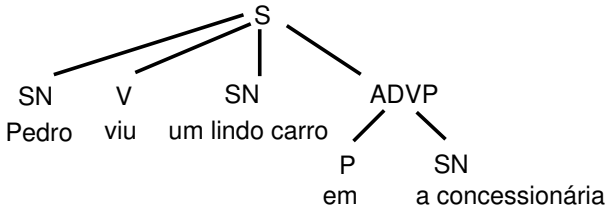


Figura 3.1: Árvore sintática simplificada da sentença (31).

No processamento da sentença (31), primeiro são coletados todos os candidatos a referente, isto é, "Pedro", "um lindo carro" e "a concessionária"; a seguir, seus valores de saliência são calculados, e novas classes de equivalência são criadas para representá-los. Ao primeiro é atribuído o valor de saliência 310, pois "Pedro" tem função de sujeito (80), é um sintagma nominal não contido (80), não está contido numa locução adverbial preposicionada demarcada (50), e está presente na sentença que está sendo processada

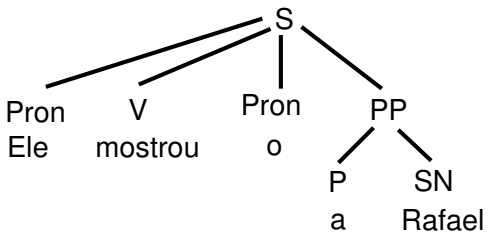


Figura 3.2: Árvore sintática simplificada da sentença (32).

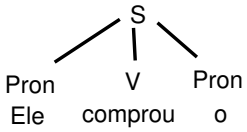


Figura 3.3: Árvore sintática simplificada da sentença (33).

(100). Ao segundo sintagma nominal, isto é, "um lindo carro", é atribuído o valor de saliência 280, pois ele exerce a função de objeto direto (50), é um sintagma nominal não contido (80), não está contido numa locução adverbial preposicionada demarcada (50), e está presente na sentença que está sendo processada (100). Ao último sintagma nominal, "a concessionária", é atribuído o valor de saliência 230, pois é um sintagma nominal não contido (80), não está contido numa locução adverbial preposicionada demarcada (50), e está presente na sentença que está sendo processada (100). As classes de equivalência criadas no processamento da sentença (31) são ilustradas na figura 3.4(a).

Ao iniciar o processamento da sentença (32), todas as classes de equivalência tem seus valores de saliência divididos por dois, como mostra a figura 3.4(b). O possível candidato extraído nessa sentença é "Rafael", cujo valor de saliência é 270, pois tem a função de objeto indireto (40), é sintagma nominal não contido (80), não está contido numa locução adverbial preposicionada demarcada (50), e está na sentença que está sendo processada (100). A figura 3.4(c) mostra o estado atual do modelo do discurso.

A seguir, os pronomes da sentença são resolvidos. Na resolução do pronome "Ele", não há candidatos intra-sentenciais, já que a escolha do sintagma nominal "Rafael" resultaria em catáfora. O possíveis candidatos inter-sentenciais são "Pedro" e "um lindo carro", pois ambos concordam em gênero e número com o pronome que está sendo resolvido, e estão contidos em uma janela de 4 sentenças. Os valores da tabela 3.2 são então aplicados para calcular os valores de saliência finais dos candidatos. O candidato "Pedro" tem adicionado 35 ao seu valor de saliência, pois ele exerce a mesma função sintática do pronome, isto é, ambos são sujeitos; já o candidato "um lindo carro" não tem nenhum acréscimo ao seu valor de saliência. Note que a atribuição dos fatores de saliência da tabela 3.2 é

(a)

Referente	Anáforas	valor
Pedro		310
um lindo carro		280
concessionária		230

(b)

Referente	Anáforas	valor
Pedro		155
um lindo carro		140
concessionária		115

(c)

Referente	Anáforas	valor
Pedro		155
um lindo carro		140
concessionária		115
Rafael		270

(d)

Referente	Anáforas	valor
Pedro	Ele(1)	465
um lindo carro		140
concessionária		115
Rafael		270

(e)

Referente	Anáforas	valor
Pedro	Ele(1)	465
um lindo carro	o(1)	420
concessionária		115
Rafael		270

(f)

Referente	Anáforas	valor
Pedro	Ele(1)	232,5
um lindo carro	o(1)	210
concessionária		57,5
Rafael		135

(g)

Referente	Anáforas	valor
Pedro	Ele(1), Ele(2)	542,5
um lindo carro	o(1)	210
concessionária		57,5
Rafael		135

(h)

Referente	Anáforas	valor
Pedro	Ele(1), Ele(2)	542,5
um lindo carro	o(1), o(2)	490
concessionária		57,5
Rafael		135

Figura 3.4: Evolução do estado do modelo do discurso das sentenças (31), (32) e (33).

temporária; assim, o acréscimo de 35 ao valor de saliência do candidato "Pedro" somente será considerado durante a resolução do pronome "Ele". Portanto, o candidato "Pedro" é escolhido como antecedente para o pronome "Ele", pois possui o maior valor de saliência. O modelo do discurso é agora atualizado, e o fator de saliência do pronome é calculado. O pronome "Ele" tem o valor de saliência 310, pois é sujeito (80), é um sintagma nominal não contido (80), não está contido numa locução adverbial preposicionada demarcada (50), e está na sentença que está sendo processada (100). Posteriormente, o pronome é adicionado à classe de equivalência do candidato e o valor de saliência do pronome é adicionado ao valor de saliência da classe de equivalência do candidato escolhido. A figura 3.4(d) mostra o modelo do discurso após essa atualização.

O próximo pronome a ser resolvido é o pronome "o". O possível candidato a antecedente intra-sentencial é "Ele", já que a escolha do sintagma nominal "Rafael" resultaria em catáfora. Note que o pronome "Ele" é considerado candidato a co-referência, pois foi resolvido anteriormente e, portanto, é tratado como o sintagma nominal ao qual ele faz referência. Portanto, a co-referência com esse candidato é impedida pela restrição 3 de Reinhart. Logo, o único candidato inter-sentencial é "um lindo carro", pois concorda em gênero e número com o pronome. O candidato "Pedro" não é considerado no próximo passo, pois pertence a classe de equivalência do pronome "Ele", que foi eliminado após a verificação das restrições de Reinhart. Assim, o sintagma nominal "um lindo carro" é o único candidato a referente. Os valores da tabela 3.2 são então aplicados para calcular o

valor de saliência final do candidato, que terá um acréscimo de 35 ao seu valor de saliência, pois ele exerce a mesma função sintática do pronome, isto é, ambos são objeto direto. Assim, o sintagma nominal "um lindo carro" é escolhido como antecedente do pronome "o". Novamente, o modelo do discurso é atualizado e o valor de saliência do pronome é calculado. O pronome "o" recebe o valor de saliência 280, pois é objeto direto (50), é sintagma nominal não contido (80), não está contido numa locução adverbial preposicionada demarcada (50), e está na sentença que está sendo processada (100). O processo de atualização do valor de saliência da classe de equivalência do candidato escolhido é o mesmo processo descrito anteriormente. A figura 3.4(e) mostra o estado do modelo do discurso atualizado.

Ao iniciar o processamento da sentença (33), todas as classes de equivalência tem seus valores de saliência divididos por dois novamente, como mostra a figura 3.4(f). Como não existem possíveis candidatos nessa sentença, é iniciado o processo de resolução de pronomes. Na resolução do pronome "Ele", não há candidatos intra-sentenciais, pois a escolha de qualquer sintagma nominal ou pronome, na mesma sentença, resultaria em catáfora. Os candidatos inter-sentenciais da sentença (31) são "Pedro" e "um lindo carro"; já os candidatos da sentença (32) são os pronomes resolvidos "Ele", "o" e o sintagma nominal "Rafael". Note que todos os pronomes resolvidos são tratados como o sintagma nominal ao qual se referem; assim, os pronomes "Ele" e "o" são tratados como os sintagmas nominais "Pedro" e "um lindo carro", respectivamente. Os valores da tabela 3.2 são então aplicados para calcular os valores de saliência finais dos candidatos. Os candidatos "Pedro", da sentença (31), e "Ele", da sentença (32), terão um acréscimo de 35 ao seus valores de saliência, pois ambos exercem a mesma função sintática do pronome, ou seja, ambos são sujeito. Assim, haverá um empate entre os candidatos "Pedro" e "Ele", mas o pronome "Ele" será escolhido como antecedente, pois está mais próximo ao pronome que está sendo resolvido. Novamente, o modelo do discurso é atualizado e o valor de saliência do pronome é calculado. O pronome "Ele" recebe o valor de saliência 310, pois exerce a função de sujeito (80), é um sintagma nominal não contido (80), não está contido numa locução adverbial preposicionada demarcada (50), e está presente na sentença que está sendo processada (100). Como o antecedente escolhido é um pronome resolvido anteriormente, o pronome resolvido será adicionado à classe de equivalência do sintagma nominal do pronome candidato escolhido, ou seja, à classe do sintagma nominal "Pedro", como mostra a figura 3.4(g).

O próximo pronome a ser resolvido é o pronome "o". O possível candidato a antecedente intra-sentencial é o pronome resolvido "Ele", mas co-referência com esse candidato é impedida pela restrição 3 de Reinhart. Assim, os candidatos inter-sentenciais "Pedro", da sentença (31), e "Ele", da sentença (32), não serão considerados no próximo passo, pois pertencem a mesma classe de equivalência do pronome "Ele", da sentença (33). Logo, os

candidatos serão os sintagmas nominais "um lindo carro", da sentença (31), e "Rafael", da sentença (32), e o pronome "o" da sentença (32). Aplicando os fatores de equivalência adicionais, segundo a tabela 3.2, os candidatos "o", da sentença (32), e "um lindo carro", da sentença (31), terão um acréscimo de 35 ao seus valores de saliência, pois ambos exercem a mesma função sintática do pronome, ou seja, eles são objeto direto. Assim, haverá empate entre esses dois candidatos, mas o pronome "o" da sentença (32) será o candidato escolhido, pois está mais próximo ao pronome que está sendo resolvido. O modelo do discurso é agora atualizado, e o fator de saliência do pronome é calculado. O pronome "o" recebe o valor de saliência 280, pois ele é objeto direto (50), é sintagma nominal não contido (80), não está contido numa locução adverbial preposicionada demarcada (50), e está na sentença que está sendo processada (100). Como o antecedente escolhido é um pronome resolvido anteriormente, o pronome resolvido será adicionado na classe de equivalência do sintagma nominal do pronome candidato escolhido, ou seja, na classe do sintagma nominal "um lindo carro". A figura 3.4(h) mostra o estado final do modelo do discurso, após o término da execução do algoritmo.

3.3 Considerações

Neste capítulo, o algoritmo de Lappin e Leass e sua adaptação para o português foram apresentados. Além disso, foi apresentado um exemplo ilustrando o funcionamento do algoritmo desenvolvido.

No capítulo seguinte, serão apresentadas as principais ferramentas utilizadas neste trabalho.

Capítulo 4

Ferramentas utilizadas

Neste capítulo, são apresentadas as ferramentas utilizadas nos experimentos: o analisador sintático PALAVRAS [Bic00], que foi usado na análise gramatical dos textos; a ferramenta Xtractor [GVGQ03], que foi usada na conversão da saída do analisador em XML; e a ferramenta MMAX (*Multi-Modal Annotation in XML*) [MS01], que foi usada na marcação manual do corpora.

4.1 PALAVRAS e Xtractor

O algoritmo de Lappin e Leass requer que o texto a ser analisado seja processado previamente por um analisador sintático, já que ele utiliza a estrutura sintática da sentença na resolução das anáforas. Neste trabalho, o analisador sintático utilizado foi o PALAVRAS, que está sendo desenvolvido no Departamento de Língua e Comunicação da University of Southern Denmark, em Odense, por Eckhard Bick e sua equipe, e consta de um analisador sintático automático para o português. O código fonte não está disponível, mas ele pode ser utilizado para demonstrações pela Internet[1].

A extração de qualquer informação sintática do corpus analisado pelo PALAVRAS depende da construção de ferramentas específicas para cada aplicação. Portanto, visando aumentar a aplicabilidade do analisador, foi proposta em [GVGQ03] uma representação da saída do PALAVRAS em XML e a ferramenta Xtractor. A ferramenta converte a saída do analisador em três arquivos XML [ACC+04]: o arquivo *Words*, que contém uma lista das palavras do texto e seus respectivos identificadores; o arquivo *Pos (Part of Speech)*, contendo informações morfossintáticas sobre as palavras do texto; e o arquivo *Chunks*, que contém a estrutura sintática das sentenças.

[1] Disponível em: `http://visl.sdu.dk/visl/pt/`

Considere a seguinte sentença:

(34) Maria e Pedro feriram-se.

A figura 4.1 mostra a análise gerada pelo PALAVRAS. Em cada linha da figura, o primeiro símbolo representa a função sintática ('SUBJ'=sujeito, 'CO'=conjunção, 'P'=predicado, 'ACC'=objeto direto[2]); após o símbolo ':' é indicada a categoria sintática para um grupo de palavras, ou a categoria morfossintática de uma única palavra ('fcl'=oração finita, 'prop'=nome próprio, 'conj-c'=conjunção coordenativa, 'v-fin'=verbo finito, 'pron-pers'=pronome pessoal, dentre outros); entre parênteses estão a forma canônica da palavra e algumas outras marcações, como 'F', para indicar gênero feminino, e 'S', indicando que a palavra está no singular; após o fecha-parênteses está a palavra tal qual ela ocorre no corpus. O sinal de igual no início de cada linha indica o nível da sentença na árvore sintática.

```
STA:fcl
=SUBJ:prop('Maria' F S) Maria
=CO:conj-c('e') e
=SUBJ:prop('Pedro' M S) Pedro
=P:v-fin('ferir' PS/MQP 3P IND <hyfen>) feriram-
=ACC :pron-pers('se' M 3S/P ACC <refl>) se
=.
```

Figura 4.1: Estrutura da árvore sintática da sentença (34).

A formatação usada nas árvores geradas pelo PALAVRAS dificulta a extração das informação sintáticas do corpus analisado; portanto, a extração dessas informações precisa ser customizada para cada aplicação de língua natural [GVGQ03]. Visando simplificar o uso do corpus analisado, o resultado da análise do PALAVRAS é convertido para XML através do Xtractor. Os arquivos *Words*, *Chunks* e *Pos* da sentença (34), gerados pelo Xtractor, são mostrados nas figuras 4.2, 4.3 e 4.4, respectivamente.

O arquivo *Words* contém as palavras como elas ocorrem no texto. Cada palavra é contida em um elemento *word* e recebe um identificador único, como o elemento *word* cujo atributo "id" é "word_1", na figura 4.2, e que se refere a palavra "Maria" da sentença (34).

[2]A lista completa dos símbolos pode ser encontrada em http://visl.sdu.dk/visl/pt/

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE words SYSTEM "words.dtd">
<words>
  <word id="word_1">Maria</word>
  <word id="word_2">e</word>
  <word id="word_3">Pedro</word>
  <word id="word_4">feriram-</word>
  <word id="word_5">se</word>
  <word id="word_6">.</word>
</words>
```

Figura 4.2: Arquivo *Words* da sentença (34).

As informações sintáticas são extraídas a partir do arquivo *Chunks*, de acordo com a necessidade de cada aplicação de língua natural. Essas informações são indicadas pelos atributos do elemento *chunk*. São eles:

- Atributo *ext*: indica a função sintática do *chunk*; por exemplo, objeto direto (ext="acc"), sujeito (ext="subj") e núcleo (ext="h");
- Atributo *form*: indica a forma ou estrutura morfossintática do *chunk*; por exemplo, sintagma nominal (form="np"), substantivo (form="n") e pronome pessoal (form="pron_pers");
- Atributo *span*: refere-se aos elementos do arquivo *Words* representados pelo elemento *chunk*; por exemplo, caso o atributo "form" possua o valor "word_1..word_10" (form="word_1..word_10"), isso significa que o elemento *chunk* representa as palavras "word_1" até "word_10" do texto.

Os elementos *sentence* são os pais dos elementos *chunk* referentes à estrutura sintática da sentença em questão. Esses elementos possuem os atributos "span" e "id". O primeiro possui o identificador das palavras que constituem a sentença, e o segundo é um identificador único atribuído a cada sentença. Portanto, na figura 4.3, o elemento *sentence*, cujo atributo "id" é "sentence_1", e cujo atributo "span" é "word_1..word_6", se refere a toda sentença (34).

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE text SYSTEM "text_ext.dtd">
<text>
  <paragraph id="paragraph_1">
    <sentence id="sentence_1" span="word_1..word_6">
      <chunk id="chunk_1" ext="sta" form="fcl" span="word_1..word_5">
        <chunk id="chunk_2" ext="subj" form="prop" span="word_1"/>
        <chunk id="chunk_3" ext="co" form="conj_c" span="word_2"/>
        <chunk id="chunk_4" ext="subj" form="prop" span="word_3"/>
        <chunk id="chunk_5" ext="p" form="v_fin" span="word_4"/>
        <chunk id="chunk_6" ext="acc" form="pron_pers" span="word_5"/>
      </chunk>
    </sentence>
  </paragraph>
</text>
```

Figura 4.3: Arquivo *Chunks* da sentença (34).

Todas as palavras do texto devem possuir um elemento *word* correspondente no arquivo *Pos*. Esse elemento possui elementos filhos que indicam a informação morfossintática da palavra em questão. Portanto, o elemento "word" cujo atributo "id" é "word_1", na figura 4.4, contém as informações morfossintáticas da palavra no arquivo *Words*, cujo atributo "id" também é "word_1". Caso a palavra não seja reconhecida pelo PALAVRAS, ou ocorra uma extração incorreta pelo Xtractor, o elemento *word* correspondente a ela no arquivo *Pos* terá o texto "unknown". Os atributos relevantes para a implementação deste trabalho são:

- Atributo *gender*: representa o gênero da palavra, cujos possíveis valores são masculino (gender="M"), feminino (gender="F") e masculino/feminino (gender="M-F"), isto é, indeterminado;
- Atributo *number*: indica o número da palavra, que pode ser singular (number="S"), plural (number="P") e singular/plural (number="S-P");
- Atributo *person*: representa a pessoa em que o pronome ou verbo ocorre no texto, como terceira pessoa do plural (person="3P"), ou primeira pessoa do singular (person="1S");
- Atributo *tag*: no caso dos pronomes, esse atributo indica se o pronome é reflexivo (tag="refl" ou tag="si"), ou recíproco (tag="reci").

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE words SYSTEM "wordsPOS.dtd">
<words>
  <word id="word_1">
    <prop canon="Maria" gender="F" number="S"/>
  </word>
  <word id="word_2">
    <conj canon="e">
      <secondary_conj tag="c"/>
    </conj>
  </word>
  <word id="word_3">
    <prop canon="Pedro" gender="M" number="S"/>
  </word>
  <word id="word_4">
    <v canon="ferir">
      <fin tense="PS-MQP" person="3P" mode="IND"/>
      <secondary_v tag="hyfen"/>
    </v>
  </word>
  <word id="word_5">
    <pron canon="se">
      <pers gender="M" number="3S-P" case="ACC"/>
      <secondary_pron tag="refl"/>
    </pron>
  </word>
</words>
```

Figura 4.4: Arquivo *POS* da sentença (34).

4.2 O MMAX

Os corpora dos experimentos foram anotados manualmente com informações de co-referência anafórica. Essa anotação foi realizada com a ferramenta MMAX, que utiliza os arquivos de *Chunks* e *Words*, gerados pelo Xtractor e apresentados na seção 4.1. A marcação gerada pelo MMAX é armazenada separadamente do corpus; assim, é criado um repositório de marcações com apontadores para os elementos do corpus.

O resultado do processo de anotação no MMAX é o arquivo XML de *markables*. Esse arquivo é composto por elementos *markable*, que codificam as informações anotadas em seus atributos. Os atributos podem ser especificados conforme as informações que serão anotadas.

A figura 4.5, referente a anotação da sentença (34) da página 26, ilustra o formato do arquivo de *markables* utilizado. Os atributos do elemento *markable* especificados são:

- Atributo *id*: identificador único do elemento *markable*;
- Atributo *span*: refere-se aos elementos do arquivo de *Words* representados pelo elemento *markable*. Por exemplo, na figura 4.5, o elemento *markable* com atributo "id" igual a "markable_2" e atributo "span" igual a "word_5", se refere ao elemento

word do arquivo *Words* do Xtractor cujo atributo "id" é "word_5", ou seja, se refere ao pronome "se";

- Atributo *form*: identifica o tipo do elemento. Caso o elemento seja um pronome, o atributo terá valor "pronome", como o elemento *markable* da figura 4.5, cujo atributo "id" é "markable_2", que se refere ao pronome "se". Se o elemento for o antecedente de um pronome, o atributo "form" será "antecedent", como o elemento *markable* da figura 4.5, cujo atributo "id" é "markable_1", que se refere a "Maria e Pedro";

- Atributo *pointer*: indica o identificador do elemento *markable* que é antecedente do pronome. Por exemplo, na figura 4.5, o elemento *markable* com atributos "id" igual a "markable_2" e "pointer" igual a "markable_1", indica que o antecedente do pronome em questão, ou seja, o pronome "se", é o elemento *markable* cujo atributo "id" é "markable_1", isto é, o sintagma nominal "Maria e Pedro";

- Atributo *anaphor_type*: corresponde a classificação da anáfora. Os possíveis valores são: "inter_sentential", para anáforas inter-sentenciais; "intra_sentential", para anáforas intra-sentenciais; e "none", quando não há antecedente. Na figura 4.5, a relação de co-referência entre os elementos *markable* com atributos "id" igual a "markable_1" e "id" igual a "markable_2", ou seja, entre o pronome "se" e seu antecedente "Maria e Pedro", é classificada como intra-sentencial, pois o elemento *markable* que corresponde ao pronome possui atributo "anaphor_type" igual a "intra_sentential". Esse atributo foi usado apenas na anotação dos corpora literário e jornalístico.

```
<?xml version="1.0"?>
<markables>
  <markable id="markable_1" span="word_1..word_3" form="antecedent" />
  <markable id="markable_2" span="word_5" form="pronome"
    pointer="markable_1" anaphor_type="intra_sentential" />
</markables>
```

Figura 4.5: Arquivo de markables da sentença (34).

4.3 Considerações

Neste capítulo foram apresentados o analisador sintático utilizado, o PALAVRAS, e a ferramenta Xtractor, que visa facilitar a extração das informações geradas pelo analisador, convertendo a saída do analisador em XML. Além disso, também foi apresentada a ferramenta MMAX, utilizada na anotação manual dos corpora.

No capítulo seguinte, será apresentada a arquitetura do sistema desenvolvido. Também serão detalhados a extração dos sintagmas nominais, dos pronomes e o processamento de sujeitos compostos.

Capítulo 5

Implementação

O paradigma adotado no desenvolvimento deste trabalho foi o orientado a objetos, e a linguagem de implementação escolhida foi Java [HC02b, HC02a, McL01]. A API (*Application Program Interface*) e o suporte nativo ao processamento e manipulação de documentos XML foram os fatores determinantes na escolha da linguagem.

Neste capítulo, são apresentadas a arquitetura e a modelagem utilizada na implementação do algoritmo de Lappin e Leass, adaptado para o português e descrito na seção 3.2.1.

5.1 Arquitetura do Sistema

A arquitetura do software se refere à descrição dos elementos com os quais um sistema é construído, às interações entre estes elementos, aos padrões que guiam a sua composição e às restrições sobre esses padrões [SG96].

A figura 5.1 ilustra a arquitetura do sistema desenvolvido, cujo padrão arquitetural adotado foi o *pipes and filters* [SG96, BMR+96]. Os elementos que compõem a arquitetura e que foram desenvolvidos neste trabalho são:

- Manipulador de Sujeito Composto: responsável por identificar e agrupar os elementos que compõem um sujeito composto (descrito na seção 5.1.1);
- Extrator de Sintagmas Nominais: responsável por extrair os sintagmas nominais (descrito na seção 5.1.2);
- Extrator de Pronomes: responsável por extrair as anáforas pronominais (descrito na seção 5.1.3);
- Algoritmo de Lappin e Leass: responsável por resolver as anáforas pronominais (descrito na seção 3.2.1) ;

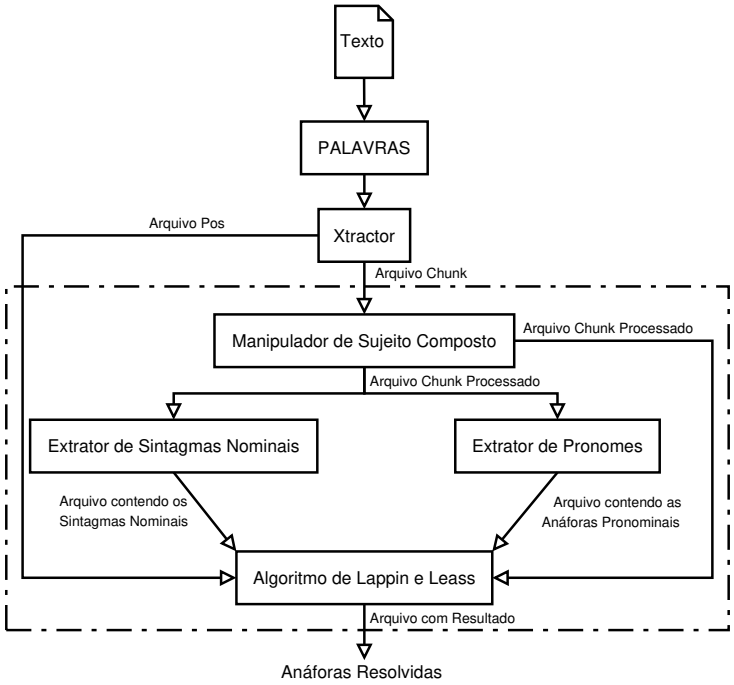


Figura 5.1: Arquitetura do Sistema.

A entrada do sistema consiste dos arquivos *Chunks*, *Words* e *Pos* (apresentados no capítulo 4) gerados pelo Xtractor, a partir dos quais ocorre a identificação e agrupamento de sujeitos compostos pelo Manipulador de Sujeitos Compostos, que será apresentado na seção 5.1.1. Em seguida, são extraídos os possíveis candidatos à co-referência, pelo Extrator de Sintagmas Nominais, apresentado na seção 5.1.2, e as anáforas pronominais, pelo Extrator de Pronomes, apresentado na seção 5.1.3. Finalmente, as anáforas são resolvidas pelo algoritmo de Lappin e Leass, adaptado para o português e apresentado na seção 3.2, gerando um arquivo XML com as anáforas encontradas e seus antecedentes (descrito na seção 5.1.4).

5.1.1 O processamento de sujeitos compostos

Visando resolver as anáforas que se referem a sujeitos compostos, foi necessário fazer um processamento adicional sobre a estrutura sintática das sentenças gerada pelo PALAVRAS, pois não é feita nenhuma marcação indicando quais sintagmas nominais compõem o sujeito composto. Por exemplo, considere o texto (35), abaixo:

(35) Maria e Pedro feriram-se.
Eles morreram após o acidente.

As figuras 5.2 e 5.3(a) ilustram a estrutura sintática da primeira sentença do texto (35), gerada pelo PALAVRAS, e o arquivo *Chunks*, gerado pelo Xtractor, respectivamente. Analisando essas figuras, observa-se que não existe marcação identificando quais sintagmas nominais são membros do sujeito composto "Maria e Pedro", o que dificulta a resolução do pronome "Eles", que faz referência a esse sujeito.

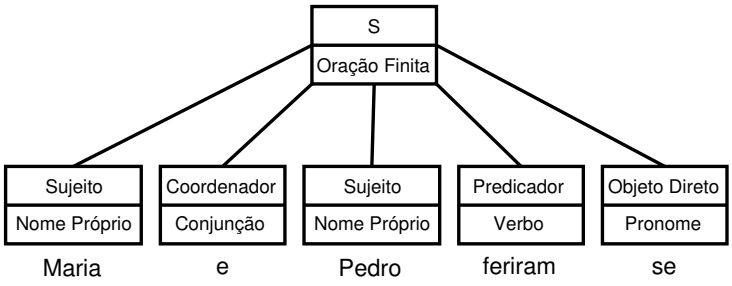


Figura 5.2: Árvore sintática da primeira sentença do texto (35) gerada pelo PALAVRAS.

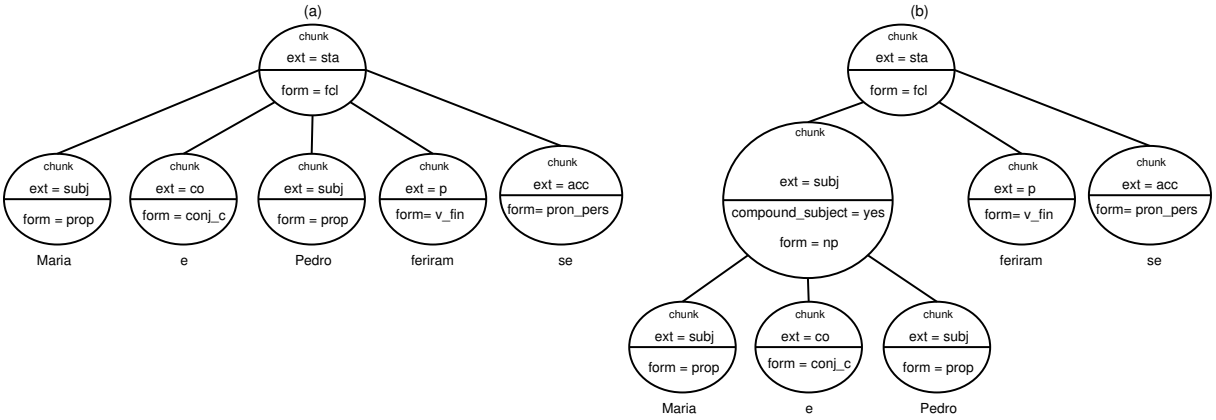


Figura 5.3: Árvore sintática da primeira sentença do texto (35) gerada pelo Xtractor e após a identificação dos sujeitos compostos.

O algoritmo de identificação de sujeitos compostos consiste em agrupar os nós que são irmãos adjacentes e sintagmas nominais, classificados como sujeito, precedidos ou não por uma preposição, a partir do arquivo *Chunks* gerado pelo Xtractor. Os elementos *chunk* considerados sintagmas nominais e sujeito da sentença são aqueles cujo atributo "form" é igual a "np", "prop" ou "n", e cujo atributo "ext" é igual a "subj". Já as conjunções são os elementos *chunk* cujos atributos "form" e "ext" possuem os valores "co" e "conj_c", respectivamente. Após a identificação dos elementos *chunk* membros do sujeito composto, é inserido um novo nó na estrutura da sentença, como pai desses elementos, como mostra a figura 5.3(b). O novo nó inserido na estrutura da árvore possui um atributo adicional,

o atributo "compound_subject" com valor "yes", visando facilitar a etapa de extração dos sintagma nominais, que será apresentada na seção 5.1.2.

A estrutura do arquivo XML, gerado pelo algoritmo de identificação dos sujeitos compostos, é idêntica à estrutura do arquivo *Chunks* do Xtractor, com exceção do atributo adicional "compound_subject". Esse atributo é utilizado para indicar o elemento pai dos nós que foram identificados como membros do sujeito composto. A figura 5.4 ilustra o XML da figura 5.3(b), que é resultado da identificação dos sujeitos compostos da primeira sentença do texto (35). O arquivo *Words* do texto (35), gerado pelo Xtractor e necessário para a interpretação dessas figuras, é apresentado na figura 5.5.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE text SYSTEM "chunk_processado.dtd">
<text>
 <paragraph id="paragraph_1">
  <sentence id="sentence_1" span="word_1..word_6">
   <chunk ext="sta" form="fcl" id="chunk_1" span="word_1..word_5">
     <chunk compound_subject="yes" ext="subj" form="np"
                                id="chunk_15" span="word_1..word_3">
       <chunk ext="subj" form="prop" id="chunk_2" span="word_1"/>
       <chunk ext="co" form="conj_c" id="chunk_3" span="word_2"/>
       <chunk ext="subj" form="prop" id="chunk_4" span="word_3"/>
     </chunk>
     <chunk ext="p" form="v_fin" id="chunk_5" span="word_4"/>
     <chunk ext="acc" form="pron_pers" id="chunk_6" span="word_5"/>
   </chunk>
  </sentence>
  <sentence id="sentence_2" span="word_7..word_12">
   <chunk ext="sta" form="fcl" id="chunk_7" span="word_7..word_11">
    <chunk ext="subj" form="pron_pers" id="chunk_8" span="word_7"/>
    <chunk ext="p" form="v_fin" id="chunk_9" span="word_8"/>
    <chunk ext="advl" form="pp" id="chunk_10" span="word_9..word_11">
      <chunk ext="h" form="prp" id="chunk_11" span="word_9"/>
      <chunk ext="p" form="np" id="chunk_12" span="word_10..word_11">
        <chunk ext="n" form="art" id="chunk_13" span="word_10"/>
        <chunk ext="h" form="n" id="chunk_14" span="word_11"/>
      </chunk>
    </chunk>
   </chunk>
  </sentence>
 </paragraph>
</text>
```

Figura 5.4: Arquivo *Chunks* do texto (35) após processamento dos sujeitos compostos.

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE words SYSTEM "words.dtd">
<words>
  <word id="word_1">Maria</word>
  <word id="word_2">e</word>
  <word id="word_3">Pedro</word>
  <word id="word_4">feriram-</word>
  <word id="word_5">se</word>
  <word id="word_6">.</word>
  <word id="word_7">Eles</word>
  <word id="word_8">morreram</word>
  <word id="word_9">após</word>
  <word id="word_10">o</word>
  <word id="word_11">acidente</word>
  <word id="word_12">.</word>
</words>
```

Figura 5.5: Arquivo *Words* do texto (35).

5.1.2 A extração dos sintagmas nominais

O algoritmo de resolução anafórica requer a identificação dos sintagmas nominais existentes no texto, já que eles serão os possíveis referentes das anáforas.

Portanto, foi necessário desenvolver um algoritmo para extração dos sintagmas nominais. Os passos a seguir descrevem o algoritmo em detalhes:

1. Analisar o arquivo *Chunks* gerado pelo Manipulador de Sujeito Composto:
   (a) Verificar se o elemento *chunk* é sujeito composto, ou seja, possui atributo "compound_subject" igual a "yes". Caso possua, ir ao passo 1b; caso contrário, ir ao passo 1c;
   (b) Processar Sujeito Composto:
      i. Determinar os núcleos dos sintagmas nominais que compõem o sujeito;
      ii. Determinar o gênero do sujeito composto, analisando os elementos do arquivo *Pos* correspondentes aos núcleos do sujeito;
      iii. Processar os sintagmas nominais que compõem o sujeito;
      iv. Incluir o sujeito composto no arquivo XML de saída;
   (c) Processar outros elementos *chunk*:
      i. Caso o elemento seja um sintagma nominal, ou seja, possua o atributo "form" igual a "np", e não possua pai sujeito composto, ir ao passo 1d. Caso contrário, ir ao passo 1(c)ii;
      ii. Caso o elemento seja um nome próprio, isto é, possua atributo "form" igual a "prop", não seja núcleo de nenhum sintagma (possui atributo "ext" diferente de "h"), e não possua pai sujeito composto ou sintagma nominal, ir ao passo 1d. Caso contrário, ir ao passo 1(c)iii;

iii. Caso o elemento seja um substantivo (possui atributo "form" igual a "n"), não seja núcleo de nenhum sintagma (possui atributo "ext" diferente de "h"), e não possua pai sujeito composto ou sintagma nominal, ir ao passo 1d. Caso contrário, descartar o elemento *chunk*;

(d) Processar sintagma nominal, substantivo ou nome próprio:

i. Determinar o núcleo do sintagma nominal;

ii. Determinar o gênero e o número do sintagma nominal, analisando o elemento do arquivo *Pos* correspondente ao núcleo do sintagma;

iii. Incluir o sintagma nominal no XML de saída.

O resultado do algoritmo de extração dos sintagmas nominais é um arquivo XML que contém todos os sintagmas nominais extraídos do texto. As informações sobre os sintagmas nominais, necessárias à resolução de anáforas, são codificadas no elemento *np*, cujo atributo "gender" indica o gênero, o atributo "number" indica o número, o atributo "span" se refere aos identificadores do arquivo *Words* das palavras que compõe o sintagma, o atributo "head_span" se refere aos identificadores do arquivo *Words* das palavras que compõem o núcleo do sintagma, e o atributo "chunk_id" possui o identificador do elemento *chunk* onde o sintagma nominal foi encontrado. Os sintagmas extraídos são agrupados por sentença; assim, todos os elementos *np* filhos de um elemento *sentence* com o mesmo atributo "id" terão sido extraídos da mesma sentença.

A figura 5.6 apresenta o documento resultante da extração dos sintagmas nominais do texto (35) da página 33. O arquivo XML gerado pela extração segue a estrutura definida no DTD (*Document Type Definition*[1] [Ray03]) apresentado na figura 5.7.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE np-set SYSTEM "np-extract.dtd">
<np-set>
  <sentence id="sentence_1">
    <np chunk_id="chunk_15" gender="M" head_span="word_1,word_3"
                              number="P" span="word_1..word_3" />
    <np chunk_id="chunk_2" gender="F" head_span="word_1"
                                      number="S" span="word_1" />
    <np chunk_id="chunk_4" gender="M" head_span="word_3"
                                      number="S" span="word_3" />
  </sentence>
  <sentence id="sentence_2">
    <np chunk_id="chunk_12" gender="M" head_span="word_11"
                            number="S" span="word_10..word_11" />
  </sentence>
</np-set>
```

Figura 5.6: Arquivo resultante da extração dos sintagmas nominais do texto (35).

[1] O arquivo DTD descreve a estrutura do documento XML, ou seja, ele define a lista dos possíveis elementos, seus atributos e valores.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT np-set (sentence*) >

<!ELEMENT sentence (np*) >
<!ATTLIST sentence id ID #REQUIRED >

<!ELEMENT np EMPTY >
<!ATTLIST np chunk_id ID #REQUIRED >
<!ATTLIST np gender (M|F|M-F|unknown) #REQUIRED >
<!ATTLIST np head_span CDATA #REQUIRED >
<!ATTLIST np span CDATA #REQUIRED >
<!ATTLIST np number CDATA #REQUIRED >
```

Figura 5.7: DTD do arquivo gerado pelo processo de extração dos sintagmas nominais.

É importante ressaltar que, caso o algoritmo de extração dos sintagmas nominais não consiga determinar o núcleo do sintagma, o gênero, o número e o núcleo do sintagma nominal encontrado serão anotados como "unknown", ou seja, indeterminados. Portanto, este sintagma não será considerado na geração da lista de candidatos à co-referência. Isso ocorrerá caso seja encontrada alguma anomalia nos arquivos gerados pelas ferramentas utilizadas (anomalias descritas na seção 6.2), ou caso o gênero do sintagma nominal seja inválido (diferente de "M", "F", "M-F" ou "F-M").

5.1.3 A extração dos pronomes

O sistema desenvolvido resolve anáforas pronominais e, portanto, foi necessário desenvolver um algoritmo para a extração dos pronomes. O algoritmo de extração dos pronomes seque os seguintes passos:

1. Analisar o Arquivo *Chunks* gerado pelo Manipulador de Sujeito Composto:
   (a) Caso o elemento *chunk* seja um pronome pessoal, ou seja, possua o atributo "form" igual a "pron_pers":
      i. Localizar no arquivo *Pos* o elemento *word* correspondente ao pronome;
      ii. Analisar o elemento *word* encontrado:
         A. Descartar o pronome caso ele não tenha sido reconhecido corretamente pelo analisador sintático, ou tenha sido extraído incorretamente pelo Xtractor, ou seja, caso o elemento possua o texto "unknown";
         B. Descartar o pronome caso possua número inválido (isto é, o filho com atributo "number" não possua texto contendo "P", "S", "S-P" ou "P-S"), ou gênero inválido (o filho com atributo "gender" for diferente de "M", "F" , "M-F" ou 'F-M");

C. Verificar se o pronome é reflexivo ou recíproco: caso o elemento possua um filho com atributo "tag" igual a "refl", "si" ou "reci", incluir o pronome no arquivo XML de saída; caso contrário, ir ao passo 1(a)iiD;

D. Verificar se o pronome está na terceira pessoa: caso o elemento possua um filho com atributo "number" que contenha o texto "3", incluir o pronome no XML de saída; caso contrário, descartar o pronome.

O resultado da extração dos pronomes é um arquivo XML contendo todos os pronomes que o algoritmo tentará resolver. Os pronomes são agrupados em elementos *sentence*, cujo atributo "id" indica sua sentença de origem. Cada elemento *pron* representa um pronome, e possui as informações utilizadas na sua resolução em seus atributos. São eles: "gender", "number", "reci", "refl", "span" e "chunk_id". O primeiro atributo, ou seja "gender", indica o gênero do pronome; o segundo, isto é, "number", se refere ao número em que o pronome ocorre no texto. Já os atributos "reci" e "refl" contém a classificação do pronome, ou seja, se ele é reflexivo ou recíproco, respectivamente. O atributo "span" possui o identificador do arquivo *Words* das palavras que constituem o pronome. Finalmente, o atributo "chunk_id" possui o identificador do elemento *chunk* onde o pronome foi encontrado.

A figura 5.8 apresenta o documento resultante da extração dos pronomes do texto (35) da página 33. O arquivo XML gerado pela extração segue a estrutura definida no DTD apresentado na figura 5.9.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE pronoun-set SYSTEM "pronoun.dtd">
<pronoun-set>
  <sentence id="sentence_1">
    <pron chunk_id="chunk_6" gender="M" number="3S-P"
                          reci="no" refl="yes" span="word_5" />
  </sentence>
  <sentence id="sentence_2">
    <pron chunk_id="chunk_8" gender="M" number="3P"
                           reci="no" refl="no" span="word_7" />
  </sentence>
</pronoun-set>
```

Figura 5.8: Arquivo resultante da extração dos pronomes do texto (35).

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!ELEMENT pronoun-set ( sentence* ) >

<!ELEMENT sentence ( pron* ) >
<!ATTLIST sentence id ID #REQUIRED >

<!ELEMENT pron EMPTY >
<!ATTLIST pron chunk_id ID #REQUIRED >
<!ATTLIST pron gender CDATA #REQUIRED >
<!ATTLIST pron number CDATA #REQUIRED >
<!ATTLIST pron reci ( no | yes | unknown ) #REQUIRED >
<!ATTLIST pron refl ( no | yes | unknown ) #REQUIRED >
<!ATTLIST pron span NMTOKEN #REQUIRED >
```

Figura 5.9: DTD do arquivo gerado pelo processo de extração dos pronomes.

5.1.4 Os resultados do algoritmo

O resultado da última etapa do processamento, ou seja, da resolução de anáforas, é um arquivo XML com todos os pronomes que o algoritmo tentou resolver acompanhado de seus referentes. Cada elemento "result" representa um par formado pelo pronome e pelo seu antecedente, escolhido pelo algoritmo, mapeados nos atributos "pronoun" e "antecedent", respectivamente. Esses dois atributos possuem identificadores do arquivo de *Words* gerado pelo Xtractor. O atributo "pronoun" contém o identificador do pronome e o atributo "antecedent", o identificador de seu referente.

A figura 5.10 apresenta o documento resultante da resolução dos pronomes do texto (35) da página 33. O arquivo XML com os resultados segue a estrutura definida no DTD apresentado na figura 5.11.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<results>
  <result antecedent="word_1..word_3" pronoun="word_5"/>
  <result antecedent="word_1..word_3" pronoun="word_7"/>
</results>
```

Figura 5.10: Arquivo resultante da resolução dos pronomes do texto (35).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT results ( result* ) >
<!ELEMENT result EMPTY >
<!ATTLIST result antecedent CDATA #REQUIRED >
<!ATTLIST result pronoun NMTOKEN #REQUIRED >
```

Figura 5.11: DTD do arquivo gerado pelo processo resolução dos pronomes.

5.2 Modelagem de classes

Um modelo de classes visa capturar, principalmente, aspectos de informação de um sistema. A identificação das classes de um modelo é fundamental para o desenvolvimento de um sistema, pois representa o ponto de partida para se obter uma descrição cada vez mais detalhada do contexto, em termos de associações e operações.

A Linguagem de Modelagem Unificada (*Unified Modeling Language*-UML) [RJB99] foi a notação utilizada neste trabalho. Ela é um padrão de linguagem para criar modelos que servem de base para o desenvolvimento de sistemas orientados a objetos.

O algoritmo desenvolvido é composto pelos pacotes *AnaphorResolution* e *XmlUtils*. O diagrama de classes do primeiro pacote é ilustrado na figura 5.12. Esse pacote engloba as classes responsáveis por implementar o núcleo do algoritmo. São elas:

- *LappinLeassResolver*: implementa o algoritmo de resolução das anáforas, descrito na seção 3.2.1;
- *FiltroMorfologico*: implementa o filtro morfológico, responsável por verificar a concordância de gênero e número entre a anáfora e seu candidato a co-referente;
- *SalienceFactorHandle*: responsável por calcular os fatores de saliência, apresentados na seção 3.1.1;
- *Constraints*: essa interface representa as restrições que o algoritmo utilizará para selecionar os candidatos intra-sentenciais;
- *ReinhartConstraints*: implementa as restrições de Reinhart, descritas na seção 2.2.1;
- *EquivalenceClass*: implementa o conceito de classes de equivalência, descrito na seção 3.1.2;
- *EquivalenceClassList*: responsável por gerenciar todas as classes de equivalência criadas no processo de resolução;
- *DataHolder*: responsável por manter os dados da palavra necessários ao processo de resolução, como a sentença de origem, o elemento do arquivo *Chunks* onde a palavra ocorre, e informações como gênero e número, contidas no elemento do arquivo de extração (resultado do processo de extração de pronomes e sintagmas nominais);
- *NounPhraseHolder*: abstração que representa um sintagma nominal;
- *PronounHolder*: abstração que representa uma anáfora pronominal;

- *ExtratorSintagmasNominais*: implementa o algoritmo de extração dos sintagmas nominais, apresentado na seção 5.1.2;
- *ExtratorPronomes*: implementa o algoritmo de extração das anáforas pronominais, apresentado na seção 5.1.3;
- *ManipuladorSujeitoComposto*: implementa o algoritmo de identificação e agrupamento dos membros de sujeitos compostos, descrito na seção 5.1.1;
- *SentenceIterator*: responsável por gerar a lista de candidatos inter-sentenciais e intra-sentenciais. Essa classe também implementa funcionalidades para facilitar a navegação entre os sintagmas nominais e os pronomes extraídos por sentença;

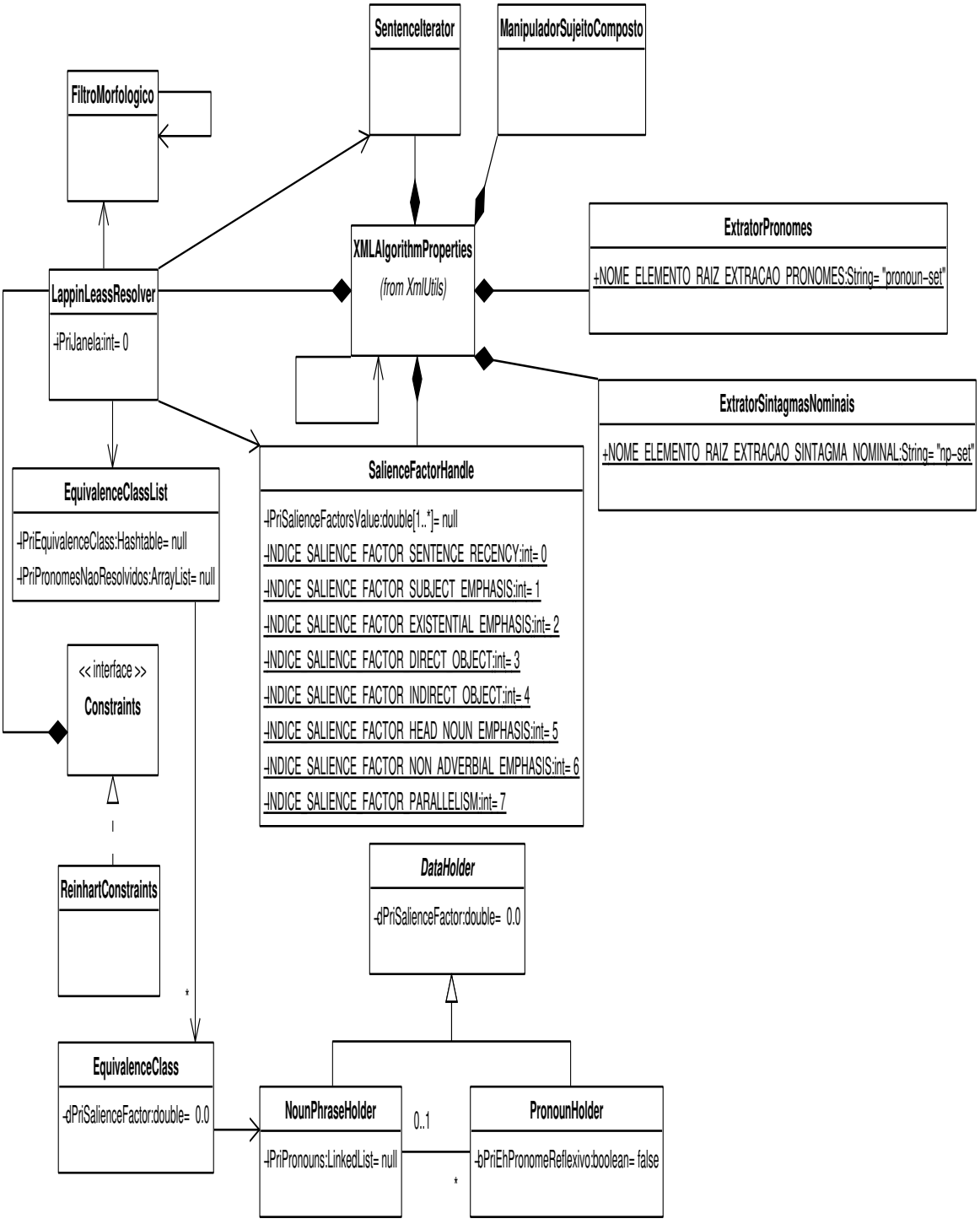


Figura 5.12: Diagrama de Classes do Pacote AnaphorResolution.

O diagrama de classes do pacote *XmlUtils* é ilustrado na figura 5.13. Esse pacote engloba classes utilitárias para o processamento de Arquivos XML. São elas:

- *XmlProperties*: implementa funcionalidades para ler configurações a partir de um arquivo XML;
- *XmlAlgorithmProperties*: responsável por gerenciar as configuração do algoritmo, como valor dos fatores de saliência, localização dos arquivos gerados pelo Xtractor, entre outros;
- *XmlHandle*: implementa funções necessárias para validar e manipular documentos XML;
- *Constantes*: mantém todas as constantes utilizadas pelo algoritmo;
- *FiltroNomeElemento*: utilizada para filtrar os elementos no documento XML.

O padrão Singleton [GHJV94] foi empregado nas classes *XMLAlgorithmProperties*, *XmlHandle* e *FiltroMorfologico*, visando otimizar a utilização de memória pelo algoritmo.

Figura 5.13: Diagrama de Classes do Pacote XmlUtils.

5.3 Considerações

Neste capítulo foram apresentadas a arquitetura e a modelagem do sistema desenvolvido, que implementa o algoritmo proposto na seção 3.2.1. Também foram detalhadas as etapas de pré-processamento implementadas, como o tratamento dos sujeitos compostos, extração dos sintagmas nominais e a extração dos pronomes.

No próximo capítulo, serão apresentados os corpora utilizados na avaliação do algoritmo desenvolvido e os resultados obtidos.

Capítulo 6

Avaliação do Algoritmo

O algoritmo desenvolvido foi avaliado em três corpora: um corpus jurídico, um literário e um jornalístico. Como o desempenho do algoritmo no corpus jurídico ficou abaixo das expectativas, foram criados dois novos corpora – o corpus literário e o jornalístico – para melhor avaliá-lo. Neste capítulo, são apresentadas as características dos corpora utilizados na avaliação e os resultados obtidos.

6.1 Descrição do Corpora

Os corpora foram anotados automaticamente pelo PALAVRAS, com informações morfossintáticas, e a anotação dos pronomes pessoais foi feita manualmente, utilizando a ferramenta de anotação de discurso MMAX (*Multi-Modal Annotation in XML*) [MS01], descrita na seção 4.2.

A anotação das anáforas pronominais no corpus jurídico não englobou todos os pronomes de terceira pessoa, e não incluiu os pronomes reflexivos e recíprocos. Já a anotação nos demais corpora, que foi realizada como parte deste trabalho, abrangeu todas as anáforas tratadas pelo algoritmo e reconhecidas pelo PALAVRAS; além disso, as expressões foram classificadas como inter-sentencial, intra-sentencial ou não anafóricas.

O corpus jurídico[1] é composto por Pareceres da Procuradoria Geral da República de Portugal, e é composto de sentenças longas e complexas, como a sentença abaixo:

(36) O casamento não irá afectar as legítimas expectativas dos filhos já existentes, visto que quer eles, quer os eventuais e futuros irmãos germanos, serão herdeiros de ambos os progenitores, quaisquer que sejam os bens, próprios ou comuns, destes, qualquer que seja o regime de bens convencionado.

[1] Cedido pela Profa. Renata Vieira, do Centro de Ciências Exatas e Tecnológicas, da Universidade do Vale do Rio dos Sinos (Unisinos)

Já o corpus literário consiste do livro *O Alienista*, de Machado de Assis[2], também de natureza complexa, como mostra o seguinte fragmento:

(37) Ela foi uma verdadeira rainha naqueles dias memoráveis; ninguém deixou de ir visitá-la duas e três vezes, apesar dos costumes caseiros e recatados do século, e não só a cortejavam como a louvavam; porquanto,– e este fato é um documento altamente honroso para a sociedade do tempo, – porquanto viam nela a feliz esposa de um alto espírito, de um varão ilustre, e, se lhe tinham inveja, era a santa e nobre inveja dos admiradores.

A distribuição das anáforas no corpus literário com relação ao tipo dos pronomes e ao tipo das expressões anafóricas é apresentada nas tabelas 6.1 e 6.2, respectivamente.

Tabela 6.1: Distribuição das anáforas no corpus literário por tipo de pronome.

Tipo do Pronome	**Total de pronomes anotados**
Pronomes de terceira pessoa	595 (85,49%)
Pronomes reflexivos/recíprocos anotados	101 (14,51%)
Total	696

Tabela 6.2: Classificação das expressões anafóricas no corpus literário.

Tipo da Anáfora	**Total anáforas anotadas**
Intra-sentenciais	219 (31,46%)
Inter-sentenciais	372 (53,45%)
Pronomes não anafóricos[3]	105 (15,09%)
Total	696

O corpus jornalístico é constituído por 14 textos jornalísticos e é composto de sentenças mais simples. A distribuição das anáforas nesse corpus por tipo de pronome é apresentada na tabela 6.3, e a classificação das expressões anáforas anotadas é ilustrada na tabela 6.4.

Tabela 6.3: Distribuição das anáforas no corpus jornalístico por tipo de pronome.

Tipo do Pronome	**Total de pronomes anotados**
Pronomes de terceira pessoa	162 (72,00%)
Pronomes reflexivos/recíprocos anotados	63 (28,00%)
Total	225

[2]Disponível em http://bibvirt.futuro.usp.br/textos/autores/machadodeassis/alienista/alienista.html
[3]Pronomes que não possuem antecedente.

Tabela 6.4: Classificação das expressões anafóricas no corpus jornalístico.

Tipo da Anáfora	**Total anáforas anotadas**
Intra-sentenciais	113 (50,22%)
Inter-sentenciais	70 (31,11%)
Pronomes não anafóricos	42 (18,67%)
Total	225

6.2 Resultados

A avaliação do algoritmo desenvolvido foi realizada em três experimentos. No primeiro experimento, que foi realizado com o corpus jurídico, o processo de resolução foi considerado correto caso a solução gerada pelo algoritmo tenha sido idêntica a solução anotada manualmente, ou caso ela fosse um sintagma nominal contido no sintagma dado pela anotação manual. Assim, as soluções geradas pelo sistema foram automaticamente comparadas com as soluções anotadas manualmente. A tabela 6.5 apresenta os resultados globais obtidos com o corpus jurídico.

Tabela 6.5: Análise global do corpus jurídico.

Total anáforas anotadas[4]	297
Anáforas mal resolvidas[5]	190
Anáforas resolvidas corretamente	103
Anáforas mal identificadas[6]	4
Percentagem de sucesso	35.15%

No segundo experimento, onde foi utilizado o corpus literário, o processo de resolução foi considerado correto se a solução gerada pelo algoritmo é idêntica a solução anotada manualmente, ou caso ela seja um sintagma nominal que co-refere com a solução dada pelos anotadores. Portanto, todas as soluções geradas pelo algoritmo foram comparadas manualmente com as soluções dadas pelos anotadores.

A avaliação com o corpus literário é apresentada nas tabelas 6.6, 6.7 e 6.8. A primeira tabela apresenta os resultados globais, a segunda ilustra os resultados em relação aos tipos dos pronomes, e a última tabela apresenta os resultados em relação ao tipo das expressões anafóricas.

O último experimento foi realizado com o corpus jornalístico e o literário. Devido a um problema encontrado com a anotação de informações morfossintáticas geradas pelas

[4]Número total de anáforas pronominais de terceira pessoa anotadas manualmente; os pronomes recíprocos/reflexivos não foram anotados.

[5]Anáforas onde o algoritmo escolheu o antecedente incorreto.

[6]Número de pronomes anotados manualmente que não foram identificados pelo algoritmo.

Tabela 6.6: Resultados globais do corpus literário.

Anáforas resolvidas corretamente	218 (31,32%)
Anáforas mal resolvidas	478 (68,68%)
Total anáforas anotadas	696

Tabela 6.7: Resultados do corpus literário por tipo do pronome.

Tipos do pronome	**Resolvidos corretamente**	**Mal resolvidos**	**Total**
Pronomes em terceira pessoa	161 (27,01%)	435 (72,99%)	596
Pronomes reflexivos/recíprocos	57 (57,00%)	43 (43,00%)	100
Total	218	478	696

ferramentas empregadas, descritos posteriormente nesta seção, o gênero dos personagens principais do corpus literário foram anotados manualmente. Nesse experimento foram adotados os mesmos procedimentos do anterior na análise dos resultados.

As tabelas 6.9, 6.10 e 6.11 apresentam os resultados globais do corpus literário, os resultados com relação ao tipo de pronomes e ao tipo das expressões anafóricas, respectivamente. Apesar da correção manual do gênero realizada no corpus literário, houve uma melhora de apenas 1.29% com relação a avaliação do experimento anterior.

Os resultados obtidos com o corpus jornalístico são apresentados nas tabelas 6.12, 6.13 e 6.14. A primeira tabela apresenta os resultados globais, na segunda os resultados são apresentados com relação ao tipo dos pronomes e, na terceira, com relação ao tipo da expressão anafórica.

Tabela 6.8: Resultados do corpus literário por tipo de expressão anafórica.

Tipos das Anáforas	**Resolvidas corretamente**	**Mal resolvidas**	**Total**
Intra-sentenciais	80 (36,53%)	139 (63,47%)	219
Inter-sentenciais	102 (27,42%)	270 (72,58%)	372
Pronomes não anafóricos	36 (34,29%)	69 (65,71%)	105
Total	218	478	696

Tabela 6.9: Resultados globais do corpus literário.

Anáforas resolvidas corretamente	227 (32,61%)
Anáforas mal resolvidas	469 (67,39%)
Total anáforas anotadas	696

Tabela 6.10: Resultados do corpus literário por tipo pronome.

Tipos do pronome	**Resolvidos corretamente**	**Mal resolvidos**	**Total**
Pronomes em terceira pessoa	169 (28,40%)	426 (71,60%)	595
Pronomes reflexivos/recíprocos	58 (57,43%)	43 (42,57%)	101
Total	227	469	696

Tabela 6.11: Resultados do corpus literário por tipo de expressão anafórica.

Tipos das Anáforas	**Resolvidas corretamente**	**Mal resolvidas**	**Total**
Intra-sentenciais	82 (37,44%)	137 (62,56%)	219
Inter-sentenciais	109 (29,30%)	263 (70,70%)	372
Pronomes não anafóricos	36 (34,29%)	69 (65,71%)	105
Total	227	469	696

Tabela 6.12: Resultados globais do corpus jornalístico.

Anáforas resolvidas corretamente	98 (43,56%)
Anáforas mal resolvidas	127 (56,44%)
Total anáforas anotadas	225

Tabela 6.13: Resultados do corpus jornalístico por tipo pronome.

Tipos do pronome	**Resolvidos corretamente**	**Mal resolvidos**	**Total**
Pronomes em terceira pessoa	64 (39,51%)	98 (60,49%)	162
Pronomes reflexivos/recíprocos	34 (53,97%)	29 (46,03%)	63
Total	98	127	225

Tabela 6.14: Resultados do corpus jornalístico por tipo de expressão anafórica.

Tipos das Anáforas	**Resolvidas corretamente**	**Mal resolvidas**	**Total**
Intra-sentenciais	53 (46,90%)	60 (53,10%)	113
Inter-sentenciais	32 (45,71%)	38 (54,29%)	70
Pronomes não anafóricos	13 (30,95%)	29 (69,05%)	42
Total	98	127	225

A resolução de pronomes reflexivos e recíprocos apresentou melhores resultados devido ao seu processo de resolução – dado que somente candidatos intra-sentenciais são coletados, o número de possíveis candidatos para esse tipo de pronome é menor.

Embora o algoritmo tenha apresentado uma melhora de desempenho com o corpus jornalístico, onde a taxa de acerto global foi de 43,56%, os resultados obtidos foram bastante inferiores ao percentual de acerto de 86% obtido por Lappin e Leass, que utilizaram um corpus de manuais de computadores na língua inglesa. É importante ressaltar que a avaliação do algoritmo original foi realizada num corpus de natureza mais simples que os corpora utilizados nesta avaliação.

Suspeita-se que a baixa taxa de acerto nos corpora jurídico e literário seja devida, em parte, à natureza complexa desses corpora. Além disso, 46,84% das anáforas pronominais do corpus literário consistem do pronome *lhe(s)* e *se*, cujos candidatos a antecedente podem ser do gênero masculino ou feminino. Isso dificulta a resolução desses pronomes, já que o filtro morfológico não elimina nenhum candidato a co-referência.

Uma análise manual da execução do algoritmo desenvolvido apontou alguns problemas na anotação do PALAVRAS, como informações morfossintáticas incorretas, identificação incorreta de pronomes reflexivos e recíprocos, e sentenças com nós duplicados na sua estrutura sintática. Considere as seguintes sentenças:

(38) Crispim Soares, ao tornar a casa, trazia os olhos entre as duas orelhas da besta ruana em que vinha montado.

(39) Dona Evarista, contentíssima com a glória do marido, vestiu-se luxuosamente, cobriu-se de jóias, flores e sedas.

(40) Era a revolta que tornava a si da ligeira síncope e ameaçava arrasar a Casa Verde.

As figuras 6.1, 6.2 e 6.3 mostram parte da árvore sintática gerada pelo PALAVRAS para as sentenças (38), (39) e (40), respectivamente. A primeira figura ilustra a anotação incorreta de informações morfossintáticas. O sintagma nominal "Crispim Soares", em destaque, é marcado indevidamente pelo PALAVRAS com gênero indeterminado. Isso dificulta o processo de resolução dos pronomes, já que o filtro morfológico poderá selecionar candidatos a antecedente inadequados.

```
STA:fcl
=SUBJ:prop('Crispim_Soares' M/F S) Crispim_Soares
=,
=ADVL:prp('a' <sam->) a
=>N:art('o' M S <artd> <-sam>) o
=P<:icl
==P:v-inf('tornar' 0/1/3S) tornar
==ADVS:pp
===H:prp('a') a
===P<:n('casa' F S <build>) casa
...
```

Figura 6.1: Parte da árvore sintática da sentença (38).

A figura 6.2 mostra em destaque dois pronomes reflexivos classificados indevidamente pelo PALAVRAS, os dois pronomes "se". Na classificação de ambos os pronomes o analisador sintático não conseguiu determinar que eles são reflexivos. Uma vez que o processo de resolução de pronomes reflexivos e recíprocos, e pronomes em terceira pessoa são distintos, a classificação incorreta do pronome força o algoritmo a escolher o processo de resolução inadequado e, possivelmente, a escolha incorreta de seu antecedente.

```
STA:fcl
=SUBJ:prop('D._Evarista' M/F S) D._Evarista
=,
=N<PRED:adj('contentíssimo' F S <DERS>) contentíssima
=ADVL:pp
==H:prp('com') com
==P<:np
===>N:art('o' F S <artd>) a
===H:n('glória' F S <am>) glória
===N<:pp
====H:prp('de' <sam->) de
====P<:np
=====>N:art('o' M S <artd> <-sam>) o
=====H:n('marido' M S <Hfam>) marido
=,
=P:v-fin('vestir' PS 3S IND <hyfen>) vestiu-
=ACC:pron-pers('se' M/F 3S ACC <obj>) se
=ADVL:adv('luxuosamente') luxuosamente
=,
=P:v-fin('cobrir' PS 3S IND <hyfen>) cobriu-
=ACC:pron-pers('se' M/F 3S ACC <obj>) se
...
```

Figura 6.2: Parte da árvore sintática da sentença (39).

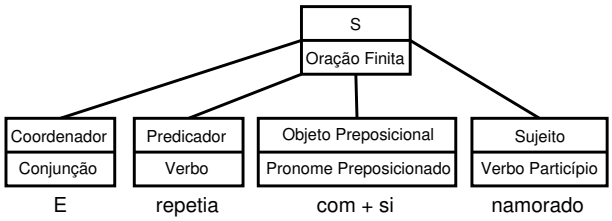
Na figura 6.3 estão em destaque os nós duplicados na estrutura sintática da árvore gerada pelo PALAVRAS. A árvore sintática gerada incorretamente influencia na escolha do candidato à co-referência intra-sentencial, já que o algoritmo analisa essa estrutura para determinar se a co-referência é permitida.

```
STA:cu
=CJT:fcl
==P:v-fin('ser' IMPF 1/3S IND) Era
==SC:np
===>N:art('o' F S <artd>) a
===H:n('revolta' F S <occ>) revolta
==ACC:fcl
===ACC:fcl
====ACC:fcl
=====ACC:fcl
======SUB:conj-s('que') que
======P:v-fin('tornar' IMPF 1/3S IND) tornava
======PIV:pp
=======H:prp('a') a
=======P<:pron-pers('se' M/F 3S/P PIV <refl>) si
======ADVL:pp
=======H:prp('de' <sam->) de
=======P<:np
========>N:art('o' F S <artd> <-sam>) a
========>N:adj('ligeiro' F S) ligeira
========H:n('síncope' F S <ac-cat>) síncope
...
```

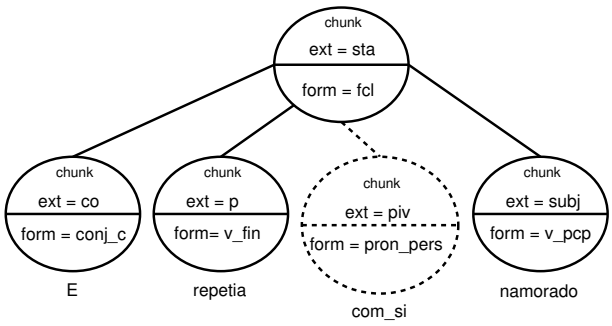
Figura 6.3: Parte da árvore sintática da sentença (40).

Além dos problemas descritos anteriormente, também foram identificadas algumas anomalias no XML gerado pelo Xtractor. Uma delas é a ausência de nós na estrutura sintática de algumas sentenças, como na primeira sentença do texto (41), abaixo, onde o pronome "consigo" não é extraído. A figura 6.4(a) mostra a árvore sintática gerada pelo PALAVRAS e a figura 6.4(b) ilustra o resultado da extração dessa árvore pelo Xtractor, sendo que o nó ausente na extração está em destaque. Como mencionado anteriormente, essa anomalia influencia na escolha dos candidatos a co-referência intra-sentenciais.

(41) E repetia consigo namorado: - Bastilha da razão humana!



(a) Árvore gerada pelo PALAVRAS.



(b) Árvore gerada pelo Xtractor (Nó ausente na extração está tracejado).

Figura 6.4: Árvore sintática da primeira sentença do texto (41).

Outra anomalia identificada pela análise manual é a extração incorreta de informações morfossintáticas pelo Xtractor, como ocorre com o sintagma nominal "desafeiçoado" da sentença (42), abaixo. As figuras 6.5(a) e 6.5(b) ilustram parte da árvore sintática gerada pelo PALAVRAS e o elemento do arquivo *Pos* gerado pelo Xtractor correspondente ao sintagma nominal "desafeiçoado", respectivamente. A extração incorreta dessas informações influencia na seleção dos candidatos a antecedente pelo filtro morfológico, como descrito anteriormente.

(42) Um dia, como um desses incuráveis devedores lhe atirasse uma chalaça grossa, e ele se risse dela, observou um desafeiçoado, com certa perfídia:

```
...
=ACC:np
==>N:art('um' M S <arti>) um
==H:v-pcp('desafeiçoar' <n> M S <DERP>) desafeiçoado
...
```

(a) Parte da árvore sintática gerada pelo PALAVRAS.

```
...
<word id="word_23">
  unknown(v(pcp,desafeiçoar,'<n>','M','S','<DERP>'))
</word>
...
```

(b) Parte do arquivo Pos gerado pelo Xtractor. A informação extraída incorretamente está destacada.

Figura 6.5: Extração de informação morfossintática incorreta da sentença (42).

Outro fator que pode ter influenciado no desempenho são os pesos dos fatores de saliência aplicados, que foram os mesmos utilizados no algoritmo original. Os pesos foram otimizados para o corpus em inglês e, portanto as estruturas sintáticas contempladas por eles também podem ter contribuído para os baixos resultados.

As conclusões e sugestões para trabalhos futuros serão apresentadas no próximo capítulo.

Capítulo 7

Conclusão

Neste trabalho foi implementada e avaliada uma versão modificada do algoritmo de Lappin e Leass para a língua portuguesa. Esse algoritmo é capaz de resolver anáforas inter-sentenciais e intra-sentenciais pronominais de terceira pessoa, e pronomes reflexivos e recíprocos. O algoritmo foi avaliado em três corpora distintos: jurídico, literário e jornalístico.

O corpus jurídico havia sido previamente anotado; entretanto, a anotação não englobava os pronomes reflexivos e recíprocos. Já os corpora literário e jornalístico, cuja anotação foi feita como parte deste trabalho, também englobou esses dois tipos de pronomes. Além disso, na anotação desses dois últimos corpora, as expressões anafóricas foram classificadas em relação a sua localização na sentença, isto é, em intra-sentenciais e inter-sentenciais.

A avaliação do algoritmo nos três corpora mostrou um desempenho inferior ao do algoritmo original para o inglês, que obteve uma taxa de sucesso de 86%. A anotação adicional dos pronomes reflexivos e recíprocos nos corpora literário e jornalístico não ocasionou nenhuma melhora significativa nos resultados. O algoritmo implementado apresentou melhores resultados com o corpus jornalístico, com o qual obteve uma taxa de sucesso de 43,56%.

É importante ressaltar que a avaliação do algoritmo original foi feita em um corpus composto por manuais de computadores em inglês, ou seja, o corpus utilizado na avaliação era mais simples. Isso provavelmente contribuiu para a diferença entre os resultados obtidos por Lappin e Leass e os resultados apresentados neste trabalho. Além disso, os pesos dos fatores de saliência, empiricamente obtidos por Lappin e Leass para o inglês, e as estruturas sintáticas comtempladas pelos fatores, podem não ser adequados para o português. As ferramentas utilizadas neste trabalho também introduziram alguns erros no processo de resolução de anáforas, que não foram contabilizados.

7.1 Trabalhos futuros

Como aperfeiçoamentos deste trabalho, pretende-se fazer uma análise do impacto dos erros introduzidos pelas ferramentas utilizadas no desempenho do algoritmo desenvolvido. Assim, será possível averiguar se haveria necessidade de revisão do sistema de pesos, com uma possível adequação para as características da língua portuguesa, visando otimizar o desempenho do algoritmo.

Uma outra possibilidade seria fazer uma comparação com o algoritmo de Centering, também adaptado para o português, e avaliado em [ACC+04], com o algoritmo de Hobbs [Hob78], empregando o mesmo corpora e as ferramentas utilizadas neste trabalho. Isso possibilitaria indagar qual dessas abordagens computacionais é a mais adequada para o português, e possivelmente apontar as características que mais auxiliariam na resolução de anáforas em português.

Finalmente, também será avaliada a possibilidade de desenvolver um algoritmo híbrido, que utilize o sistema de pesos usados no RAP e o emprego de restrições sintáticas mais específicas para o português, juntamente com alguma abordagem orientada a discurso, como por exemplo a Teoria de *Centering*.

Referências Bibliográficas

[ACC+04] Ana Margarida Aires, Jorge Cesar Barbosa Coelho, Sandra Collovini, Paulo Quaresma, and Renata Vieira. Avaliação de centering em resolução pronominal da língua portuguesa. In *Taller de Herramientas y Recursos Lingüísticos para el Español y el Portugués*, volume 1, pages 1–8, August 2004.

[All95] James Allen. *Natural language understanding.* The Benjamin/Cummings Publishing Company, 1995.

[BFP87] Susan E. Brennan, Marilyn W. Friedman, and Carl J. Pollard. A centering approach to pronouns. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, pages 155–162, Morristown, NJ, USA, 1987.

[Bic00] Eckhard Bick. *The parsing system PALAVRAS: Automatic grammatical analysis of Portuguese in a constraint grammar frammework.* PhD thesis, Århus University, 2000.

[BMR+96] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-oriented software archtecture: A system of patterns.* Wiley, 1996.

[Car89] Ariadne Maria Brito Rizzoni Carvalho. *Logic grammars and pronominal anaphora.* PhD thesis, University of Reading, March 1989.

[CC05a] Thiago Thomes Coelho and Ariadne Maria Brito Rizzoni Carvalho. Lappin and Leass' algorithm for pronoun resolution in Portuguese. In Carlos Bento, Amílcar Cardoso, and Gaël Dias, editors, *the 12th Portuguese Conference on Artificial Intelligence (EPIA 2005), LNAI 3808*, volume 3808 / 2005 of *Lecture Notes in Computer Science*, pages 680–692, Covilhã, Portugal, December 2005. Springer-Verlag.

[CC05b] Thiago Thomes Coelho and Ariadne Maria Brito Rizzoni Carvalho. Uma adaptação do algoritmo de Lappin e Leass para resolução de anáforas em português. In *Anais do XXV Congresso da Sociedade Brasileira de Computacao (III Workshop em Tecnologia da Informação e da Linguagem Humana)*, July 2005.

[CJ81] Joaquim Mattoso Camara Junior. *Dicionário de lingüística e gramática referente à lingua Portuguesa.* Editora Vozes, 9ª edition, 1981.

[DGG+04] Jean Dubois, Mathée Giacomo, Louis Guespin, Christiane Marcellesi, Jean-Baptiste Marcellesi, and Jean-Pierre Mevel. *Dicionário de lingüística.* Editora Cultrix, 9ª edition, 2004.

[FPM98] Antonio Ferrández, Manuel Palomar, and Lidia Moreno. Anaphora resolution in unrestricted texts with partial parsing. In *the 36th Annual Meeting of the Association for Computational Linguistics*, pages 385–391, Morristown, NJ, USA, 1998.

[Fáv91] Leonor Lopes Fávero. *Coesão e coerência textuais.* Série Princípios. Editora Ática, 1991.

[GHJV94] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Desing patterns: Elements of reusable object-oriented software.* Addison-Wesley Professional, 1ª edition, 1994.

[Gri92] Ralph Grishman. *Computational linguistics: an introduction.* Cambridge University Press, 1992.

[GVGQ03] Caroline Varaschin Gasperin, Renata Vieira, Rodrigo Rafael Villarreal Goulart, and Paulo Quaresma. Extracting xml chunks from Portuguese corpora. In *Proceedings of the Workshop on Traitement automatique des langues minoritaires*, 2003.

[GWJ95] Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. Centering: A framework for modeling the local coherence of discourse. *Association for Computational Linguistics*, 21(2):203–225, 1995.

[HC02a] Cay Horstmann and Gary Cornell. *Core Java 2: Advanced Features*, volume 2. Prentice Hall, 2002.

[HC02b] Cay Horstmann and Gary Cornell. *Core Java 2: fundamentals*, volume 1. Prentice Hall, 2002.

[HH76] Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. *Cohesion in English*. Number 9 in English Language Series. Longman Pub Group, 1976.

[Hir81] Graeme Hirst. *Anaphora in natural language understanding: a survey*, volume 119 of *Lecture notes in computer science*. Springer, 1981.

[Hob78] Jerry Hobbs. Resolving pronoun references. *Lingua*, 44:311–338, 1978.

[JHM00] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, 1ª edition, 2000.

[Koc89] Ingedore Grunfeld Villaça Koch. *A coesão textual*. Repensando a língua portuguesa. Contexto, 6ª edition, 1989.

[LL94] Shalom Lappin and Herbert J. Leass. An algorithm for pronomial anaphora resolution. *Computational Linguistics*, 20(4):535–561, December 1994.

[LM90a] Shalom Lappin and Michael McCord. Anaphora resolution in slot grammar. *Computational Linguistics*, 16(4):197–212, 1990.

[LM90b] Shalom Lappin and Michael McCord. A syntactic filter on pronominal anaphora for slot grammar. In *28th Annual Meeting of the Association for Computational Linguistics*, pages 135–142, Morristown, NJ, USA, 1990.

[MBDF89] Maria Helena Mira Mateus, Ana Maria Brito, Inês Duarte, and Isabel Hub Faria. *Gramática de Língua Portuguesa*. Série Lingüística. Caminho, 2ª edition, 1989.

[McC90] Michael McCord. Slot grammar: A system for simpler construction of practical natural language grammars. In *International Symposium on Natural Language and Logic*, pages 118–145, London, UK, 1990. Springer-Verlag.

[McC93] Michael McCord. Heuristics for broad-coverage natural language parsing. In *ARPA Human Language Technology Workshop*, University of Pennsylvania, 1993.

[McL01] Brett McLaughlin. *Java and XML*. O'Reilly, 2001.

[Mit01] Ruslan Mitkov. Outstanding issues in anaphora resolution. In *Proceedings of the Second International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2001)*, pages 110–125, 2001.

[Mit02] Ruslan Mitkov. *Anaphora Resolution*. Longman, 2002.

[Mit03] Ruslan Mitov, editor. *The Oxford handbook of computational linguistics*, chapter 14, pages 266–283. Oxford University Press, 2003.

[MS01] Christoph Müller and Michael Strube. Mmax: A tool for the annotation of multi-modal corpora. In *the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 45–50, Washington, USA, August 2001.

[MTB97] Tony McEnery, Izumi Tanaka, and Simon Botley. Corpus annotation and reference resolution. In *the Workshop on Operational Factors In Practical, Robust, Anaphora Resolution for Unrestricted Texts*, Madrid, Spain, July 1997.

[PMP+01] Manuel Palomar, Lidia Moreno, Jesús Peral, Rafael Muñoz, Antonio Ferrández, Patricio Martínez-Barco, and Maximiliano Saiz-Noeda. An algorithm for anaphora resolution in Spanish texts. *Computacional Linguistics*, 27(4):545–567, 2001.

[Ray03] Erik Ray. *Learning XML*. O'Reilly, 2003.

[Rei83] Tanya Reinhart. *Anaphora and semantic interpretation*. Croom Helm Ltd, 1983.

[RJB99] James Rumbaugh, Ivar Jacobson, and Grady Booch. *The Unified Modeling Language user guide*. Addison Wesley, 1ª edition, 1999.

[SG96] Mary Shaw and David Garlan. *Software architecture: Perspectives on an emerging discipline*. Prentice Hall, 1ª edition, 1996.

[Smi91] George W. Smith. *Computer and human language*. Oxford University Press, 1991.

[Wal89] Marilyn A. Walker. Evaluating discourse processing algorithms. In *the 27th Annual Meeting of the Association for Computational Linguistics*, pages 251–261, Morristown, NJ, USA, 1989.