Luana Vieira Morellato

Metodologia Computacional para Identificação de Sintagmas Nominais da Língua Portuguesa

Vitória - ES, Brasil

8 de janeiro de 2010

Luana Vieira Morellato

Metodologia Computacional para Identificação de Sintagmas Nominais da Língua Portuguesa

Dissertação apresentada para obtenção do Grau de Mestre em Informática pela Universidade Federal do Espírito Santo.

Orientador:
Sérgio Antônio Andrade de Freitas

DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória - ES, Brasil

8 de janeiro de 2010

Dissertação de Projeto Final de Mestrado sob o título *"Metodologia Computacional para Identificação de Sintagmas Nominais da Língua Portuguesa"*, defendida por Luana Vieira Morellato e aprovada em 8 de janeiro de 2010, em Vitória, Estado do Espírito Santo, pela banca examinadora constituída pelos professores:

Prof. Dr. Sérgio A. A. de Freitas
Orientador

Prof. Dr. Berilhes Borges Garcia
Universidade Federal do Espírito Santo

Profa. Dra. Aline Villavicêncio
Universidade Federal do Rio Grande do Sul

Resumo

Sintagmas são unidades de sentido e com função sintática dentro de uma frase, [Nicola 2008]. De maneira geral, as frases que compõem qualquer enunciado expressam um conteúdo por meio dos elementos e das combinações desses elementos que a língua proporciona. Dessa forma, vão se formando conjuntos e subconjuntos que funcionam como unidades sintáticas dentro da unidade maior que é a frase – os sintagmas, que podem ser divididos em sintagmas nominais e verbais. Dentre esses, os nominais representam maior interesse devido ao maior valor semântico contido.

Os sintagmas nominais são utilizados em tarefas de Processamento de Linguagem Natural (PLN), como resolução de anáforas (correferências), construção automática de ontologias, em *parses* usados em textos médicos para geração de resumos e na criação de vocabulário, ou ainda como uma etapa inicial em processos de análise sintática. Em Recuperação de Informação (RI), os sintagmas podem ser aplicados na criação de termos em sistemas de indexação e busca em documentos, gerando resultados melhores.

Esta dissertação propõe uma metodologia computacional para identificação de sintagmas nominais da língua portuguesa em documentos digitais escritos em linguagem natural. Nesse trabalho é explicitada a metodologia adotada para identificar e extrair sintagmas nominais por meio do desenvolvimento do SISNOP – Sistema Identificador de Sintagmas Nominais do Português. O SISNOP é um sistema composto por um conjunto de módulos e programas, capaz de interpretar textos irrestritos disponíveis em linguagem natural, através de análises morfológicas e sintáticas, a fim de recuperar sintagmas nominais. Além disso, são obtidas informações sintáticas, como gênero, número e grau das palavras contidas nos sintagmas extraídos.

O SISNOP testou, entre outros corpus, o CETENFolha, composto por mais 24 milhões de palavras, e o CETEMPúblico, com aproximadamente 180 milhões de palavras em português europeu, e muito utilizado em trabalhos da área. Foram obtidos 98,12% e 94,59% de frases reconhecidas pelo sistema, obtendo mais de 24 milhões de sintagmas identificados. Os módulos do SISNOP: EM Etiquetador Morfológico, ISN Identificador de Sintagmas Nominais e IGNG Identificador de Gênero, Número e Grau, foram testados de maneira individual utilizando um conjunto de dados menor que o anterior, visto que a análise dos resultados foi feita manualmente. O módulo identificador de sintagmas obteve precisão de 82,45% e abrangência de 69,20%.

Abstract

In Portuguese language, syntagmas are units of meaning and with syntactic function in a phrase [Nicola, 2008]. Generally speaking, phrases that compose any enunciate express some content through their elements and these elements combinations that the language allows. Therefore, sets and subsets are made and they work as syntactic units in the bigger unit which is the phrase - the syntagmas, that can be separated in noun phrase and verb phrase. Among those, the noun phrases represent a bigger interest due to the biggest semantic value in it.

Noun phrases are used in Natural Language Processing (NLP) tasks, such as resolving co-references (anaphora), automatic building of ontologies, in parsers used in medical texts to generate resumes and vocabulary building, or as an initial part in syntactic analyses processes. In Information Retrieval, noun phrases can be applied as atomic terms in indexing systems and documents search, delivering better results.

This dissertation proposes a computational methodology to identify noun phrases in digital documents written in natural language. This research explains the adopted methodology to identify and to extract noun phrase through the development of SISNOP (Portuguese Noun Phrase Identifying System - SISNOP, in Portuguese). SISNOP is a system composed by a set of modules and programs, that is able to interpret any kind of text available in the natural language, using morphological and syntactic analyses, in order to recover noun phrases. Besides that, the system obtains syntactic information, as gender, number and degree of the words in the extracted noun phrases.

The SISNOP tested, among other *corpora*, CETENFolha, composed by 24 million words, and CETEMPúblico, about 180 million words in European Portuguese and widely used in papers like of this study field. It was obtained 98,12% and 94,59% of sentences recognized by the system, getting up to 24 million identified noun phrases. The SISNOP modules: EM – Morphologic Tagger, ISN – Noun Phrases Identifier and IGNG – Gender, Number and Degree, were tested individually using a smaller set of data than the former one, because the results analyses were made manually. Noun phrase identifier module got 82,45% of precision and 69,20% of recall.

Dedicatória

Dedico este trabalho
ao meu pai, Wanderley, e à minha mãe, Odete.
Sem o amor, o carinho e a educação de vocês eu nada seria.

Agradecimentos

Agradeço a Deus, por sempre olhar por mim e por minha família e à minha irmã Alice, por quem tenho um amor sem medidas.

Ao Sérgio, por ser muito mais que um orientador. Obrigada pela confiança, pela paciência, por ensinar com a dedicação de um pai e por não me deixar órfã, mesmo quando foi necessária uma mudança de cidade.

Aos amigos que conheci na faculdade e não me largaram mais, Kbelo e Salomão. É impossível ser feliz sem a presença de vocês na minha vida, LvU. À Mary, que fez uma falta gigante, mas está de volta para nossa alegria. Você é a menor (em estatura) das PSPs, mas jamais a menos importante. Amo vocês. Aos demais amigos da graduação e do mestrado, pelo companheirismo. Ao calouro Lessandro pela competência na ajuda com a codificação.

À Débora, essa sim eu não tenho palavras para agradecer. Não só porque ela leu cada letra desta dissertação, mas também porque foi minha dupla em todas as matérias do mestrado, em muitos rocks para desestressar do mestrado, em idas ao *shopping* para gastar o dinheiro da bolsa do mestrado. Debs, muito obrigada por fazer parte da minha vida.

À Bruna, à Cris, à Magda e à Nick por estarem comigo na alegria e na tristeza, na saúde e na doença. Vocês são amigas incondicionais na minha vida. À Nessa e à Bruna que me acolheram em sua casa e escutarem minhas reclamações nos últimos meses.

Ao Marcus, pelo carinho e paciência diários, pela companhia, pelo colo e pelo gráfico. Não sei explicar por que, nem quando, nem onde, nem para quas você está na minha vida, e pode ter certeza de que gosto muito de você.

Aos amigos do iMasters, em especial às iLuluzinhas, Nath e Rina. À Nath, meu muito obrigada pelas correções, e à Rina, meu muito obrigada por sempre se importar e se oferecer para ajudar. Gosto muito de todos vocês.

Pelos amigos que mesmo longe, ou afastados pela correria do dia-a-dia, são extremamente especiais em minha vida: Carni, Caique, Domi, Fabs, Japa, Rato, Téfin, Taiane e Wellington.

A todos vocês os meus mais sinceros agradecimentos.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

"*A sabedoria é uma só, apenas a forma de expressá-la é que muda.*"
Mara Chan

Este capítulo apresenta a motivação e o objetivo deste trabalho, além de uma visão geral do que se encontra nesta dissertação.

1.1 Introdução

Esta dissertação propõe uma metodologia de identificação automática de sintagmas nominais da língua portuguesa. As frases que compõem qualquer enunciado expressam um conteúdo por meio dos elementos e das combinações desses elementos que a língua proporciona. Dessa maneira, formam-se conjuntos e subconjuntos que funcionam como unidades sintáticas dentro da unidade maior, que é a frase, chamadas de sintagmas [Nicola 2008]. A frase do Exemplo 1.1 tem o sintagma nominal *"A dúvida"* e o sintagma verbal *"é o princípio da sabedoria"*. O sintagma verbal é formado por um verbo (*"é"*) e por um sintagma nominal *"o princípio da sabedoria"* que, por sua vez, contém o sintagma preposicionado *"da sabedoria"*.

"A dúvida é o princípio da sabedoria." (1.1)

As expressões de maior poder discriminatório são, em geral, aquelas de sentido substantivo que podem realizar funções temáticas, como sujeito e objeto, e certas funções semânticas, como agente e instrumento, ou certas funções retóricas, como tópico ou tema. As expressões desse tipo são em grande parte sintagmas nominais, por isso é grande o interesse na extração automática dos mesmos [Oliveira e Quental 2003], como visto no Exemplo 1.1.

A identificação de sintagmas nominais em textos apresenta aplicações em diversos campos, como recuperação de informações, resolução de anáforas, ontologia, entre outros. Nessas aplicações, a precisão na identificação dos sintagmas é um componente crítico e está diretamente ligado à qualidade do resultado dos sistemas. O tempo de processamento é um fator importante a ser considerado, pois são programas utilizados, em sua maioria, em grandes bases de dados. O problema de identificação de sintagmas envolve frases simples como as duas do exemplo a seguir, em que a primeira contém 5 palavras e 2 sintagmas nominais, e a outra tem 62 palavras e 18 sintagmas nominais.

"A menina chegou em casa." (1.2)

"É um pouco a versão de uma espécie de outro lado de a noite, a meio caminho entre os devaneios de uma fauna periférica, seja de Lisboa, Londres, Dublin ou Faro e Portimão, e a postura circunspecta de os fiéis de a casa, que de ela esperam a música geracionista dos 60 ou dos 70."

Assim, esta dissertação apresenta a metodologia definida, os algoritmos envolvidos e as experimentações sobre a identificação de sintagmas nominais em textos da língua portuguesa através do desenvolvimento do SISNOP – Sistema Identificador de Sintagmas Nominais do Português. A seção a seguir apresenta as motivações para a realização do trabalho.

1.2 Motivação

Com o crescimento rápido na quantidade de dados disponíveis em formato digital, surge a necessidade de melhorar a qualidade das informações recuperadas através de estruturas mais complexas. Dentro desse contexto, torna-se uma opção processar automaticamente linguagem natural, para que se permita identificar estruturas como os sintagmas nominais, foco deste trabalho.

Os sintagmas nominais são de grande importância para trabalhos que fazem processamento de linguagem natural e buscam maior valor semântico agregado. Este trabalho foi motivado em especial pelos trabalhos desenvolvidos, na área de recuperação de informação, nesta universidade, por meio dos projetos de Seibel Júnior [Seibel Júnior 2007] e Pereira [Pereira 2009], que utilizam como base a estrutura proposta por Freitas [Freitas 2005], que, por sua vez, necessita dos sintagmas nominais identificados.

Na identificação de sintagmas existe uma variedade maior de trabalhos no inglês do que na língua portuguesa. Além disso, os trabalhos disponíveis, em sua maioria, utilizam aprendizado de máquina, em que é necessário ter uma base de dados marcada, o que gera um sistema dependente do domínio tratado. Em outros casos, os trabalhos que implementam sistemas utilizando regras gramaticais, limitam-se a processar conjuntos de dados pequenos, ou dependentes de uma ferramenta não disponível para uso em grande escala.

1.3 Objetivos

Esta dissertação tem como principais objetivos:

- Analisar as metodologias apresentadas em trabalhos da literatura direcionados à identificação de sintagmas na línguas portuguesa, considerando questões relacionadas à representação linguística dos sintagmas e aos métodos de implementação;
- Propor uma metodologia a partir da definição da técnica computacional e da forma de representação linguística, e implementar um sistema que recupere sintagmas nominais de

textos escritos em português, disponíveis de maneira textual em formato digital. Além disso, o sistema deve determinar as informações sintáticas dos elementos contidos nos sintagmas;

- Avaliar a utilização do sistema de identificação de sintagmas em corpora compostos por grandes quantidade de dados, considerando o tempo de processamento e a precisão das respostas, analisando assim a possibilidade de usá-lo em tarefas de processamento de linguagem natural, de maneira eficiente e não restritiva a um domínio.

1.4 Metodologia

Para a realização deste trabalho foram realizados estudos sobre os conceitos da área de linguística relacionados à definição da estrutura de textos e frases com foco em sintagmas nominais, além de estudos de teorias da área de Processamento de Linguagem Natural (PLN) sobre formas de representação léxica e sintática, e entendimento e construção de analisadores léxicos/morfológicos e sintáticos. Foi feito um levantamento dos trabalhos encontrados na literatura que desenvolviam a identificação de sintagmas e também dos que utilizam sintagmas em suas aplicações.

O comparativo entre as teorias de Perini [Perini 2003] e Liberato [Liberato 1997] sobre definição de sintagmas nominais serviram de base para a metodologia adotada. Assim, foi elaborado e implementado um algoritmo para o sistema de identificação automática de sintagmas nominais. Com o processo de testes, utilizando diferentes conjuntos de dados, foram descobertas deficiências na identificação de sintagmas relacionadas ao tempo de processamento e a abrangência da análise gramatical. Assim, métodos que aprimoram o processo de recuperação dos sintagmas na frase foram desenvolvidos, além de terem proporcionado otimização de tempo.

1.5 Estrutura da dissertação

Os dois primeiros capítulos desta dissertação, seguintes a este, contêm a revisão bibliográfica necessária para o entendimento do presente trabalho. No capítulo 2 são apresentadas abordagens de sistemas identificadores de sintagmas nominais tanto do português como da língua inglesa presentes na literatura. Além disso, são apresentados trabalhos de diversas áreas que fazem uso dos sintagmas nominais em aplicações relacionadas à recuperação de informação, à resolução de anáforas, à geração automática de ontologia e também a trabalhos utilizados na

área médica.

No capítulo 3 é apresentada uma contextualização da área linguística de estruturação de frase em sintagmas, além de definições relacionadas à análise gramatical nos campos léxico e sintático. O capítulo mostra também duas diferentes abordagens de identificação de sintagmas nominais: uma direcionada ao sintático e outra ao semântico e, a partir delas, apresentou a abordagem utilizada como base para o desenvolvimento da metodologia descrita nesta dissertação.

O capítulo 4 apresenta os algoritmos desenvolvidos para a construção do sistema identificador de sintagmas nominais do português, além dos programas utilizadas e dos métodos adotados para a melhoria na precisão e a abrangência da identificação e do tempo de processamento do sistema. O capítulo 5 apresenta os experimentos realizados que avaliaram os módulos do sistema de maneira qualitativa, além da utilização de conjuntos de dados maiores para uma avaliação quantitativa do sistema. No capítulo 6 são apresentadas as conclusões sobre este trabalho e algumas propostas de continuação.

2 *Revisão da Literatura*

"*Os computadores fazem apenas a apresentação da informação,*
porém o processo de interpretação fica a cabo dos seres humanos mesmo."
Karin Koogan Breitman, autor do livro "Web Semântica: A Internet do Futuro", 2006

Neste capítulo, são apresentados trabalhos da literatura que tratam da identificação de sintagmas nominais e de trabalhos que fazem uso dos sintagmas.

2.1 Introdução

Este capítulo apresenta uma revisão bibliográfica de trabalhos relacionados a sintagmas nominais. São descritos estudos destinados especificamente à identificação de sintagmas nominais, buscando mostrar a metodologia adotada, os conjuntos de testes utilizados e os resultados obtidos.

Os sintagmas nominais podem ser utilizados em diversas aplicações da área de processamento de linguagem natural. Neste capítulo, são apresentados trabalhos que empregam sintagmas nominais, relacionados à recuperação de informação, à resolução de anáforas, à geração automática de ontologia e à área médica. A seção seguinte destaca a importância do processamento de linguagem natural no contexto dos sistemas de informações.

2.2 Processamento de Linguagem Natural

O estudo de linguagem natural faz parte de muitas disciplinas além da linguística, incluindo tradução, crítica literária, filosofia, antropologia e psicologia. Cada uma aplica metodologias diferentes para reunir observações, desenvolver teorias e testar hipóteses. O grande desafio para a análise linguística computacional é apresentado pela explosão de texto e conteúdo multimídia na internet, considerando que uma grande e crescente parcela do tempo de trabalho e lazer é gasta navegando e acessando esse universo de informação.

O Processamento de Linguagem Natural (PLN), então, está sujeito a um rápido crescimento, e suas teorias e métodos têm sido utilizados em uma variedade de novas tecnologias da linguagem. Por essa razão é importante que muitas pessoas tenham um conhecimento efetivo do PLN. No meio acadêmico, esse grupo inclui pessoas das áreas de corpus linguístico, de ciência da computação e de inteligência artificial. No meio industrial, esse grupo inclui, por exemplo, pessoas que trabalham com interação homem-computador, analistas de informações de negócios e desenvolvimento de *software*, principalmente para o ambiente *web*.

O diagrama visto na Figura 2.1 mostra os fluxos de produção e de compreensão de linguagem natural que envolvem tanto os níveis concretos como abstratos. Seguindo o fluxo do lado esquerdo, tem-se o caminho de alguns componentes do reconhecimento de fala que mapeiam a entrada oral para algum tipo de representação de significado. Pelo lado direito do diagrama, há um caminho inverso de componentes para a geração de fala a partir de um conceito. Na coluna central do diagrama, podem ser encontrados alguns repositórios de informações relacionadas à linguagem que são utilizadas pelos componentes de processamento.

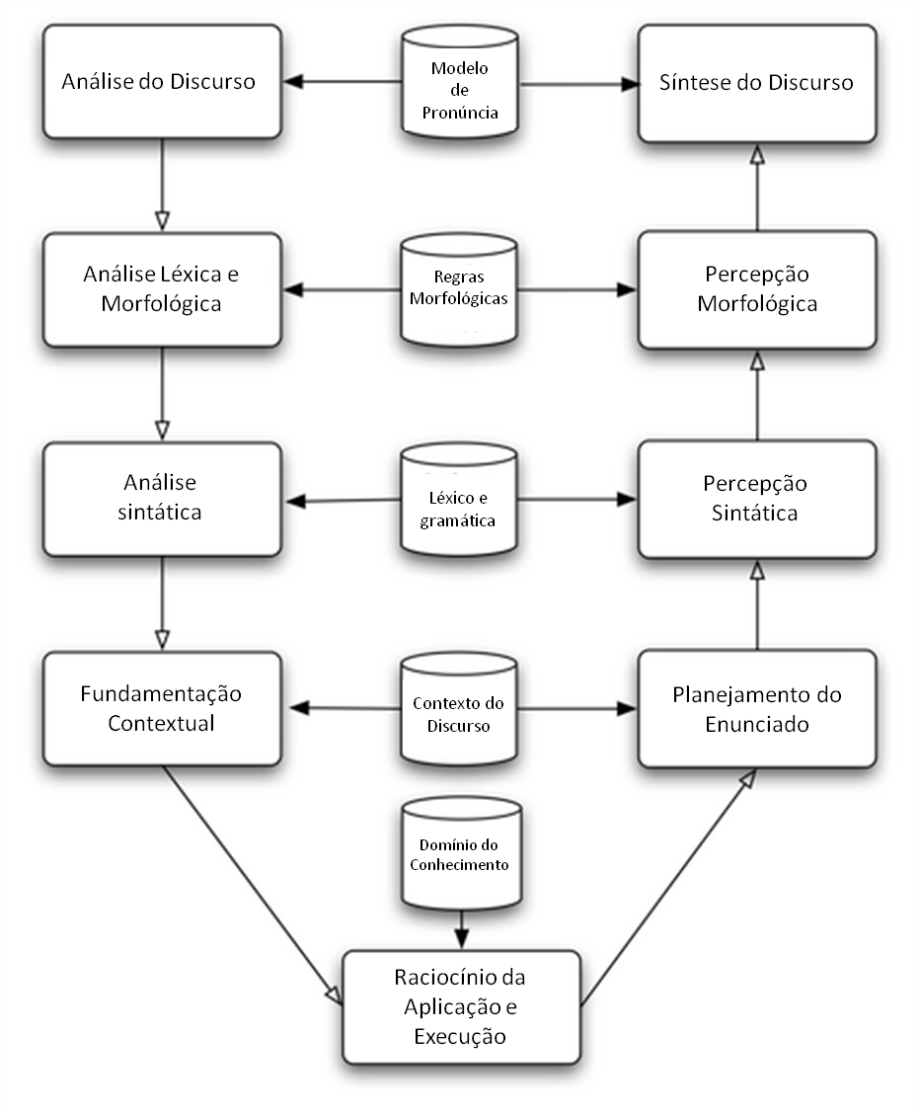


Figura 2.1: Diagrama de produção e compreensão de linguagem natural

O diagrama também ilustra formas de modularizar o conhecimento linguístico em sistemas computacionais, ou seja, os vários componentes são organizados de tal forma que os dados que eles compartilham correspondem aproximadamente aos diferentes níveis de representação. A saída do componente de análise da fala, por exemplo, irá conter sequências de representações léxicas das palavras, e a saída do contexto do discurso será uma representação semântica. A maioria dos sistemas de PLN divide seu processamento em um conjunto de passos semelhantes ao diagrama descrito.

Na década de 80, começaram a ser desenvolvidos os primeiros sistemas computacionais de busca em documentos. No início, as palavras-chave de cada texto eram retiradas manualmente, e a busca era feita nesse pequeno conjunto de dados. Os principais utilizadores desses sistemas eram bibliotecas e arquivos, logo, a área não atraía grande interesse até o surgimento

da Internet. O novo conceito de repositório universal de conhecimento, de acesso livre ou de baixíssimo custo, sem um corpo editorial central, gerou uma verdadeira explosão na produção e no consumo de textos.

Os maiores problemas dessa grande base de dados estão relacionados à organização e à recuperação de itens, portanto são questões cujas soluções encontram-se dentro do escopo da área Recuperação de Informações (RI) [Baeza-Yates e Ribeiro-Neto 1999]. Com o crescimento contínuo da Internet, a área de recuperação de informação vem desenvolvendo métodos cada vez mais poderosos e completos de indexação e busca. Muitas dessas novas tecnologias são baseadas em PLN, ligadas principalmente à área de processamento automático de textos. Antes de 1995 os estudos eram mais intensificados. Após o surgimento de grandes buscadores online (AltaVista, Yahoo! e Google principalmente) os estudos de PLN voltados para a recuperação de informação diminuíram e, anos depois, voltaram com outra ênfase.

2.3 Identificação de Sintagmas Nominais

A identificação de sintagmas nominais é uma tarefa que exige um grande esforço linguístico, devido à necessidade de especificar toda a regra de formação dos mesmos, de acordo com a teoria linguística, para só então ser possível a recuperação dos sintagmas de forma computacional. A seguir, são descritos trabalhos formulados com este foco: identificar e recuperar sintagmas nominais por meio de sistemas automáticos. Os estudos pioneiros encontrados na literatura são da língua inglesa, e alguns desses trabalhos são descritos na seção seguinte. Logo em seguida são apresentados os trabalhos relacionados à identificação de sintagmas nominais do português.

2.3.1 Sintagmas nominais na língua inglesa - *Nouns Phrase*

Em inglês, os sintagmas nominais, chamados de *nouns phrase* (NP), também são extraídos e aplicados de maneira semelhante ao português. As línguas portuguesa e inglesa não são totalmente dissimilares no nível sintático. No geral, a ordem das palavras, com relação às classes gramaticais, é similar. Algumas regras abrangem as duas línguas, mas pode ocorrer diferença na estrutura que define um sintagma nominal, como o fato de que a posição do adjetivo no inglês é preferencialmente pré-nominal e, no português, pós-nominal. Uma outra diferença também entre as definições de SN e NP é que em inglês o substantivo pode ocupar a posição de um predicado adjetivo, como em *"door mat"*, mas em português o mesmo tipo de sintagma nominal precisa de uma preposição, como em *"tapete da porta"*.

A noção de sintagma nominal básico tem sido usada na língua inglesa e definida como um sintagma não recursivo que inclui determinantes e pré-modificadores, mas não inclui pós-modificadores como sintagmas preposicionados e orações subordinadas. No português, essa noção provê um conjunto de sintagmas muito pobre e, para se obter SNs mais informativos, é necessário considerar sintagmas recursivos. Assim, o problema de identificar sintagmas nominais no português torna-se mais complexo do que na língua inglesa.

Os trabalhos voltados à identificação de sintagmas nominais na língua inglesa são anteriores aos estudos para a língua portuguesa, e alguns serviram de base para o início dos trabalhos no português. A seguir, são detalhados dois desses trabalhos direcionados à tarefa de extração de *nouns phrase*: [Voutilainen 1995] e [Cardie e Pierce 1998].

Voutilainen [Voutilainen 1995] desenvolveu uma ferramenta chamada *Nptool* para extrair *noun phrases* de textos escritos na língua inglesa, tendo como objetivo principal auxiliar a tarefa de reconhecer palavras representativas para a formação de índice do texto. O *Nptool* envolve etapas de pré-processamento, análise léxica, *parsing* de gramáticas de restrições para tratar desambiguação morfológica e restrição sintática, além de dois *parsers* de NP, que após a extração, fazem a interseção das respostas. O léxico do sistema foi bem desenvolvido contendo mais de 56 mil radicais de palavras e obteve 95% de precisão na extração de sintagmas. Atualmente, o *Nptool* não está mais disponível, somente sendo fornecida manutenção para quem já havia adquirido o *software*.

A extração de *noun phrases* pelo *Nptool* segue a metodologia de sistemas clássicos de PLN, sendo um sistema bastante completo e complexo. As análises léxica e sintática são feitas sobre todas as palavras do texto. O léxico possui as etiquetas morfológicas e sintáticas das palavras (e também etiquetas diferentes para as ambiguidades), e o sintático apresenta regras para toda a gramática da língua inglesa. Miorelli [Miorelli 2001], ao detalhar o trabalho de Voutilainen [Voutilainen 1995], vê como grande vantagem o fato de ele extrair outras estruturas das sentenças, uma vez que implementa regras para toda a gramática. A dificuldade de aplicar a mesma técnica à língua portuguesa se deve à maior complexidade do idioma em relação ao inglês.

Por meio do corpus do *Penn Treebank Wall Street Journal* [Penn Treebank Wall Street Journal], que possui os *noun phrases* já marcados, Cardie e Pierce [Cardie e Pierce 1998] utilizaram um etiquetador para encontrar as classes gramaticais das palavras e, a partir disso, extrair regras que são compostas das combinações das categorias das palavras presentes nos NPs marcados. O sistema, ao receber um novo documento para identificação dos *noun phrases*, utiliza o mesmo etiquetador (para marcar as classes gramaticais) e aplica as regras produzidas no processo an-

terior. O sistema é composto de três fases: uma de definição de regras (processamento sobre o corpus com os *noun phrases* marcados e processamento sobre um novo corpus), uma de aperfeiçoamento da gramática para o NP e uma de avaliação.

Essa metodologia parece ser aplicável à língua portuguesa, já que o processo de extração é simples. Os principais erros são oriundos das regras geradas automaticamente que, depois de produzidas, devem ser alteradas manualmente. Um outro problema é a limitação de domínio e a necessidade de um corpus marcado. Se as regras geradas a partir de um conjunto de textos forem aplicadas a textos de outros domínios, os resultados possivelmente não atingirão um nível de satisfação equivalente.

2.3.2 Sintagmas nominais na língua portuguesa

Um dos primeiros estudos relacionados à identificação e à extração de sintagmas do português, assim como sua utilização em trabalhos de PLN, foi feito por Kuramoto. Em [Kuramoto 1995], é proposta uma nova interface nos sistemas de buscas textuais, além disso, é defendida uma abordagem alternativa no tratamento de recuperação de informação utilizando sintagmas nominais, até então não citados na língua portuguesa para este fim.

Os sistemas tradicionais existentes possuíam interfaces, em sua maioria, orientadas a comandos com regras rígidas de utilização e que exigiam dos usuários um grande domínio de conhecimento relacionado à informática ou à estruturação dos dados. Na mesma época do estudo de Kuramoto, começaram a surgir os sistemas de busca e recuperação de informação em documentos e textos digitais, que tornaram-se os buscadores *online* atuais. As ferramentas AltaVista e Yahoo! surgiram em 1994, Cadê? em 1995 e Google em 1998.

O processo de extração de sintagmas, em [Kuramoto 1995], foi feito de forma manual, simulando um processo automatizado devido à não existência ainda de um sistema de extração de sintagmas nominais em acervos contendo documentos em língua portuguesa. Com o desenvolvimento do protótipo do sistema de busca, foi possível tornar o acesso à informação uma tarefa simples e convergente para o usuário, utilizando, para isso, os sintagmas nominais como estrutura de acesso à informação contida nas bases de dados textuais. Quanto à extração automática dos SNs, Kuramoto levantou questões relativas à resolução dos elementos anafóricos, à resolução de elipses e à identificação dos SNs sem determinação (que não são precedidos de artigos definidos e indefinidos), fato comum na língua portuguesa.

Anos depois, com o avanço nos sistemas de busca, Kuramoto focou sua pesquisa na utilização de sintagmas nominais como uma proposta para recuperação de informação, [Kuramoto 1999]

e [Kuramoto 2002]. A maioria dos modelos anteriores considera a utilização de palavras como meio de acesso à informação. Esses trabalhos mostram a inadequação do uso das palavras nesses modelos, propondo, em seu lugar, o uso de sintagmas nominais.

Kuramoto discute que considerar sintagmas nominais na recuperação de informação oferece duas alternativas possíveis de implementação, em termos de indexação automática e de interfaces de busca. Uma primeira alternativa seria implementar uma indexação automática nos moldes daquela tradicional baseada em palavras, apenas substituindo os índices contendo as palavras isoladas por índices contendo sintagmas nominais. Essa alternativa inclui também a possibilidade de utilizar os modelos de classificação ou *ranking*, como o modelo vetorial, aplicando os sintagmas nominais como unidade básica de acesso à informação. Uma segunda alternativa seria o aproveitamento da organização hierárquica em árvore dos sintagmas nominais, não apenas criando um novo conceito em termos de indexação, como também introduzindo inovação em termos de uma interface de busca.

A proposta apresentada por Kuramoto, [Kuramoto 1999], compreendeu o desenvolvimento de um protótipo de interface de busca utilizando os sintagmas nominais como forma de acesso à informação. Para testar esse protótipo, foram extraídos manualmente cerca de 8800 sintagmas nominais de uma amostra de 15 artigos selecionados aleatoriamente devido à não existência no momento de nenhum *software* capaz de reconhecer, extrair e indexar os sintagmas nominais. Os resultados obtidos com a implementação do protótipo comprovaram a viabilidade técnica de implementar uma interface de busca capaz de navegar em uma estrutura hierárquica em árvore de sintagmas nominais.

Miorelli, em [Miorelli 2001], extrai sintagmas nominais de textos em português e os utiliza como base na formação das palavras-chave ou expressões-chave para representar o conteúdo dos documentos. O método proposto utiliza um módulo seletor que recupera da sentença um candidato a sintagma nominal e o envia ao módulo analisador. Os sintagmas, que estão de acordo com as regras gramaticais segundo abordagem Perini [Perini 1995], são então extraídos. O módulo seletor, proposto por Miorelli, encontra candidatos eliminando palavras com etiquetas gramaticais que conhecidamente não fazem parte de um sintagma nominal. As classes de palavras que são definidas como classes que não pertencem a um sintagma nominal (denominadas etiquetas de corte) são: verbos, sinais de pontuação e locuções.

O trabalho de Souza, [Souza 2005], apresenta uma metodologia para viabilizar o processo de atribuição de descritores a textos digitalizados. Esse processo, chamado indexação, extrai sintagmas nominais e analisa fatores como a frequência de ocorrência desses sintagmas em relação ao texto e ao conjunto de documentos. Os processos mais comuns de indexação

automática descrevem os documentos através de uma lógica simplista, advinda da análise de frequência das palavras que neles ocorrem. Visando a uma maior eficácia, [Souza 2005] propõe um processo de indexação, que analisa as palavras e as expressões no âmbito de seus contextos linguísticos, assumindo que a utilização de sintagmas nominais como descritores apresenta vantagens em relação ao uso de palavras-chave.

Souza utilizou os sintagmas nominais do corpus de 15 documentos do trabalho de Kuramoto, [Kuramoto 1999], para testar o processo de extração automática e um corpus de 60 documentos provenientes de publicações eletrônicas da área de ciência da informação. Os resultados apresentados demonstraram grande pertinência dos descritores atribuídos aos documentos e permitiram concluir que a metodologia obtém sucesso nas condições estudadas.

Santos [Santos 2005] propõe a identificação automática de sintagmas nominais como uma tarefa de classificação a ser automaticamente aprendida com o uso de uma técnica chamada Aprendizado Baseado em Transformações (do inglês *Transformation Based Learning* – TBL). O aprendizado de máquina utilizado é guiado por um corpus de treino que contém exemplos corretamente classificados. A própria classificação dos exemplos foi derivada automaticamente de um corpus preexistente. O conhecimento linguístico gerado por essa técnica consiste de uma lista ordenada de regras de transformação, que pode ser utilizada para a classificação de novos textos.

Para a redução dos erros de classificação de preposições, Santos propôs um novo tipo de molde de regras, cuja unidade básica, aqui chamada de termo atômico, possui uma janela de contexto de tamanho variável e um teste que precede a captura dos valores que compõem as regras. Também foi mostrado que o uso de uma classificação inicial mais precisa e de um programa que encontra o lema dos verbos (sua forma infinitiva) contribuem para a redução do tempo de treinamento, e, ainda, trazem alguns benefícios para a eficácia das regras aprendidas. Na identificação de sintagmas nominais do português brasileiro com o uso da ferramenta de TBL desenvolvida foi obtida uma precisão de 86.6% e uma abrangência de 85.9%, no melhor caso.

[Oliveira et al. 2006] propõe a utilização da técnica de *chuncking* como uma solução rápida e robusta para a identificação de sintagmas nominais em textos. Um "*chunk* SN" é, por definição, um SN não recursivo, ou um SN que não contém outro SN descendente. Mesmo sendo essas condições muito restritivas para o sintagma do português, utilizando corpus anotado e aprendizado de máquina baseado em transformações TBL, foram obtidos bons resultados.

Morellato, em [Morellato 2007], desenvolveu o SIDSN, um sistema de identificação de sintagmas composto de dois módulos, um de pré-processamento do texto que o estrutura em

frases, e outro que realiza a análise sintática da sentença e extrai os sintagmas nominais identificados. A análise sintática foi feita utilizando DCG – *Definite Clause Grammars* – Prolog, e as informações léxicas de cada palavra eram buscadas em um banco de dados.

Tendo como base o trabalho de [Morellato 2007], Bastos em [Bastos 2007] desenvolveu um sistema utilizando um *tagger* para classificar as palavras, em vez de um dicionário de dados. As regras sintáticas foram melhoradas e não mais implementadas em Prolog. Essas mudanças geraram uma melhora nos resultados, já que um *tagger* trata melhor problemas de ambiguidade, além de ter processamento mais rápido do que um acesso ao banco de dados. O Prolog é uma linguagem de programação lógica bastante popular no desenvolvimento de *parsers* para linguagens naturais, devido seu suporte nativo para buscas e para a operação de unificação, que combina duas estruturas características em uma. Entretanto, o processo de entrada e saída de dados não é facilitado, além de ter um tempo de processamento consideravelmente grande. Bastos conseguiu resultados melhores que Morellato e uma queda no tempo de processamento, sendo isso muito importante para o direcionamento do presente trabalho.

[Costa 2007] descreve a implementação de um fragmento do português em uma gramática computacional – LXGram –, desenvolvida na Universidade de Lisboa. A LXGram é uma gramática computacional para o processamento linguístico profundo do português e pode ser utilizada para analisar frases, produzindo uma descrição formal do seu significado, ou para gerar frases a partir de representações do significado.

A LXGram é desenvolvida em uma plataforma, o *Linguistic Knowledge Builder* (LKB), desenhada especificamente para acomodar tais gramáticas. O LKB implementa algoritmos eficientes de análise e geração, e aceita um formalismo declarativo em que a operação fundamental é a unificação. Outras gramáticas desenvolvidas no LKB vêm sendo integradas em aplicações úteis, como tradução automática, sistemas de respostas automáticas a correio eletrônico, corretores gramaticais e extração de informação. Souza modelou e implementou um conjunto de fenômenos linguísticos que relacionam-se com as propriedades gramaticais e com o significado dos sintagmas nominais e focou-se em alguns aspectos que não estão muito desenvolvidos nas outras gramáticas implementadas no LKB.

2.4 Aplicação de sintagmas nominais

Os modelos discutidos de identificação de sintagmas nominais buscam obter altos valores de precisão e de abrangência para que possam ser aplicados em sistemas desenvolvidos em diversas áreas do conhecimento. A seguir são mostrados estudos da aplicação de sintagmas

nominais em recuperação de informação (descritores de textos, índices, palavras-chave) na resolução de anáforas, na geração automática de ontologia e também um destaque para trabalhos utilizados na área médica.

2.4.1 Recuperação de Informação

Brants em [Brants 2003] fez um estudo do uso de técnicas de PLN em recuperação de informação e mostrou que os resultados não são encorajadores. Brants questiona que métodos simples não proveem resultados significantes, enquanto técnicas mais complexas não são eficientes para a melhora que podem obter, e que estudos de alto nível possuem um custo muito alto de performance, não possibilitando o uso em grandes coleções de documentos.

Brants sugere que, para que as melhorias nos resultados sejam maiores, deve-se pensar em processamento de linguagem natural junto com recuperação de informação, e não desenvolver sistemas PLN separados e posteriormente utilizar como caixa-preta por RI. Ele mostrou que técnicas de processamento que foram desenvolvidas diretamente para recuperação de informação tendem a ter mais sucesso que técnicas que foram desenvolvidas independentemente, baseada em linguística. Além disso, provou que técnicas como *POS-tagger*, *chunking* e *parsing* podem ser adaptadas para tarefas de recuperação de informação ou domínios particulares. Alguns dos trabalhos descritos foram desenvolvidos dessa forma, outros não, como pode ser visto a seguir.

Evans e Zhai, em [Chengxiang, Evans e Zhai 1996], avaliam o uso de um simples, mas robusto e eficiente, sistema de identificação de sintagmas nominais da língua inglesa. Voltado principalmente à recuperação de informação, o sistema analisa o fato de os sistemas de RI (exceto o CLARIT) usarem indexação de palavras para representação de documentos, já que esta técnica é mais fácil e eficiente do que identificar estruturas mais complexas. O sistema trabalha com uma extensão do *software* CLARIT, fazendo uma extração de subcomponentes mais simples de sintagmas nominais complexos.

Além disso, os autores discutem sobre os pré-requisitos para se obter eficácia no processamento de linguagem natural em aplicações de recuperação de informação. Os sistemas devem ter a habilidade de processar uma grande quantidade de documentos, sendo eficientes nas complexidades de tempo e espaço. Eles devem ser capazes, também, de interpretar textos irrestritos que englobem diferentes domínios. E deve-se levar em consideração que nem sempre é necessário um processamento de toda a estrutura semântica da frase, por vezes muito complexa, se a aplicação que se deseja utiliza somente os sintagmas nominais.

A aplicação de algoritmos de aprendizado de máquina em tarefas de identificação de SN tem

produzido ótimos resultados experimentais, [Oliveira e Freitas 2006]. Embora, em sua maioria, os métodos sejam gerais o suficiente para serem aplicados a uma ampla variedade de línguas, existem especificidades linguísticas que podem influenciar no seu desempenho. O modelo de sintagmas nominais descrito por Oliveira e Freitas foi desenvolvido para superar dificuldades decorrentes da transposição da língua-alvo do inglês para português. Como em português os sintagmas nominais são, em média, maiores e contêm mais preposições, a motivação principal foi ampliar o conceito de *chunk* a fim de considerar um conjunto mais significativo de SNs, mantendo a eficácia do método para textos em língua portuguesa.

Sintagmas nominais extraídos a partir de textos foram utilizados como conceitos em algoritmos de agrupamentos para recuperação de informação [Moraes e Lima 2008]. Por meio de uma abordagem não-supervisionada, o trabalho de Moraes e Lima categorizou as palavras, utilizando um *POS-tagger*, e a seguir fez uso de um *shallow parser* implementado para identificar os sintagmas nominais. A análise dos conceitos identificados foi feita manualmente e uma ferramenta auxiliou o agrupamento para a extração de conceitos. Apesar de problemas relatados na geração dos grupos de conceitos, os resultados do trabalho mostraram a viabilidade da metodologia.

2.4.2 Resolução de anáforas

Anáfora é o fenômeno linguístico de referenciar um item previamente mencionado no texto, sendo um recurso frequente em documentos escritos em linguagem natural. O processo para resolução de uma anáfora consiste em, uma vez que foram identificados o antecedente e a anáfora, estabelecer um relacionamento entre os dois. A resolução de anáforas é importante para diversas aplicações de PLN como, por exemplo, geração automática de resumo, tradução automática e, ainda, recuperação automática de informação.

Para alguns tipos de anáforas, como as pronominais (pronome pessoal ou demonstrativo), somente informações léxicas sobre os termos constituintes das orações são suficientes para a identificação do antecedente da anáfora. Anáforas nominais definidas exigem mais informação para a sua resolução, principalmente quando o relacionamento entre a anáfora e o seu antecedente não é uma correferência (duas ou mais referências identificam o mesmo referente). Essas situações podem exigir conhecimento sobre a semântica dos termos envolvidos para que seja possível realizar a resolução.

O trabalho de Vieira et al [Vieira et al. 2000] trata da extração semiautomática de sintagmas nominais para resolver correferência, utilizando as árvores sintáticas geradas pelo *software* do projeto *Visual Interactive Syntax Learning* (VISL), da Universidade de Aarhus na Dinamarca.

O corpus usado nesse trabalho é constituído por um conjunto de 15 textos do Jornal Correio do Povo (aproximadamente 5 mil palavras). O estudo apresenta um detalhamento sobre as descrições definidas (sintagmas nominais iniciados por artigo definido) na língua inglesa e tem sido usado como base para trabalhos sobre a resolução de descrições definidas da língua portuguesa. Seu objetivo é, primeiramente, quantificar o uso de descrições definidas no corpus. Como resultado, o trabalho classifica aproximadamente 50% dos sintagmas extraídos como descrições definidas, taxa que os autores afirmam ser suficiente para justificar seu propósito.

O método desenvolvido inicia submetendo os textos ao *software VISL*, que gera a árvore sintática das sentenças em que os sintagmas nominais estão contidos. Um programa em linguagem C gera, a partir da árvore sintática, listas com sintagmas nominais em linguagem Prolog. A lista dos sintagmas passa por uma revisão manual, na qual os erros encontrados (10%) foram corrigidos manualmente. Ao final, a lista de sintagmas serve como entrada no sistema para anotação automática de correferência.

[Soon et al. 2001] apresenta um método de resolução de correferência de sintagmas nominais em textos irrestritos utilizando aprendizado de máquina. As tarefas de processamento de linguagem natural, como definição de *tokens*, processamento morfológico, utilização de *POS-tagger*, identificação de sintagmas, entre outras, servem para determinar os candidatos à resolução de correferência. O *POS-tagger* é baseado no modelo da Cadeia de Markov e, de maneira similar, foi construído o módulo que identifica os sintagmas nominais.

O treino do método de aprendizado de máquina, em [Soon et al. 2001], foi realizado em um pequeno corpus, anotado, e a resolução não foi restrita às anáforas pronominais. Segundo os autores, esse foi o primeiro sistema supervisionado que ofereceu performance comparável com o estado da arte de sistemas não-supervisionados com os dados testados.

Em [Angheluta et al. 2004], são apresentados quatro algoritmos de agrupamento (*clustering*) para resolução de correferência de sintagmas nominais em textos. O foco está na detecção de um tipo específico de anáfora, o relacionamento de identidade. Duas entidades são ditas correferentes se ambas se referem ao mesmo sintagma nominal em uma situação; por exemplo, em: *"Sérgio conheceu sua esposa no colégio. O professor de matemática casou-se com ela após ele terminar os estudos."*, a expressão *"professor de matemática"* e *"ele"* se referem a *"Sérgio"*. O método, primeiro, extrai as entidades (sintagmas nominais), e um conjunto de propriedades relacionados a elas: posição no texto, categoria gramatical, gênero e número. A seguir é feito o agrupamento utilizando uma variação mais complexa da lógica *fuzzy*, a qual é comparada com dois algoritmos da literatura. Um projeto futuro proposto é a integração da ferramenta a um sistema de sumarização de textos.

Freitas apresenta em [Freitas 2005] a abordagem de resolução de anáforas baseada em regras pragmáticas para a identificação dos antecedentes anafóricos e proporciona uma metodologia para a obtenção de uma representação estruturada do texto. A Estrutura Nominal do Discurso (END) surge a partir da interpretação de um documento, no qual elementos linguísticos que sugerem a utilização de anáforas, tais como pronomes, elipses e sintagmas nominais definidos, são identificados juntamente com os antecedentes candidatos ao estabelecimento de uma relação anafórica. Para a resolução de uma anáfora, é necessário que o leitor identifique um relacionamento entre o antecedente da partícula anafórica e a própria partícula anafórica.

Freitas propõe que para a interpretação automatizada de textos e, consequentemente a resolução de anáforas, seja identificada a relação entre a partícula anafórica e seu antecedente. A relação deve ser categorizada com base em cinco relações básicas: correferência, membro-de, parte-de, subcategorização e acomodação. Na interpretação em contexto, o segmento criado é interpretado com base nos outros segmentos já interpretados na estrutura. Com isso, caso realmente exista uma anáfora na frase, é possível identificar qual entidade é o seu antecedente.

Seibel Júnior, em [Seibel Júnior 2007], apresenta a utilização da estrutura provida pela teoria de Freitas [Freitas 2005] na recuperação de informação, apresentando uma metodologia para a realização de buscas nessa estrutura. Seibel propõe uma modificação na estrutura de maneira que armazene somente os termos indexados e seus valores de relevância para o documento.

[Pereira, Morellato e Freitas 2009] apresentam um modelo de recuperação estrutural de informação utilizando, também, a END. O trabalho fez uso de sintagmas nominais a fim de permitir uma melhor representação de texto. Esse trabalho buscou mostrar os benefícios que a área de recuperação de informação alcança ao utilizar a estrutura nominal do discurso, e mostrou, também, uma comparação do sistema desenvolvido, baseado em anáfora, com o tradicional modelo vetorial.

A partir também da proposta de Freitas [Freitas 2005], e das modificações propostas por [Seibel Júnior 2007], em [Pereira, Seibel Júnior e Freitas 2009] é apresentada uma nova metodologia para a RI baseada na resolução de anáforas. A construção da estrutura para buscas é realizada transpondo todas as entidades identificadas durante o processo de resolução anafórica, o que possibilita uma melhora na forma de representação do texto dos documentos e na qualidade dos resultados obtidos pelas pesquisas. Pereira, em [Pereira 2009], detalha a proposta descrita em [Pereira, Seibel Júnior e Freitas 2009], apresentando os algoritmos envolvidos na sua definição e experimentações sobre a nova metodologia de buscas baseada na resolução de anáforas.

Os últimos seis trabalhos descritos utilizam sintagmas nominais como termos candidatos na

resolução de anáforas por meio da END (Estrutura Nominal do Discurso) e foram desenvolvidos no Departamento de Informática da Universidade Federal do Espírito Santo.

2.4.3 Ontologia

Ontologia tem sido empregada para conceituar, estruturar e representar, em um documento, o conhecimento de um domínio, de maneira que possa ser compartilhado. Entretanto, é sabido que a construção de ontologia é um processo trabalhoso, que exige muito tempo e esforço, principalmente para utilização em larga escala. Alguns estudos voltados à geração automática de ontologias utilizam como base a identificação de sintagmas para descrever conceitos.

Duque propõe, em [Duque 2006], uma abordagem ontológica para indexação de textos em português, utilizando um sistema de recuperação de informação baseado em processamento de linguagem natural. A indexação é focada na extração de rótulos sintáticos para a geração de rótulos semânticos e, posteriormente, no desenvolvimento de uma ontologia leve.

Os textos são convertidos para o formato XML, a seguir as classes gramaticais das palavras são reconhecidas e, logo depois, são identificados os sintagmas nominais e verbais. Os sintagmas nominais foram utilizados na geração de subclasses de duas classes da ontologia durante o experimento de validação. O estudo fez uso das ferramentas PALAVRAS e Protégé [Protégé], e desenvolveu um sistema chamado GeraOnto. A metodologia desenvolvida foi comparada com o modelo vetorial na recuperação de informação, apresentando resultados melhores em 22.75% de precisão e 15.00% de *recall*.

O estudo de [Lopes et al. 2009] mostra o desenvolvimento e utilização da ferramenta OntoLP no processo de construção de ontologias em um experimento na área da saúde. OntoLP é um *plug-in* para a ferramenta Protégé [Protégé], um editor que dá suporte à construção de ontologias, seguindo as tecnologias da *Web* Semântica, como, por exemplo, a construção de ontologias em linguagem OWL *Ontology Web Language*, conforme o padrão definido pelo *World Wide Web Consortium* (W3C).

O processo de construção automática de ontologias é dividido em 5 etapas, sendo os sintagmas nominais utilizados na primeira delas: a extração de termos candidatos a conceitos de um domínio. Após a extração dos termos é feita uma comparação com os resultados de referência de uma lista de termos construída manualmente. Foram analisados bigramas e trigramas obtidos através de diferentes métodos. Por fim, são observadas vantagens no processamento com inclusão de informação linguística complexa, como análise sintática e semântica. As técnicas avaliadas foram incorporadas ao editor de ontologias Protégé, por meio do plug-in OntoLP.

2.4.4 Área médica

Os trabalhos descritos a seguir mostram a utilização de identificação de sintagmas nominais da língua inglesa (*noun phrases*) em aplicações da área médica. Eles obtiveram melhora na recuperação de informação utilizando sintagmas nominais de forma específica para o domínio médico, mas também alcançaram bons resultados sem essa especificidade.

[Spackman e Hersh 1996] avalia dois *parsers*, CLARIT e Xerox *Part-of-Speech Tagger*, para identificar sintagmas nominais em resumos médicos. Sem modificar os sistemas para o domínio médico, eles foram testados em um conjunto de 20 textos obtendo um resultado de 77.0% de precisão na identificação exata da frase e 85.7% na identificação parcial, sendo esses valores referentes ao *parser* que alcançou o melhor resultado, o CLARIT [CLARIT]. O estudo sugere, baseado nos resultados, a utilização dos SNs para controle e criação de um vocabulário médico utilizando o sistema em um corpus bem maior. O *parser* CLARIT foi desenvolvido com o objetivo de ser rápido na identificação de sintagmas nominais em grandes corpora.

Com objetivo semelhante, [Bennett et al. 2004] discute o desenvolvimento de um *parser* de linguagem natural que seja capaz de extrair sintagmas nominais para todos os textos médicos, ajudando assim a análise do conteúdo para recuperação de informação. Utilizando um *parser* não especializado para o domínio médico, o trabalho extraiu sintagmas nominais no MEDLINE, a principal base de dados bibliográficos da *National Library of Medicine*, com aproximadamente 9,3 milhões de resumos.

O *parser* é composto de um módulo que define os *tokens* e remove as pontuações, um *part-of-speech tagger* e um extrator de sintagmas nominais composto de um conjunto de regras, que foram baseadas na ferramenta comercial *NPTool* [Voutilainen 1995]. O módulo que define os *tokens* foi modificado visando a cometer uma quantidade menor de erros provindos da nomenclatura médica e, da identificação das frases. [Bennett et al. 2004] utilizou um método de janela deslizante com o tamanho máximo de sete palavras. A precisão e o *recall* foram melhores em comparação a outros três *parsers* da língua inglesa previamente avaliados.

Em [Huang et al. 2005], foi desenvolvido e avaliado um método de extração de sintagmas nominais em um conjunto de relatórios médicos da área de radiologia. O sistema implementado utilizou a estrutura de analisar a frase inteira, e não somente partes dela, e buscou melhorar os resultados utilizando uma ferramenta léxica especializada no domínio da medicina.

O módulo desenvolvido é composto de um separador de texto em frases, por meio de um detector de fronteiras de sentença, de um *parser* estatístico treinado em domínio médico e de um *tagger* que identifica os sintagmas nominais. Os resultados foram melhores, obtendo maior

precisão e *recall*, se comparado a *parsers* de domínio geral e com outros estudos da área médica.

2.5 Considerações Finais

Este capítulo descreveu alguns trabalhos da literatura que desenvolveram sistemas de identificação de sintagmas nominais na língua portuguesa e na inglesa. Os estudos na área são mais aprofundados e em maior número para a língua inglesa, neste caso apresentando estruturas mais simples se comparados com os estudos de sintagmas para o português. Neste capítulo foram citados alguns trabalhos no inglês mas um maior destaque foi dado aos estudos de indentificação da língua portuguesa, foco desta dissertação. O capítulo também mostrou a descrição de alguns trabalhos relacionados a aplicações que utilizam sintagmas nominais em resolução de anáforas, recuperação de informação, ontologia e da área médica.

3 *Sintagmas Nominais*

"Algumas pessoas acham que foco
significa dizer sim para a coisa em que você irá se focar.
Mas não é nada disso. Significa dizer não às centenas de outras
boas ideias que existem. Você precisa selecionar cuidadosamente."
Steve Jobs, Apple

Neste capítulo são apresentados os conceitos que definem a estrutura da frase, os sintagmas nominais e o método adotado para a representação deles.

3.1 Introdução

São apresentados, neste capítulo, os conceitos da área linguística relacionados à definição dos sintagmas nominais. Para tal, a teoria da gramática da frase na qual estão contidos os sintagmas deve ser abordada, além das definições relacionadas às análises léxica e sintática. Tendo como foco os sintagmas nominais, são apresentadas duas diferentes abordagens de identificação de sintagmas nominais, uma utilizando argumentos de nível sintático e outra de nível semântico. A partir desses conceitos, é descrita a abordagem utilizada como base no desenvolvimento da metodologia do sistema proposto.

3.2 A gramática da frase

A cada uma das unidades da fala que expressam uma ideia, uma emoção, um ordem, um apelo, enfim, um enunciado de sentido completo que estabelece comunicação, chamamos frase [Nicola 2008]. São exemplos de frase:

"Música, maestro!" (3.1)

"As noites estão quentes." (3.2)

"Itaipu diz em nota que causa do apagão não teve origem na usina." (3.3)

Para que a frase tenha significado, é necessário que obedeça a algumas regras gerais da língua e que tenha sentido gramatical completo. Deve apresentar uma gramaticalidade facilmente reconhecível pelos falantes da língua, que têm conhecimento da gramática nesse idioma.

A parte da gramática que estuda todas as relações entre as palavras chama-se sintaxe. As palavras obedecem a uma certa disposição, uma certa ordem para que a frase seja inteligível e, assim, se realize o ato da comunicação. Não só há relação entre as palavras que formam a frase como também entre as orações que formam um período e entre as várias frases que formam um discurso. Quando analisamos uma palavra levando em consideração os aspectos morfológicos e sintáticos simultaneamente, estamos fazendo uma análise morfossintática.

Sempre que a frase, ou um membro da frase, for constituída de um predicado e de um sujeito, ou às vezes só de um predicado, teremos uma oração. Toda oração contém um verbo (ou locução verbal), não podendo haver dois verbos (ou duas locuções verbais) em uma única oração.

"Socorro!" (3.4)

"A menina pediu socorro." (3.5)

"É necessário que você chame o socorro." (3.6)

O Exemplo 3.4 é uma frase, mas não uma oração e, no Exemplo 3.5, tem-se uma frase constituída de uma oração. Por outro lado, o Exemplo 3.6 mostra duas orações, dois verbos, mas apenas uma frase. A oração *"É necessário"* não tem sentido completo, logo, não é uma frase, e sim um membro dela.

Período é definido como uma frase constituída de uma ou mais orações, formando um todo com sentido completo, como no Exemplo 3.6. É chamado de composto quando formado por duas ou mais orações; do contrário, é classificado como simples.

3.2.1 Sintagmas

As frases que compõem qualquer enunciado expressam um conteúdo por meio dos elementos e das combinações desses elementos que a língua proporciona. Dessa maneira, formam-se conjuntos e subconjuntos que funcionam como unidades sintáticas dentro da unidade maior que é a frase.

A essas unidades de sentido e com função sintática chamamos sintagmas [Nicola 2008]. Uma frase pode estar composta de um ou mais sintagmas que, por sua vez, podem ser compostos de um ou mais elementos. Alguns sintagmas podem ser vistos nas frases do Exemplo 3.7, em que o núcleo do sintagma é destacado em negrito.

O modelo básico de oração se estrutura a partir de um sujeito (o ser sobre o qual declaramos algo) e de um predicado (a informação que passamos sobre o sujeito), os chamados termos essenciais da oração. De maneira geral, podemos dizer que temos um nome (substantivo) e um verbo estruturando os enunciados.

*"[A grande **orquestra**] [**tocou** suavemente]."* (3.7)

*"[As grandes **orquestras**] [**tocaram** suavemente]."* (3.8)

A relação entre quem desempenha a função de sujeito e quem organiza o predicado se manifesta tanto no campo semântico (uma relação de significado) como no campo morfológico

(evidenciando concordância por meio das flexões de pessoa e número, por exemplo).

No Exemplo 3.8 os termos *"orquestra tocou"* se relacionam pelo significado, sendo que o mesmo não ocorreria com o verbo *"nevar"*, por exemplo *"orquestra nevou"*. Eles também se relacionam pela concordância: o primeiro conjunto apresenta como elemento central um nome, *"orquestra"* (substantivo no singular), e o segundo, um verbo, *"tocou"* (terceira pessoa do singular). Em consequência, ao alterar o sintagma nominal, o sintagma verbal deverá se adequar como no Exemplo 3.8, em que nome e verbo estão no plural.

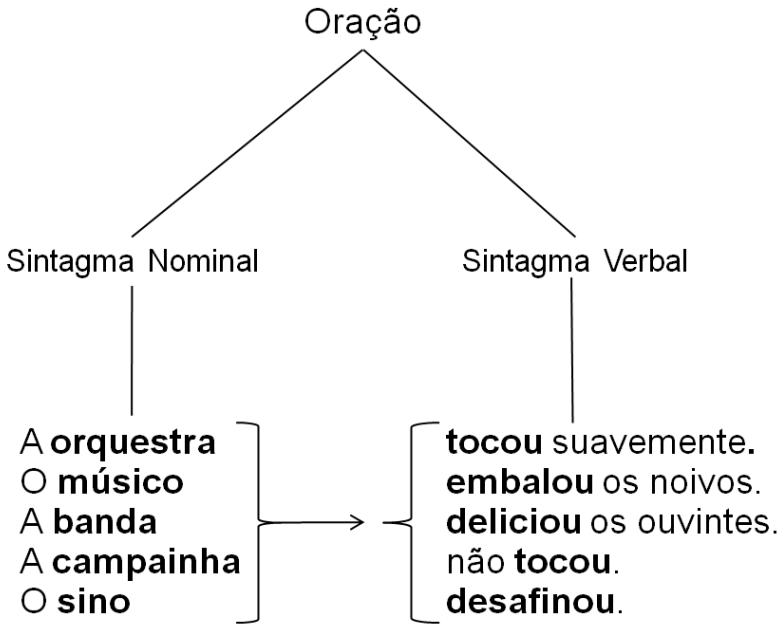


Figura 3.1: Estrutura básica de formação de uma oração

Pode-se, assim, montar uma estrutura básica da frase em língua portuguesa, que permite construir inúmeras orações. Essa possibilidade pode ser vista na Figura 3.1, que mostra vários conjuntos de sintagmas nominais que podem ser substituídos por outros, mantendo a mesma função, e que podem ser combinados com qualquer estrutura de sintagmas verbais, mostradas no outro ramo da árvore, e assim formar frases com sentido completo.

A estrutura básica da oração tem, portanto, um sujeito, um predicado e dois sintagmas distintos, um que se organiza em torno de um nome (sintagma nominal) e outro do verbo (sintagma verbal). Em alguns casos, o sintagma nominal não está explícito na oração mas pode ser identificado pelo contexto. Há casos também em que a oração só contém o sintagma verbal, ou seja, o sintagma nominal com função de sujeito não está presente.

Sintagma Nominal

O Sintagma Nominal (SN) pode exercer a função de sujeito, complemento verbal (objeto), complemento nominal, predicativo, aposto, vocativo, ou seja, qualquer função substantiva. O núcleo de um SN é sempre um nome - ou pronome substantivo ou elemento substanciado - que pode constituir o sintagma sozinho ou aparecer acompanhado de outras palavras, basicamente formando dois grupos:

- os determinantes: termos que se referem ao núcleo para indicar gênero e número (os artigos), localização no tempo e espaço (pronomes demonstrativos), posse (pronomes possessivos) e quantificação (numerais e pronomes indefinidos).
- os modificadores: normalmente representados por adjetivos (sintagmas adjetivais), locuções adjetivas (sintagmas preposicionais), sintagmas nominais preposicionados (quando há transitividade no nome nuclear) e orações adjetivais.

Os determinantes e os modificadores podem estar antes ou depois do núcleo do sintagma nominal para estabelecer com ele relações de concordância. Em consequência, se alterarmos o núcleo do sintagma nominal, os determinantes e os modificadores deverão se adequar. No Exemplo 3.9, tem-se o determinante *"minhas"* antes do núcleo do sintagma e o modificador *"linda"* posicionado depois do núcleo. Enquanto que no Exemplo 3.10, as posições do determinante e do modificador estão invertidas em relação ao exemplo anterior, e, além disso, foram modificadas para o singular e assim, se adequar ao núcleo *"camisa"*.

*"[As minhas **camisas** lindas] foram compradas ontem."* (3.9)

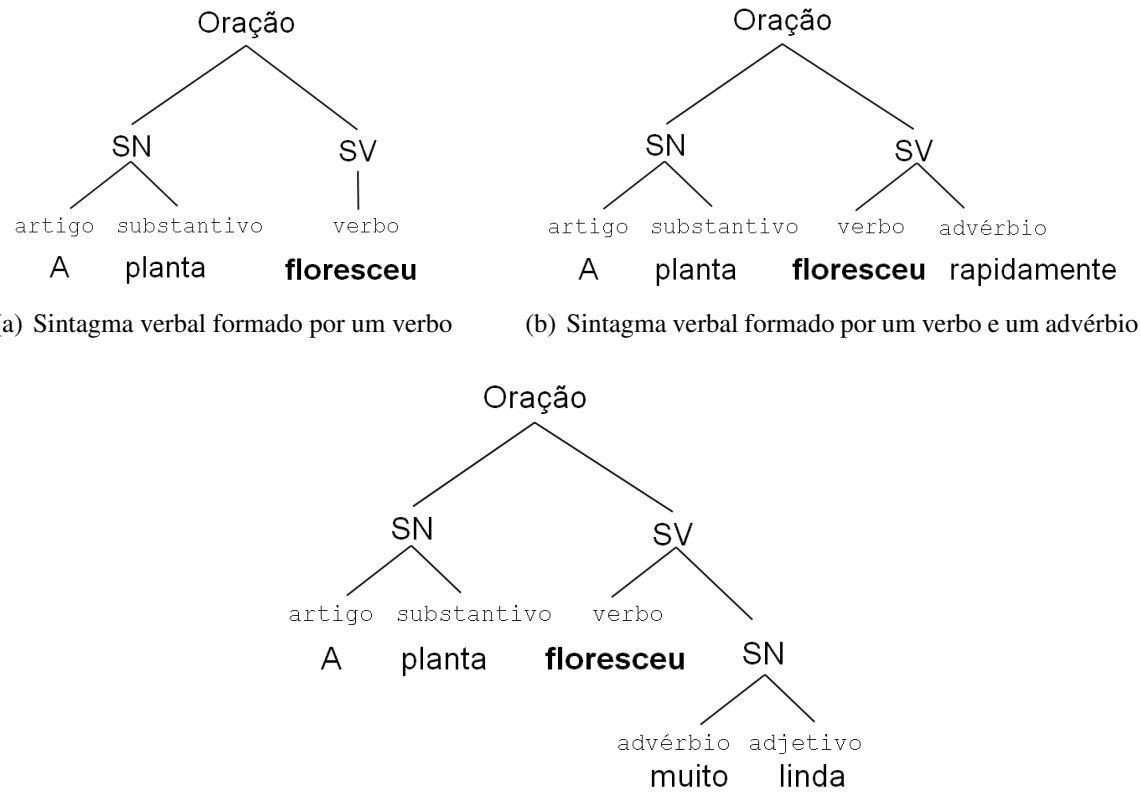
*"[A linda **camisa** minha] foi comprada ontem."* (3.10)

Entre os modificadores do núcleo nominal, o mais comum é o chamado sintagma adjetival (SAdj), isto é, a unidade de sentido que tem valor de adjetivo. O sintagma adjetival pode ser composto de só um elemento (um adjetivo) ou de um adjetivo acompanhado de outras palavras que o modificam. No Exemplo 3.11, há uma frase na qual o sintagma nominal é modificado por um adjetivo (*"brasileira"*) e um sintagma em que o núcleo é um adjetivo (*"contagiante"*). Os elementos que modificam o adjetivo com o intuito de quantificar ou enfatizar, são, geralmente, representados por advérbios, como a palavra *"muito"* do Exemplo 3.11.

*"[A música brasileira] é [muito **contagiante**]."* (3.11)

Sintagma Verbal

O Sintagma Verbal (SV) tem como núcleo um verbo ou uma locução verbal. Se o núcleo do sintagma for um verbo intransitivo, existe a possibilidade de ele ser composto de um único elemento, mas, no caso de verbos transitivos ou de ligação, necessariamente existirão outros sintagmas dentro do sintagma verbal. Alguns exemplos de sintagma verbal podem ser vistos na Figura 3.2.



(a) Sintagma verbal formado por um verbo

(b) Sintagma verbal formado por um verbo e um advérbio

(c) Sintagma verbal formado por um verbo e um sintagma nominal

Figura 3.2: Exemplos de diferentes estruturas de sintagmas verbais

Mesmo com verbos intransitivos, o sintagma pode apresentar, além do núcleo, um advérbio. Os advérbios são modificadores do verbo e formam o sintagma adverbial (SAdv), que pode ser composto de um ou mais elementos, sendo um intensificador mais um advérbio ou uma locução adverbial.

Sintagmas Preposicionados

Os sintagmas preposicionados são unidades de sentido que vêm introduzidas por preposição. Essas preposições desempenham um relevante papel na organização dos enunciados, já que relacionam dois sintagmas, tornando o sintagma introduzido por preposição dependente do outro, como ocorre nos casos em que um termo regente (verbo, substantivo, adjetivo, advérbio) exige um complemento. Além disso, as preposições introduzem sintagmas representados por locuções (adjetivas ou adverbiais).

*"Os músicos [da **França**], amantes [do **Jazz**], gostaram [da **versão brasileira**] [com **rapidez**]."* (3.12)

O Exemplo 3.12 destaca sintagmas nominais preposicionados que têm diferentes funções dentro da frase. O termo *"da França"* é um sintagma preposicional com valor adjetival exercendo a função de adjunto adnominal, *"do Jazz"*, de complemento nominal, *"da versão brasileira"*, de complemento indireto, e o sintagma preposicional com valor adverbial *"com rapidez"* exerce a função de adjunto adverbial.

3.3 Análise gramatical

A linguística teórica argumenta que os seres humanos possuem tipos diferentes de conhecimento linguístico, organizados em diferentes módulos: por exemplo, os conhecimentos da estrutura dos sons de uma língua (fonologia), da estrutura das palavras (morfologia), da estrutura das frases (sintaxe) e do significado (semântica). Cada tipo de conhecimento linguístico é tornado explícito como um módulo diferente da teoria, consistindo de uma coleção de elementos de base juntamente com uma forma de combiná-los em estruturas complexas. Computacionalmente tem-se programas que tratam da tarefa de análise de cada um desses módulos.

O analisador morfológico ou léxico identifica palavras, ou expressões isoladas, em um texto por meio de pontuação e espaços em branco, são os chamados *tokens*. Ele também é responsável pela classificação das palavras de acordo com a categoria gramatical em relação ao texto. A análise léxica/morfológica tem importância pois, para se formar uma estrutura coerente de uma frase, é necessário compreender o significado de cada uma das palavras que a compõe [Rich e Knight 1993].

Por meio das informações obtidas do analisador morfológico e tendo definida a gramática

da linguagem, o analisador sintático verifica se uma sentença formada é uma sentença válida da linguagem. Ou seja, se as palavras relacionadas entre si geram uma árvore de derivação a partir das regras da linguagem. As gramáticas utilizadas por um analisador sintático podem ser classificadas como regulares, livres de contexto ou sensíveis ao contexto.

O analisador semântico avalia o sentido das estruturas das palavras que foram reagrupadas pelo analisador sintático, uma vez que o analisador morfológico permitiu identificar essas palavras individualmente. Em suma, denomina-se análise semântica o processo de criar uma ou mais representações semânticas às árvores de derivação sintáticas.

3.3.1 Análise léxica

As classes de palavras têm sido definidas segundo suas propriedades semânticas, sintáticas e morfológicas, e representam seres, ações e itens gramaticais responsáveis por estruturar as frases. Na Língua Portuguesa existem dez classes gramaticais de palavras. Qualquer palavra usada estará inserida em uma das classes, [Bechara 2009], enumeradas a seguir.

1. Substantivo: palavra que nomeia os seres (visíveis ou não, animados ou não) e também nomeia estados, desejos, sentimentos e ideias dos seres. Podem ser classificados em comum ou próprio, abstrato ou concreto, simples ou composto e primitivo ou derivado;

2. Adjetivo: palavra que caracteriza os seres. Refere-se sempre a um substantivo explícito ou implícito na frase, com o qual concorda em gênero, número e grau;

3. Numeral: palavra que expressa a quantidade de pessoas ou seres, e também o lugar que elas ocupam em uma determinada sequência. São classificados em cardinais, ordinais, multiplicativos e fracionários;

4. Artigo: palavra que antecede o substantivo, indicando-lhe gênero, número e quantidade, ao mesmo tempo em que determina, de forma precisa ou vaga, os tipos definido e indefinido;

5. Advérbio: palavra ligada ao verbo, ao adjetivo ou a outro advérbio (formando uma locução adverbial), com a função de modificá-lo, geralmente atribuindo uma circunstância ou característica relativas a lugar, tempo, intensidade, afirmação etc.;

6. Pronome: palavra que substitui, indica ou acompanha um substantivo, como pessoa do discurso. Pode ser do tipo pessoal, demonstrativo, indefinido, entre outros;

7. Preposição: palavra invariável que liga termos de uma oração, estabelecendo entre elas diversas relações. Algumas preposições contraem-se com o artigo ou com o pronome demonstrativo (“do” = “de” + “o”, “neste” = “em” + “este”) ou pode acontecer de duas ou mais palavras formarem uma locução prepositiva (“a fim de”, “através de”, “junto a”);

8. Conjunção: palavra invariável usada para unir orações ou termos semelhantes de uma oração estabelecendo entre eles uma relação de dependência ou de simples coordenação;

9. Interjeição: palavra invariável usada para exprimir emoções e sentimentos;

10. Verbo: palavra que costuma indicar uma ação, um estado ou um fenômeno da natureza. Flexiona-se em número, pessoa, tempo, modo e voz, podendo se apresentar como regular, irregular ou de ligação.

Algumas classes gramaticais descritas anteriormente têm subclassificações, como os substantivos que podem ser categorizados em abstrato e concreto, ou os advérbios que podem ser relativos a lugar ou a tempo. Esse nível de detalhamento, para algumas dessas classes, não é de interesse do presente trabalho. É de importância para tal as variações de gênero, número e grau dos substantivos e adjetivos, que são detalhados a seguir.

Flexão de gênero, número e grau

A flexão dos substantivos e adjetivos possui três naturezas: gênero, número e grau. De uma forma geral, eles são flexionados para acompanhar e concordar entre si, com adjetivos e substantivos e também para intensificar ou comparar o seu valor. Variações de gênero são aquelas utilizadas para fazer a distinção entre os sexos – masculino ou feminino – (por exemplo: *“irmão”*, *“irmã”*). Flexão de número é utilizada para distinguir palavras no singular ou no plural (*“carro”*, *“lindos”*) e o grau é utilizado, no substantivo, para indicar o tamanho do ser que se nomeia (*“casinha”*, *“mulherona”*, *“maleta”*), podendo ser aumentativo ou diminutivo.

“Maria está muito feliz.” (3.13)

“Maria está felicíssima.” (3.14)

A variação de grau nos adjetivos ocorre quando se deseja fazer uma comparação ou intensificar o seu valor, classificados como comparativo e superlativo, respectivamente. No Exemplo 3.13 tem-se um caso de flexão comparativa, e no Exemplo 3.14, de flexão superlativa.

3.3.2 Análise sintática

A análise sintática determina a sintaxe, ou estrutura, de uma linguagem e é dada a partir das regras de uma gramática, [Louden]. Uma gramática é um mecanismo gerador que permite definir formalmente uma linguagem, a qual é muito utilizada na especificação de linguagens computacionais. Através de uma gramática pode-se gerar todas as sentenças da linguagem definida por ela e é determinada de maneira formal como $G = (V, T, P, S)$, em que:

- V é um conjunto finito de símbolos não-terminais (ou variáveis);
- T é um conjunto finito de símbolos terminais disjunto de V;
- P é um conjunto finito de regras de produção e
- S é um elemento de V, denominado símbolo inicial (ou símbolo de partida).

Os símbolos de T equivalem ao alfabeto em cima do qual a linguagem é definida. Os elementos de V são símbolos auxiliares criados para permitir a definição das regras da linguagem e correspondem às categorias sintáticas da linguagem definida. As regras de produção definem as condições de geração das sentenças. A aplicação de uma regra de produção é denominada derivação. O símbolo inicial é aquele através do qual o processo de derivação de uma sentença deve ser iniciado.

Noam Chomsky, em [Chomsky 1956], classificou as gramáticas formais segundo a chamada hierarquia de Chomsky. Essa classificação possui quatro níveis, começando com maior grau de liberdade em suas regras e aumentando as restrições até o último nível. Segundo a hierarquia de Chomsky, as gramáticas podem ser classificadas pelos seguintes tipos:

- Gramáticas com estrutura de frase (tipo 0): são aquelas às quais nenhuma limitação é imposta. São as linguagens que podem ser definidas através dos mecanismos gerativos obtidos pela gramática e podem ser reconhecidas pela máquina de Turing;
- Gramáticas sensíveis ao contexto (tipo 1): produzem gramáticas sensíveis ao contexto, entretanto, surge o problema da complexidade computacional do algoritmo de resolução da gramática que é exponencial sobre o tamanho da sentença;
- Gramáticas livres de contexto (tipo 2): essas gramáticas não levam em consideração o contexto em que estão inseridas e permitem representar linguagens com grau de complexidade maior que as regulares. Porém, apresentam problemas para expressar dependências, por exemplo, ao se trabalhar com concordância verbal;

- Gramáticas regulares (tipo 3): geram linguagens regulares que são reconhecidas com facilidade por ter uma restrição nas regras de produção da gramática. Por possuírem propriedades adequadas para a obtenção de reconhecedores simples, denominados de expressão regular, essas gramáticas são de grande importância no estudo dos compiladores. Todavia, por apresentarem baixo poder de expressão, o uso em linguagem natural é limitado.

Cada classe sucessiva, na hierarquia de Chomsky, pode gerar um conjunto mais amplo de linguagens formais que a classe imediatamente anterior. Chomsky argumenta que a modelagem de alguns aspectos da linguagem humana necessita de uma gramática formal mais complexa que a modelagem de outros aspectos. Enquanto que uma linguagem regular é suficientemente poderosa para modelar a morfologia da língua inglesa, por exemplo, ela não é suficientemente poderosa para modelar a sintaxe da mesma.

O analisador sintático para uma gramática é um programa que aceita como entrada uma sentença e constrói por ela sua árvore gramatical (ou equivalentemente uma sequência de derivação) ou, caso a sentença não pertença à linguagem descrita, uma indicação de erro é gerada, [Ricarte 2008]. Os analisadores sintáticos se dividem nos seguintes tipos:

- Analisadores *top-down*: fazem a análise sintática da frase por meio de uma construção descendente, começando com uma lista que contém a princípio apenas o símbolo inicial da linguagem. A seguir, a partir da análise dos símbolos presentes na sentença, busca aplicar regras que permitam expandir os símbolos na lista até alcançar a sentença desejada.

  Na construção descendente, o objetivo é obter uma derivação mais à esquerda para uma sentença. Em termos de árvores gramaticais, o analisador *top-down* busca a construção de uma árvore a partir da raiz usando varredura em pré-ordem para definir o próximo símbolo não-terminal, que deve ser considerado para análise e expansão.

- Analisadores *bottom-up*: na construção ascendente, o analisador sintático varre a sentença buscando aplicar produções que permitam substituir seqüências de símbolos da sentença pelo lado esquerdo das produções, até alcançar como único símbolo restante o símbolo inicial da linguagem.

Uma gramática que gera uma cadeia com duas árvores de análise sintática distintas é denominada gramática ambígua. Uma gramática com essa característica representa um problema sério para o analisador sintático, pois ela não especifica com precisão a estrutura sintática de um programa. Essa ambiguidade não pode ser removida com facilidade.

3.4 Os Sintagmas Nominais

Os sintagmas nominais podem ser definidos por meio de estruturas sintáticas assim como por funções semânticas. A abordagem de definição de sintagmas nominais de Perini [Perini 2003] é voltada a uma análise sintática, e a de Liberato [Liberato 1997] mostra uma visão mais semântica. No trabalho de Miorelli [Miorelli 2001] e Souza [Souza 2005] encontra-se um comparativo entre essas duas abordagens e, assim como eles, buscou-se, a partir desses trabalhos, definir uma estrutura para determinar os sintagmas nominais na implementação do SISNOP – Sistema Identificador de Sintagmas Nominais do Português.

Perini [Perini 2003], em sua abordagem, considera que as unidades linguísticas apresentam dois aspectos fundamentais: a forma (o significante) e o significado. Assim, a descrição de uma língua é composta de uma descrição formal (fonologia, morfologia e sintaxe), uma descrição semântica e um sistema que relaciona o plano semântico ao plano formal.

O sintagma nominal tem uma estrutura bastante complexa, incluindo termos de comportamento sintático diversos, sendo possível distinguir, dentro do SN, diversas funções sintáticas. Perini descreve que os sintagmas nominais possuem uma estrutura composta por termos internos do SN, definidos por meio de traços de natureza posicional.

As funções no SN, por sua vez, definem-se pelas posições dos termos em relação uns aos outros, e não por suas posições absolutas. Um sintagma no qual todas as posições possíveis forem preenchidas por itens léxicos denomina-se SN máximo. Ele é, na verdade, uma abstração, pois na prática seria muito longo. Para entender a estrutura do sintagma nominal, pode-se dividi-lo em área à esquerda e à direita, ao redor do núcleo [Perini 2003].

Examinando um grande número de sintagmas, Perini em seu trabalho chegou à conclusão de que a área à esquerda compreende seis posições fixas e quatro variáveis. As posições fixas são definidas como (na ordem em que ocorrem no sintagma): determinante (Det), possessivo (Pos), reforço (Ref), quantificador (Qtf), pré-núcleo externo (Pne) e pré-núcleo interno (Pni). As posições variáveis ocorrem nos intervalos entre as posições fixas, exceto entre os dois pré-núcleos, e podem ser definidas como a função de numerador (Num).

Det Num Pos Num Ref Num Qtf Num Pne Pni Núcleo ModI ModE (3.15)

A área à direita no SN, por sua vez, compreende o núcleo e os termos que ocorrem depois dele, e apresenta três funções: núcleo, modificador interno (ModI) e modificador externo

(ModE). O modificador externo é aquele que pode ser separável do resto do SN por um sinal de pontuação. O núcleo caracteriza-se por constituir, sozinho, um sintagma, e o modificador interno é aquele que não pode ser separado por pontuação. O Exemplo 3.15 mostra o esquema completo de definição do sintagma nominal, segundo Perini [Perini 2003].

Enquanto Perini define uma estrutura sintática dos constituintes do sintagma nominal, Liberato [Liberato 1997] foca mais nas funções semânticas e expõe que os SNs possuem uma função referencial, isto é, referem-se a um objeto do mundo exterior como, por exemplo, uma entidade, identificada através de uma palavra ou expressão.

As funções semânticas dos componentes de um sintagma nominal definidas por Liberato são: classificador (CLA), subclassificador (SUB), qualificador (QUAL), recortador (REC) e quantificador (QUAN). Essas funções determinam como os objetos são classificados no mundo, ou seja, tipos de características de acordo com a classe gramatical. Nas frases listadas a seguir há exemplos de frases com sintagmas de cada uma das funções descritas.

"[Um exercício] vai resolver o seu problema." – CLA (3.16)

"[Um exercício aeróbico] vai resolver o seu problema." – SUB (3.17)

"[A agenda lotada] de hoje." – QUAL (3.18)

"[As poucas horas] que restam." – QUAN (3.19)

"[Os] telhados da capela." – RECPAR (3.20)

"[Uns] telhados da capela." – RECUNI (3.21)

A função classificador (CLA) delimita a classe mais ampla em que o referente é enquadrado a partir de uma determinada descrição. No Exemplo 3.16, *"um exercício"* refere-se a uma entidade ou papel e pode ser enquadrado em uma classe de entidades denominada *"exercício"*. Enquanto no Exemplo 3.17, delimita-se uma subclasse *"exercício aeróbico"* dentro de uma classe mais ampla que é *"exercício"*, exercendo a função de subclassificador (SUB).

O elemento qualificador (QUAL) fornece características descritivas, possuindo função explicativa, como no Exemplo 3.18, em que *"a agenda lotada"* é classificada como tal. Em contrapartida, o elemento quantificador (QUAN) indica a quantidade de elementos que constitui o referente, como no Exemplo 3.19.

Os recortadores podem ser divididos em duas categorias: os de recorte parcial (RECPAR) e o universal (RECUNI), e são diferenciados pelos artigos definido e indefinido. Eles indicam

se o referente é constituído da totalidade ou de parte dos elementos da menor classe delimitada qualitativamente, como nos Exemplos 3.20 e 3.21, respectivamente.

Miorelli [Miorelli 2001] faz uma comparação das funções definidas por Perini [Perini 1995] e por Liberato [Liberato 1997], que pode ser vista na Tabela 3.1. Vale destacar que as funções CLA e Núcleo são idênticas em ambas as abordagens, quanto a sua ocorrência e também quanto à função dentro do SN.

Liberato	**Perini**	**Exemplos**
CLA	Núcleo	*"o médico pediatra"*
SUB	Poss	*"minha agenda lotada"*
SUB	ModE	*"os telhados de Viena"*
SUB	Ref	*"certos dias passam rápido demais"*
QUAN	Qtf	*"várias pessoas passaram mal"*
QUAN	Num	*"os dois homens"*
QUAL	Pne	*"o velho rapaz"*
QUAL	Ref	*"um certo assunto"*
QUAL	ModE	*"Vozes dÁfrica, que é um poemeto épico"* *"representa um momento da poesia brasileira."*
RECUNI	Det	*"os pecadores que se arrependem"*
RECPAR	Det	*"um velho amigo"*
RECPAR	Qtf	*"dois homens seguiram"*

Tabela 3.1: Análise comparativa entre Liberato e Perini, feita por Miorelli [Miorelli 2001]

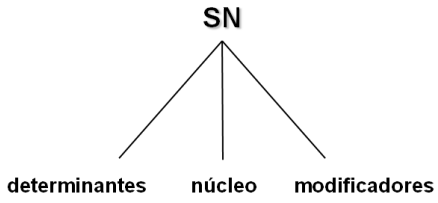
A análise semântica proposta por Liberato [Liberato 1997] discute em profundidade aspectos isolados da composição do SN sem uma preocupação com a sua estrutura geral, importando-se somente com os enunciados das sentenças, e com a relação entre os enunciados e seu significado comunicativo. Por outro lado, Perini [Perini 2003] se preocupa com as palavras e com a sequência delas para obter a forma do SN. Pode-se distinguir os dois trabalhos como sendo, respectivamente, um de avaliação sintática e um de avaliação semântico-pragmática.

Em uma análise semântica, por exemplo, ao se falar da cabeça, sabe-se que ela pertence a um corpo, que possui outras partes do tipo olhos, boca, nariz etc. Os termos *"o filho do Guto"* e *"meu sobrinho"* têm significados gramaticais diferentes, mas podem referenciar o mesmo ser. Portanto, a análise leva em conta traços não só do significado gramatical das expressões, como também do seu contexto situacional e dos locutores. Essa é uma das principais dificuldades de se tratar computacionalmente o significado das palavras propriamente dito e o sentido delas nas sentenças. Por vezes, também a estrutura da sentença não reflete o significado. Por exemplo, o sintagma *"a captura"* pode ter significado diferente, quando se fala de beisebol ou de perseguição policial, porém a estrutura sintática é a mesma; os significados diferentes surgem

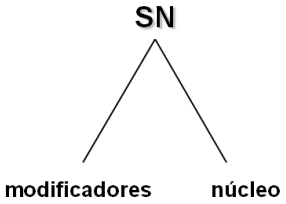
da inserção da palavra em cada contexto. Entender o significado de uma sentença depende da situação na qual a sentença é produzida.

Miorelli [Miorelli 2001] e Souza [Souza 2005] escolheram utilizar em seus trabalhos as funções e posições do SN definidas por Perini [Perini 2003], por parecer mais prático tratar o comportamento formal dos constituintes no SN e a relação entre eles, do que tratar a função semântica de cada um em relação ao núcleo do SN. O presente trabalho também utilizou como base o método de Perini.

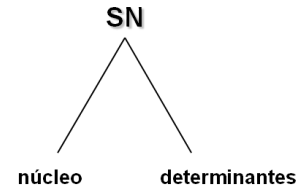
Assim como Perini, Azeredo [Azeredo 1990] tem uma abordagem mais sintática e define que um sintagma nominal comporta necessariamente um núcleo, que, sendo um substantivo comum, pode ser acompanhado por determinantes e modificadores. A Figura 3.3 mostra a estrutura básica de um sintagma nominal e como os determinantes e modificadores variam de posição em torno do núcleo.



(a) Estrutura padrão



(b) Modificadores antes do núcleo



(c) Determinantes após o núcleo

Figura 3.3: Estruturas de formação de um Sintagma Nominal

As estruturas vistas na Figura 3.3 são utilizadas na composição das regras gramaticais implementadas pelo sistema descrito neste trabalho. A seguir são feitas considerações e restrições que foram consideradas na definição da estrutura e das regras:

- o núcleo do sintagma será formado por um nome, um nome composto (*"Universidade Federal"*), um adjetivo ou um pronome pessoal. Pode também incluir sintagmas preposicionais (*"Universidade Federal do Espírito Santo"*);
- os sintagmas preposicionais, os adjetivais (SAdj) e os adverbiais (SAdv) serão tratados de maneira implícita dentro das regras de sintagmas nominais, não sendo diferenciados e

nem extraídos dos textos. Essa medida é tomada por considerar esse nível de detalhismo sem interesse para o presente trabalho;

- os números ordinais serão considerados como pré-modificadores (*"sexto grau"*), e os cardinais como pós-modificadores (*"grau seis"*). Os demais numerais (multiplicativos e fracionários) serão tratados como substantivo. As datas (*"13 de dezembro de 2009"*, *"13/12/09"*) também serão classificadas gramaticalmente como substantivos;

- os verbos no tempo particípio (como em *"a mesa quebrada"*), por serem uma forma nominal do verbo utilizado (por vezes, para caracterizar substantivos), serão considerado na classe gramatical adjetivo;

- não será recuperado o sintagma em que o núcleo é a figura de linguagem elipse – quando ocorre a omissão de termos que estão subentendidos na frase (ex.: *"os quatro vieram"*).

Assim, com uma estrutura definida e com as considerações feitas, a metodologia para o desenvolvimento de um sistema de identificação automática de sintagmas nominais pode ser determinada.

3.5 Considerações Finais

A gramática descritiva da língua portuguesa define uma frase como sendo formada por conjuntos de elementos unidos pelo sentido e função sintática, os sintagmas. Eles podem ser dos tipos verbal ou nominal, dependendo do seu núcleo. O interesse é maior nos sintagmas nominais, já que eles possuem sentido de substantivo podendo realizar funções temáticas e semânticas, além do seu alto poder discriminatório dentro do texto.

Levando em consideração os estudos linguísticos de Liberato [Liberato 1997] e Perini [Perini 2003], é possível observar que o sintagma nominal possui forma sintática, tratada por Perini, e semântica, tratada por Liberato. Optou-se por utilizar a abordagem estudada por Perini, pelo fato de, computacionalmente, existir uma facilidade maior em se trabalhar com a forma (sintático) do que o significado (semântico). Assim, levando em consideração também o estudo da teoria linguística da gramática de frases, foi possível definir a estrutura de formação dos sintagmas nominais, a ser utilizada na construção das regras gramaticais do sistema de identificação de sintagmas nominais descrito no capítulo seguinte.

4 O Algoritmo

"*Velocidade é muito mais importante do que aparência. Um carro bonito não é tão impressionante quando é ultrapassado por um carro antigo, isso também pode ser aplicado a software.*"

Sergey Brin, co-fundador da Google

Neste capítulo é detalhada a metodologia e são apresentados os algoritmos desenvolvidos para a implementação do sistema que identifica sintagmas nominais em textos.

4.1 Introdução

Este capítulo apresenta os algoritmos desenvolvidos visando a identificar os sintagmas nominais, previamente definidos no Capítulo 3. É explicitada a metodologia adotada além de soluções para problemas inerentes ao objetivo proposto através da implementação do SISNOP – Sistema Identificador de Sintagmas Nominais do Português. O SISNOP é um sistema capaz de interpretar textos irrestritos disponíveis em linguagem natural, desenvolvido com o propósito de identificar e extrair sintagmas nominais.

4.2 SISNOP – Sistema Identificador de Sintagmas Nominais do Português

O SISNOP – Sistema Identificador de Sintagmas Nominais do Português – é um sistema composto por um conjunto de módulos e programas, capazes de interpretar textos da língua portuguesa através das análises morfológica e sintática e, assim, extrair os sintagmas nominais de cada frase contida nos textos. A Figura 4.1 mostra esquematicamente quais são esses módulos e como estão organizados na linha de execução do sistema.

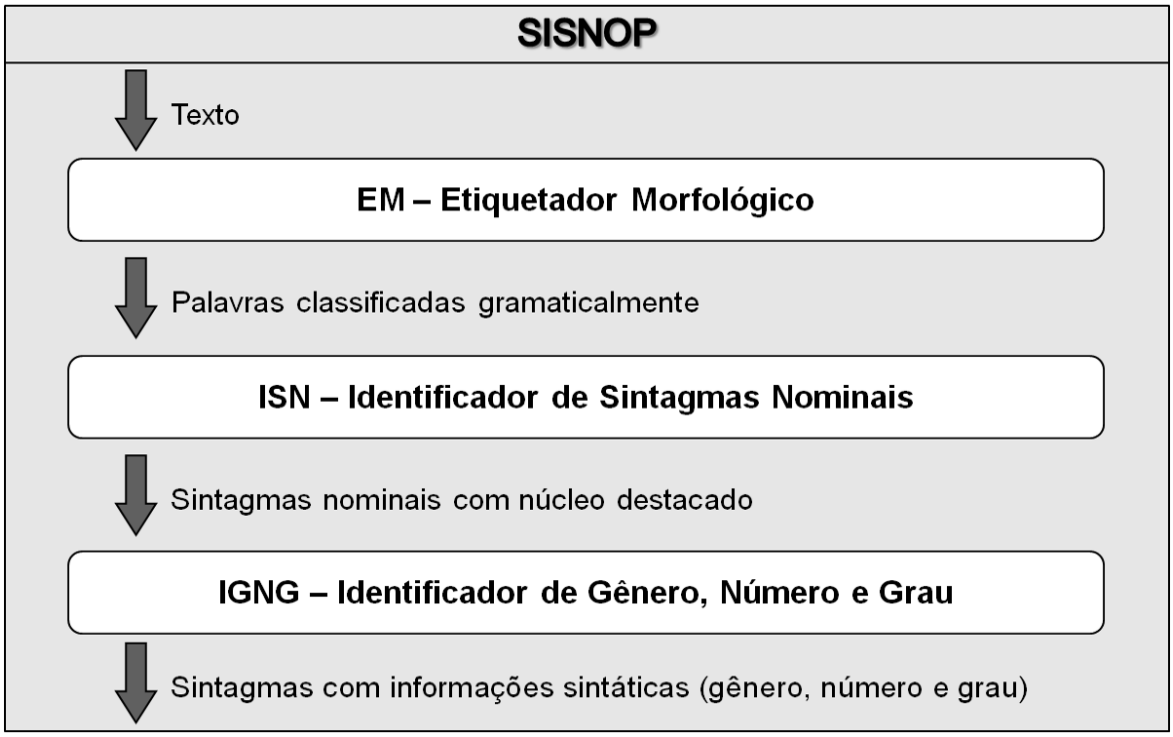


Figura 4.1: Estrutura em módulos do sistema SISNOP

Primeiramente, o texto é fornecido ao módulo chamado EM (Etiquetador Morfológico), que

é responsável pela análise léxica/morfológica, que encontra para cada palavra uma etiqueta que a classifique. A saída do EM é informada, então, ao ISN (Identificador de Sintagmas Nominais) que faz a identificação dos sintagmas nominais de cada frase do texto. Para cada um desses sintagmas, o IGNG (Identificador de Gênero, Número e Grau) faz o reconhecimento de gênero, número e grau das palavras que os compõem.

O Algoritmo 1 descreve a interação dos módulos do SISNOP, que recebe como parâmetro de entrada um texto em formato simples (.txt), sem marcações, e fornece como retorno uma lista de sintagmas nominais encontrada neste texto. Além de gênero, número e grau informados, a lista de saída apresenta destacado o núcleo de cada sintagma.

Algoritmo 1

function SISNOP (T: Texto, Gramatica :Lista) Lista
Pré-condição: Arquivo texto
Pós-condição: Lista de sintagmas nominais

```
1: conjFraseEtiq ⇐ EM(T)
2: for all frase in conjFraseEtiq do
3:     snSemGng ⇐ ISN(frase, Gramatica)
4:     SN ⇐ SN ++ IGNG(snSemGng)
5: end for
6: return SN
```

Fornecendo o texto do Exemplo 4.1 ao Algoritmo 1, o primeiro passo é fazer a identificação das frases e a etiquetagem morfológica das palavras nelas contidas por meio do módulo EM. Assim, tem-se um conjunto de frases nas quais cada palavra, inclusive as pontuações, está classificada de acordo com sua categoria gramatical (substantivos, adjetivos, verbos etc). Após essa etapa, as palavras do Exemplo 4.1 estão etiquetadas como mostrado na Tabela 4.1.

"A menininha chegou. Sua mãe está feliz." (4.1)

*"[A **menininha**] chegou."* (4.2)

*"[Sua **mãe**] está [**feliz**]."* (4.3)

Essas palavras classificadas são enviadas ao ISN para que os sintagmas nominais sejam obtidos. Os Exemplos 4.2 e 4.3 destacam os sintagmas nominais identificados pelo módulo ISN. Os sintagmas são, então, fornecidos para o IGNG, que identifica as flexões das palavras como no sintagma nominal do Exemplo 4.2: *"A menininha"*, em que as palavras estão no singular, no feminino e no diminutivo.

Nas próximas seções será detalhado o funcionamento de cada um dos módulos, EM, ISN e

A	artigo	**Sua**	pronome demonstrativo
menininha	substantivo	**mãe**	substantivo
chegou	verbo	**está**	verbo
.	pontuação	**feliz**	adjetivo
		.	pontuação

Tabela 4.1: Classificação morfológica das palavras do Exemplo 4.1

IGNG, que compõem o SISNOP. Serão explicitados os algoritmos desenvolvidos, os programas utilizados e as metodologias escolhidas na implementação do sistema.

4.3 EM – Etiquetador Morfológico

O módulo EM – Etiquetador Morfológico – é a primeira etapa do SISNOP, sendo responsável pela análise léxica/morfológica. A etiquetagem morfológica, conhecida em inglês como *Part-Of-Speech tagging* (*POS-tagging*), é o processo de encontrar uma etiqueta, marcar com um *tag* cada uma das palavras de um texto baseado em sua definição, assim como em seu contexto. Ou seja, é o método de encontrar a categoria gramatical que mais se adéqua à palavra, levando em consideração sua natureza e seu relacionamento com as palavras adjacentes a ela na frase. De forma simplificada, pode-se dizer que é o trabalho de classificar palavras como substantivos, verbos, adjetivos etc.

Realizar a atribuição da categoria gramatical de uma palavra é mais difícil do que somente fazer o relacionamento entre uma lista de palavras e uma lista de entidades, pois algumas palavras podem representar mais de uma categoria em diferentes ocasiões. Essa ambiguidade é muito comum em linguagem natural, diferente de muitas linguagens artificiais, como as linguagens de programação.

O sistema FORMA[1] é o *software* utilizado pelo módulo EM para realizar a tarefa de delimitar as palavras (também chamadas de *tokens*) do texto e classificá-las gramaticalmente. O FORMA, que tem código fonte aberto disponível para utilização, faz parte das ferramentas do Grupo de Processamento de Linguagem Natural da PUC - Rio Grande do Sul[2]. O FORMA foi utilizado em recentes trabalhos como [Moura et al. 2008], [Moraes e Lima 2008] e [Moraes e Lima 2007], e apresenta precisão em torno de 95% [Gonzalez, Lima e Lima 2006].

O FORMA é um analisador léxico, ou seja um *POS-tagger*, que atribui etiquetas morfológicas para palavras e sinais de pontuação, além de encontrar o lema de cada palavra. O lema de

[1] http://www.inf.pucrs.br/ gonzalez/tr+/forma/
[2] http://www.inf.pucrs.br/ linatural/projetos.html

um verbo é sua forma infinitiva e o de uma palavra variável é sua forma singular e masculina (quando existente). O EM não utiliza o lema fornecido pelo FORMA, somente as categorias gramaticais.

O programa FORMA é composto por uma ferramenta probabilística e por uma base de dados auxiliar que foi criada a partir de um corpus de treino de palavras etiquetadas e com lemas definidos. A base de dados auxiliar é utilizada na construção de três autômatos finitos: um para composição, um para palavras acentuadas e um para sufixos. Além disso, utiliza duas matrizes probabilísticas para auxiliar a identificação da categoria gramatical das palavras.

Os autômatos finitos [Hopcroft Rajeev Motwani 2002] buscam encontrar a maior parte (*substring*) de uma determinada palavra (*string*) de entrada. Em geral, funcionam da seguinte maneira: dado um autômato **A** e uma *string* **t** de entrada, a leitura de **t** por **A** termina se **t** for totalmente "consumido" ou se existir uma transição inválida para o próximo caractere de **t**. Então, a *substring* **st** de **t** lida é aceita se o estado corrente de **A** é um estado final, do contrário **t** é rejeitado. Uma *substring* **st** aceita pode ser a raiz, em um autômato chamado esquerda-direita, ou o sufixo de **t**, em um autômato chamado direita-esquerda.

O autômato para composição (com 577 termos) detecta unidades formadas por multipalavras: composições de preposições (ex.: "por meio de") e de advérbios (ex.: "de repente"). O autômato de acentos (com 302 termos) tem a tarefa de tratar casos de palavras de igual escrita, cuja diferenciação da classe gramatical se dá através do acento (ex.: "pêlo"/'pelo"). Contudo, pelas novas regras gramaticais da língua portuguesa, que serão obrigatórias a partir de janeiro de 2013, não existirá mais essa diferenciação. Por fim, o maior deles, com mais de 43 mil palavras, é o autômato de sufixos que analisa o sufixo das palavras para determinação do lema e da probabilidade da categoria.

Enquanto os três autômatos obtêm os lemas e as probabilidade das etiquetas, as duas matrizes, chamadas BP (*before probability* – probabilidade anterior) e AP (*after probability* – probabilidade posterior), estimam a probabilidade da ocorrência da classe gramatical em relação ao texto. A matriz BP é definida como $BP = [bp_{mj}]$, onde $bp_{mj} = Pr(m|j)$, em que cada elemento armazena a probabilidade de ocorrência da classe j dada a ocorrência anterior da classe m no corpus. Por outro lado, $AP = [ap_{nj}]$, onde $ap_{nj} = Pr(j|n)$ é a probabilidade de ocorrência da classe j dada a ocorrência posterior da classe n no corpus.

O Algoritmo 2 explicita o funcionamento do etiquetador morfológico implementado pela ferramenta FORMA que, sequencialmente, realiza as seguintes tarefas:

Algoritmo 2

function EM (T :Texto): Lista

1: $conjToken \Leftarrow definicaoToken(T)$
2: **for all** $token$ **in** $conjToken$ **do**
3: $(token, lema, probEtiqueta) \Leftarrow calculaLemaEtiqueta(token)$
4: **if** $(prob\_etiqueta == 100\%)$ **then**
5: $conjPalavraEtiq \Leftarrow conjPalavraEtiq ++ (token, lema, probEtiqueta)$
6: **else**
7: $(token, lema, listProb) \Leftarrow automatoComposicao(token)$
8: $(token, lema, listProb) \Leftarrow automatoAcentos(token)$
9: $(token, lema, listProb) \Leftarrow automatoSufixos(token)$
10: $(token, lema, listProb) \Leftarrow probabilidades(token, lema, listProb)$
11: $(token, lema, listProb) \Leftarrow ambiguidades(token, lema, listProb)$
12: $conjPalavraEtiq \Leftarrow conjPalavraEtiq ++ (token, lema, MAX(listProb)$
13: **end if**
14: **end for**
15: **return** $conjPalavraEtiq$

- Definição dos *tokens*: palavras e sinais de pontuações são identificados, usando como separador o caractere espaço.

- Definição dos lemas e das etiquetas para cada um dos *tokens*: essa etapa produz duas possíveis saídas. A primeira delas retorna o lema e a etiqueta final (100% probabilidade). Caso não seja possível a primeira opção, é fornecida como saída uma sequência de possíveis etiquetas, com seus respectivos lemas e probabilidades. Então o processo começa pelo autômato de composição, que lê sequências de até quatro *tokens* que serão aceitos como entrada válida quando a maior composição for derivada como um novo *token*. O autômato para acentos lê somente *tokens* não etiquetados e verifica a validade destes. O autômato para sufixo lê também palavras não etiquetadas, e para cada entrada aceita, produz um conjunto de possíveis etiquetas e suas probabilidades.

- Tratamento de probabilidades: para o *token* não etiquetado t_i, são analisados os *tokens* adjacentes t_{i-1} e t_{i+1} e, após efetuar o cálculo, a etiqueta de maior probabilidade e o lema relativo são aplicados a t_i.

- Tratamento de ambiguidade: palavras ambíguas (“casa”, verbo; “casa”, substantivo) são corretamente etiquetadas por meio de verificação dos *tokens* próximos.

As categorias gramaticais da língua portuguesa identificadas pelo FORMA são: artigos (definidos e indefinidos), adjetivos, advérbios, substantivos, numerais (cardinais e ordinais), verbos (no particípio, auxiliares, regulares e irregulares), pronomes (pessoais, possessivos,

demonstrativos, indefinidos e relativos), preposições, conjunções (coordenativas e subordinativas), interjeições e pontuações (ponto final, exclamação, interrogação, vírgula, parênteses e traço).

O FORMA não reconhece de maneira correta elementos específicos, como URL ("www.ufes.br"), cifra de dinheiro ("U$", "R$50,00") e e-mail ("lvmorellato@inf.ufes.br"), por exemplo. Pode ocorrer erro na análise da frase, já que a etiquetagem correta das palavras e expressões é fundamental para a próxima etapa, que é a aplicação das regras gramaticais para a identificação dos sintagmas nominais.

Com as palavras que compõem os textos definidas e etiquetadas com as classes gramaticais, o módulo EM envia essas informações para o ISN, que iniciará a próxima etapa do processamento dos textos para a recuperação dos sintagmas nominais.

4.4 ISN – Identificador de Sintagmas Nominais

O ISN – Identificador e Sintagmas Nominais – é responsável por identificar e extrair os sintagmas nominais das frases contidas nos textos. Executa tarefas como delimitar frases nos textos, analisar sintaticamente essa frases de acordo com uma gramática e implementar essa gramática.

O ISN considera que uma frase termina quando encontra um sinal de pontuação. O analisador sintático foi implementado utilizando a ferramenta Bison [Bison 2009], na qual as regras foram escritas. Usando como representação das regras uma gramática livre de contexto, o ISN implementou uma parte da gramática sintagmática do português que recebe como entrada as palavras classificadas gramaticalmente e verifica se existe uma regra através da qual seja possível gerar essa frase, e informa na saída os sintagmas nominais.

O analisador encontra a regra da frase inteira. Devido à complexidade das regras, e com o objetivo de recuperar grande percentual dos sintagmas de uma frase, foram implementados os métodos janela deslizante e janela deslizante recursiva. As seções seguintes tratam do funcionamento do identificador de sintagmas implementado como um analisador sintático de uma gramática e seus métodos de aplicá-lo na frase.

4.4.1 O identificador de sintagmas

O analisador sintático foi implementado utilizando a ferramenta Bison, um *parser* de propósito geral que converte uma gramática livre de contexto em um analisador sintático. O *parser*

implementado pelo Bison é do tipo *bottom-up* e sua arquitetura pode ser vista na Figura 4.2. A partir da sentença fornecida como entrada, a análise é feita utilizando como estruturas auxiliares uma tabela de ações, uma tabela de movimentações e uma pilha de estados.

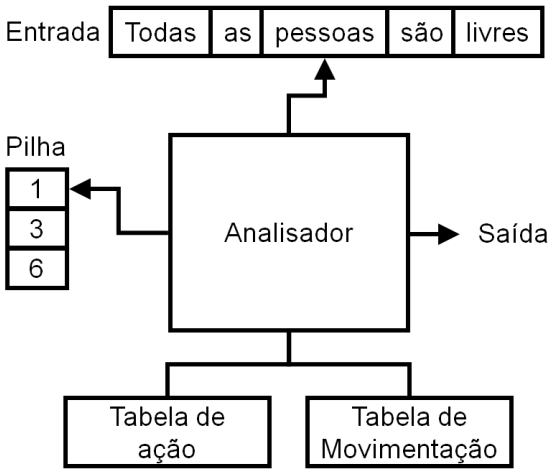


Figura 4.2: Arquitetura de um analisador sintático *bottom-up*

O funcionamento do analisador baseia-se na tentativa, por deslocamentos e reduções, de transformar um conjunto de *tokens* fornecido como entrada em um único grupo como saída, sendo o símbolo inicial da gramática. De forma geral, pode-se considerar que as palavras da frase, que chamaremos de *tokens*, são passadas para o analisador, que vai lendo uma por vez e verificando se elas se encaixam em alguma regra. Quando é confirmado, o *token* é colocado na pilha. O analisador também faz reduções quando um conjunto pode ser substituído por um único símbolo que represente o agrupamento.

O analisador sintático nem sempre reduz imediatamente assim que analisa os últimos *"n" tokens* e os agrupamentos se encaixam na regra. Isso porque tal estratégia simples é insuficiente para lidar com a maioria das linguagens. Em vez disso, quando a redução é possível, o analisador por vezes olha para o próximo *token* na ordem (*looks ahead*), a fim de decidir o que fazer. Esse *token* será chamado de *token lookahead*.

Quando um *token* é lido, não é imediatamente deslocado. Primeiro, ele torna-se o *token lookahead*, que não está na pilha. Assim, o *parser* pode executar uma ou mais reduções de *tokens* e agrupamentos na pilha, enquanto o *token lookahead* permanece de fora. Quando não há mais reduções, o *token lookahead* é deslocado para a pilha. Isso não significa que todas as possíveis reduções tenham sido feitas; dependendo do tipo do *token lookahead*, algumas regras podem optar por adiar a sua aplicação.

A função principal do analisador é executada usando uma máquina de estados finitos. Os valores colocados na pilha do *parser* não são simplesmente *tokens* com códigos, que represen-

tam toda a sequência de símbolos terminais e não-terminais. O estado atual recolhe todas as informações anteriores sobre a entrada que são relevantes para decidir o que fazer a seguir. A Figura 4.3 mostra o exemplo de uma frase sendo analisada seguindo na máquina de estados finitos e guardando seus elementos na pilha.

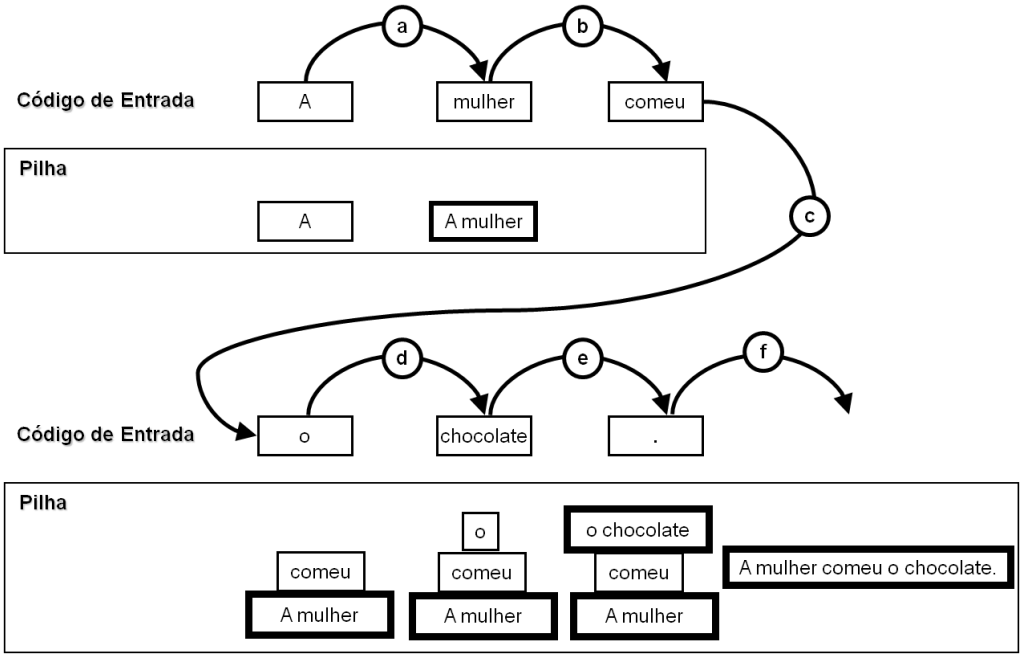


Figura 4.3: Exemplo de processamento de uma frase em um analisador *bottom-up*

Cada vez que um *token lookahead* é lido, o estado atual do *parser* juntamente com o tipo de *token lookahead* são consultados em uma tabela. A entrada da tabela pode conter o comando *"Desloque o token lookahead"*, além disso, especifica o novo estado do *parser*, que é colocado no topo da pilha. Ou, então, o comando *"Reduzir usando regra número n"*. Isso significa que um determinado número de *tokens* ou agrupamentos são tomados fora do topo da pilha e substituídos por um agrupamento. Em outras palavras, é mostrado o número de estados que estão surgindo a partir da pilha, e um novo estado é empilhado.

Existe uma outra alternativa: a tabela pode dizer que o *token lookahead* é errado no estado atual, o que provoca um erro e o processamento recomeça. Quando o analisador encontra-se nessa situação, ele efetivamente se divide em vários *parsers*, um para cada possível deslocamento ou redução. Esses *parsers* prosseguem no processamento normalmente. Algumas das pilhas podem encontrar outros conflitos e dividir ainda mais e, como resultado, em vez de uma sequência de estados, tem-se uma árvore de estados.

Cada bloco representa um palpite sobre se o que se deve analisar está correto. Uma entrada adicional pode indicar que um palpite estava errado, caso em que a pilha relativa desaparece. Caso contrário, as ações semânticas geradas em cada pilha são guardadas, em vez de serem

executadas imediatamente. Quando uma pilha desaparece, suas ações semânticas salvas não são executadas.

Quando uma redução provoca duas pilhas equivalentes, seus conjuntos de ações semânticas são salvos juntos ao estado que resultou da redução. Dizemos que duas pilhas são equivalentes quando ambas representam a mesma sequência de estados, e cada par de estados correspondentes representa um símbolo da gramática que produza o mesmo segmento do fluxo de *tokens* da entrada.

Gramáticas realmente ambíguas ou não determinísticas podem exigir grandeza exponencial em tempo e espaço para serem processadas. Elas geralmente não são de interesse prático devido a seu mau comportamento. Para tratar a gramática não ambígua implementada, o Bison atualmente usa uma estrutura de dados simples que requer tempo de processamento proporcional ao comprimento da entrada vezes o número máximo de pilhas necessárias para qualquer prefixo da entrada. Geralmente, o *parser* fica "em dúvida" somente para alguns *tokens* de cada vez.

De toda forma, o Bison permite utilizar uma estrutura de dados que pode processar qualquer gramática não ambígua em tempo quadrático (no pior caso), e qualquer gramática livre de contexto (possivelmente ambígua) em tempo cúbico (também no pior caso). No entanto, a estrutura mais simples usada pelo Bison produz resultado satisfatório neste trabalho.

As regras gramaticais da linguagem utilizadas na implementação do SISNOP foram escritas de maneira recursiva. Essa recursividade pode ser definida do lado direito ou esquerdo. Uma regra é chamada recursiva à esquerda quando o seu resultado não-terminal aparece como o primeiro símbolo do lado esquerdo da regra. Um exemplo de recursão à esquerda poder ser visto a seguir:

```
Nome_Composto: nome
             | Nome_Composto  nome
```

A mesma regra pode ser definida utilizando uma recursão à direta, em que o resultado não-terminal está mais à direta. Isso pode ser visto abaixo:

```
Nome_Composto: nome
             | nome Nome_Composto
```

Qualquer tipo de sequência pode ser definida usando recursão à esquerda ou à direita, mas deve-se buscar sempre o uso da recursão à esquerda, porque se pode analisar uma sequência de qualquer número de elementos em um espaço de pilha delimitado. A recursão à direita utiliza

espaço da pilha na proporção do número de elementos na sequência, já que todos os elementos devem ser deslocados para a pilha antes que a regra possa ser aplicada. As regras gramaticais do ISN são apresentadas na próxima seção, e foram implementadas fazendo uso de recursão à esquerda com o intuito de otimizar o analisador.

4.4.2 As regras gramaticais

Seja G uma gramática livre de contexto definida como $G =< T, N, S, P >$, em que T representa o conjunto de símbolos terminais; N é o conjunto de símbolos não-terminais; S é o símbolo inicial ($S \in N$); e P é o conjunto de produções, ou regras de derivação. Um exemplo de aplicação da definição de gramática livre de contexto é definir um subconjunto da língua portuguesa da forma $G2 =< T, N, S, P >$ em que:

- T = {“homem”, “casa”, “comprou”, “o”, “a”}
- N = {Sentença, SN, SN, Artigo, Verbo, Substantivo, Complemento}
- S = Sentença
- P = {
  - Sentença ⇒ SN SV
  - SN ⇒ Artigo Substantivo
  - SV ⇒ Verbo SN
  - Artigo ⇒ “o”
  - Artigo ⇒ “a”
  - Substantivo ⇒ “homem”
  - Substantivo ⇒ “casa”
  - Verbo ⇒ “comprou”

  }

Por essa gramática é possível reconhecer como válida a sentença: *“O homem comprou a casa”*. Permite também gerar as frases *“A casa comprou o homem”*, *“A homem comprou o casa”* e *“O casa comprou a homem”*. Além disso, é possível identificar e recuperar os sintagmas nominais *“o homem”* e *“a casa”*.

No analisador desenvolvido neste trabalho, os símbolos terminais utilizados não serão as palavras, mas sim as categorias gramaticais fornecidas pelo etiquetador morfológico. Os símbolos não-terminais serão estruturas como sintagmas nominais (SN) e verbais (SV). O símbolo inicial do analisador sintático do SISNOP é a frase, e o conjunto de produção são todas as regras que são definidas e que formam uma sentença válida, ou seja, a gramática implementada. A seguir são mostradas as primeiras regras, formadas a partir do símbolo inicial *"frase"*, para definição da gramática:

```
Frase: Oracao
     | Oracao conjuncao Oracao
     | adverbio Oracao
     | pr_interrogativo Oracao ;
```

A partir dessas regras iniciais são derivadas recursivamente as demais, implementando assim um subconjunto de regras gramaticais da língua portuguesa. Dispondo dessas regras, o ISN faz então a análise sintática da frase inteira, mas somente os sintagmas nominais são extraídos por serem de principal interesse do SISNOP. Como o foco são os sintagmas nominais, não será detalhada aqui a composição de outros elementos como o sintagma verbal, e sim as possíveis formações de um SN.

Um sintagma nominal é composto pelos elementos básicos: núcleo, determinantes e modificadores, sendo os dois últimos facultativos na formação do sintagma nominal. Os determinantes e os modificadores podem inverter sua posição inicial. A formação mais comumente encontrada pode ser vista na Figura 4.4, na qual são mostradas, para cada um dos elementos básicos que compõem um sintagma, as classes gramaticais que eles podem assumir.

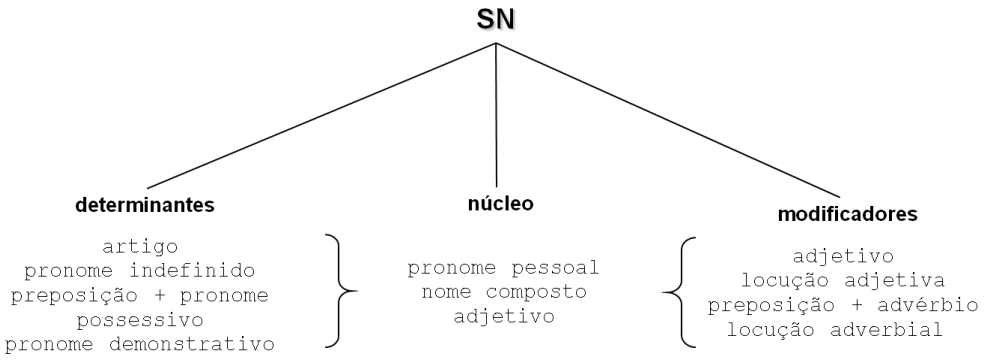


Figura 4.4: Classes gramaticais dos componentes de formação de um sintagma nominal

Como os elementos determinantes e modificadores podem estar presentes antes ou depois do núcleo, ou serem facultativos, formam-se um conjunto de composições possíveis de sintag-

mas nominais. A Tabela 4.2 mostra todas essas combinações dos elementos (núcleo, determinante e modificadores) e para cada umas delas, algumas regras formadas pela composição das classes gramaticais e exemplos de sintagmas nominais dessas regras.

As primeiras regras exibidas na Tabela 4.2 são da combinação do núcleo, sem determinantes e modificadores. As classes gramaticais que fazem parte do núcleo são nome, adjetivo e pronome pessoal conseguindo assim identificar os sintagmas nominais *"eu"* e *"menina"*, por exemplo. Na combinação *"determinante + núcleo"* em que o determinante é a classe gramatical artigo, o núcleo não poderá ser um pronome pessoal, pois não existe o sintagma nominal *"um ela"*. Analisando cada caso de formação de um sintagma nominal foram geradas as regras da gramática que estão detalhadas no Anexo A.

Regras gramaticais	
núcleo	
SN([nome composto])	*"menina"* , *"sintagma nominal"*
SN([adjetivo])	*"linda"*, *"alegre"*
SN([pronome pessoal])	*"ela"*, *"eu"*
determinante + núcleo	
SN([artigo, nome composto])	*"a menina"*
SN([artigo, adjetivo])	*""a linda"*
SN([preposição + pronome, nome])	*"da menina"*
SN([pronome possessivo, nome])	*"minha menina"*, *"nossa felicidade"*
núcleo + modificadores	
SN([nome, adjetivo])	*"menina linda"*
SN([nome, locução adverbial])	*"menina muito rápida"*
modificadores + núcleo	
SN([advérbio, adjetivo])	*"muito alegre"*
SN([preposição + advérbio, pronome pessoal])	*"de repente ela"*
SN([adjetivo, nome])	*"linda menina"*
núcleo, determinante	
SN([nome, pronome possessivo])	*"carro meu"*
SN([pronome pessoal, pronome indefinido])	*"ele quem"*
SN([adjetivo, preposição + pronome])	*"alegria dela"*
determinante + núcleo + modificadores	
SN([artigo, nome composto, adjetivo])	*"um carro bonito"*
SN([pronome possessivo, adjetivo, advérbio])	*"minha alegria certamente"*

Tabela 4.2: Exemplo de regras implementadas para cada combinação de elementos na formação do sintagma nominal

De acordo com a gramática, é possível que existam situações que deem origem a mais do que uma derivação para cada frase. O analisador do SISNOP considera a primeira árvore gerada e não processa a execução para encontrar as demais.

Para tratar um possível erro do etiquetador morfológico, como por exemplo quando ele não conseguir classificar uma palavra ou quando a classe retornada não é tratada no conjunto de possíveis classes, é criado um tipo chamado *UNKNOWN*. Se esse *token* aparecer logo depois de um verbo, será considerado também como um verbo, o mesmo ocorre se aparecer depois de um nome, como mostra as regras abaixo:

```
verbo_real:  VERBO
           | UNKNOWN  ;

nome_real:   NOME
           | UNKNOWN  ;
```

Considerando esses fatores, foram implementadas 84 combinações possíveis de sintagmas nominais da língua portuguesa, considerando nome simples e nome composto da mesma forma. Os sintagmas verbais têm 4056 possibilidades (esse valor é alto, pois um sintagma verbal pode ser um verbo mais um sintagma nominal). Assim, tem-se 340.704 regras diferentes para formação de uma oração. Vale destacar que o sistema define frase como sendo uma oração, uma oração mais uma conjunção mais outra oração, dentre outra formações, elevando ainda mais esse valor, e sua implementação é possível por meio de regras recursivas.

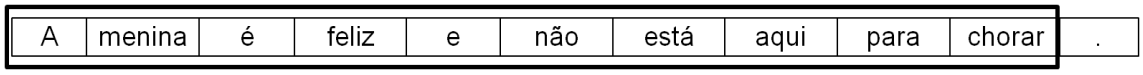
Depois de as regras gramaticais terem sido implementadas e um analisador ter sido gerado, a aplicação daquelas nas frases terá sido feita a partir de três métodos diferentes. Elas podem ser aplicadas na frase inteira, em parte dela pelo método da janela deslizante ou de maneira repetida com o método da janela deslizante recursivo.

4.4.3 Janela deslizante

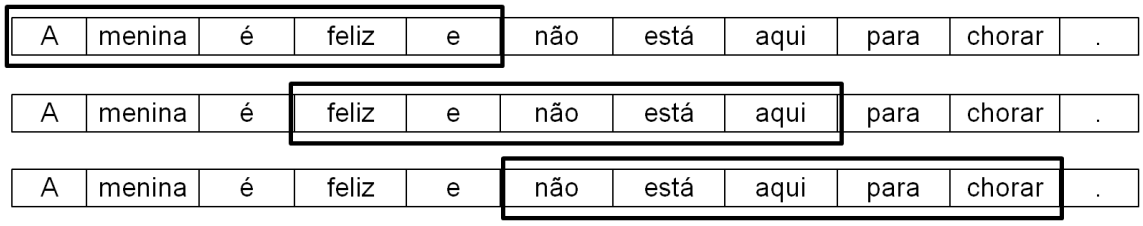
Janela deslizante (*sliding window*) é um conceito utilizado em alguns protocolos de redes [Tanenbaum 2003] para permitir a confirmação eficiente de dados trafegados ao deslizar uma janela sobre pacote de dados. Em outras áreas, como reconhecimento de imagens, é utilizada uma janela para fazer a varredura na imagem a fim de encontrar padrões [Bender e Osório 2003]. Em mineração de dados, faz-se uso de algoritmos de janela deslizante para filtragem de dados [Lee, Lin e Chen 2001], assim como em trabalhos de recuperação de informação para identificação de n-gramas.

No ISN o princípio de janela deslizante, ou seja, utilizar partes do todo de cada vez, é implementado com o objetivo de aumentar o número de sintagmas identificados. Em casos em que ocorrem erro de classe gramatical, e assim a regra não se adequa perfeitamente à frase ou, então, quando a regra não está implementada para aquela frase, mas existem regras que conseguem identificar parte dela, o método torna-se uma opção.

A Figura 4.5 contém um exemplo da aplicação do método de janela deslizante em uma frase a ser reconhecida por uma regra gramatical. Na primeira frase, Figura 4.5(a), é exibida uma janela de tamanho 10 que abrange a frase inteira. Na Figura 4.5(b) é mostrada uma janela de tamanho 5 aplicada à mesma frase, em diferentes posições após ser deslizada, até chegar ao fim da frase.



(a) Janela de tamanho 10



(b) Janela de tamanho 5 em diferentes posições na frase

Figura 4.5: Exemplo de aplicação do método janela deslizante

O Algoritmo 3 mostra a implementação do processo de janela deslizante para análise sintática de frases. A função recebe como parâmetros a frase e as dimensões da janela (tamanhos máximo e mínimo). A verificação da frase no conjunto de regras gramaticais é iniciada com uma janela de tamanho máximo (valor fornecido ou tamanho da frase, qual valor for menor). A janela desliza começando pelo início da frase e, caso o analisador sintático informe que esta não é uma sentença válida da linguagem, a janela é deslizada e um novo teste é feito. Esse processo é repetido, deslizando a janela palavra por palavra, até se chegar ao final da frase.

Caso chegue ao fim da frase sem uma resposta positiva do ISN, a largura da janela é decrementada de uma palavra, e o processo de deslizá-la é recomeçado novamente no início da frase. Isso é repetido até que o ISN reconheça a frase como válida e retorne os sintagmas encontrados, ou que tenha atingido o valor do tamanho mínimo da janela.

Os valores das dimensões estão diretamente ligados aos resultados obtidos e ao tempo de processamento da frase. Quanto maior a dimensão da janela, maior, possivelmente, será a quantidade de sintagmas identificados, em contrapartida mais tempo de processamento será

Algoritmo 3

function JanelaDeslizante (frase :Texto, min: Inteiro, max: Inteiro): Lista

Pré-condição: Frase etiquetada, valor do tamanho mínimo e máximo da janela deslizante

```
1:  tamFrase ⇐ calculaTamanho(frase)
2:  tamJanela ⇐ max
3:  while (tamJanela ≥ min) do
4:     inicioJanela ⇐ 1
5:     while inicioJanela ≥ (tamFrase − tamJanela − 1) do
6:        subFrase ⇐ frase[inicioJanela : (inicioJanela + tamJanela)]
7:        SN ⇐ ISN(subFrase)
8:        if (SN ≠ ''erro'') then
9:           return SN
10:       end if
11:       inicioJanela ⇐ inicioJanela + 1
12:    end while
13:    tamJanela ⇐ tamJanela − 1
14: end while
15: return ''erro''
```

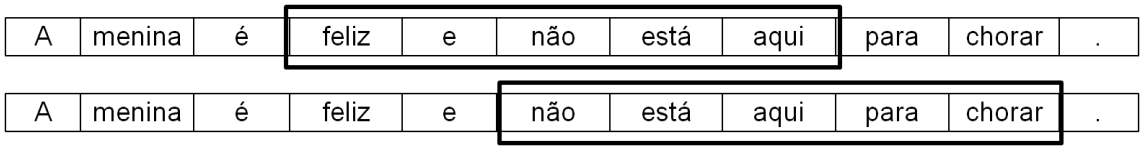
gasto na interpretação da frase. Uma adaptação no método janela deslizante foi implementada visando a melhorar os resultados. Para isso foi utilizada a recursão em cima das subfrases restantes, e as mudanças no algoritmo podem ser vistas na seção seguinte.

4.4.4 Janela deslizante recursiva

A recursividade é uma das propriedades mais importantes das línguas humanas, pois é ela que permite principalmente aos falantes produzir um número ilimitado de sentenças [Perini 2003]. Considerando a afirmação de Perini e buscando melhorar a abrangência de identificação dos sintagmas nominais, o método janela deslizante foi alterado de forma a agir recursivamente na frase.

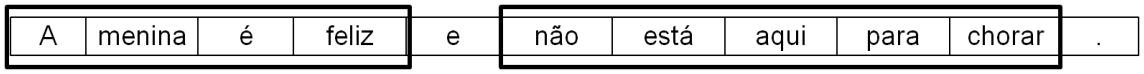
O método janela deslizante, como foi visto, busca encontrar a maior subfrase que se adeque às regras gramaticais implementadas no SISNOP. Quando uma subfrase é reconhecida como válida na linguagem, a função retorna. Com isso, algumas partes da frase podem deixar de ser identificadas, como no Exemplo 4.4. O analisador, ao identificar como frase correta o trecho *"A filha do presidente da empresa, que ainda não tem 18 anos"*, deixa de fora o restante da sentença *"não pode assumir seu lugar"*. Em um caso pior, ele poderia identificar somente *"A filha do presidente da empresa"* e perder, assim, a maior parte da frase.

"A filha do presidente da empresa, que ainda não tem 18 anos, não pode assumir seu lugar." (4.4)



(a) Método janela aplicado em uma frase

(b) Recursão aplicada ao restante da frase



(c) Resultado final

Figura 4.6: Exemplo de aplicação do método janela deslizante recursivo

Com o intuito de aumentar a abrangência na frase e melhorar os resultados obtidos recuperando o maior número de sintagmas possível, implementou-se o método janela deslizante recursiva. A Figura 4.6 mostra um exemplo de como o método recursivo pode ser aplicado em uma frase. Na Figura 4.6(a), é mostrada uma janela de tamanho 5 sendo deslizada na frase e encontrando uma regra que se adequa. A Figura 4.6(b) mostra a recursão aplicada ao restante da frase que não foi identificado. Primeiro uma janela de tamanho 5 é testada na frase, e depois uma de tamanho 4 se ajusta em uma regra implementada. Por fim, na Figura 4.6(c) é exibido o resultado final de identificação de sintagmas nominais na frase, utilizando o método de janela deslizante recursivo.

O Algoritmo 4 mostra a implementação da janela deslizante recursiva realizada por meio de uma adaptação do método anterior. A função, além de receber como parâmetros a frase etiquetada, e os tamanhos máximo e mínimo da janela, agora recebe a posição inicial e final da frase, a ser percorrida pela janela, sendo possível assim chamar a função recursivamente.

O ISN é um analisador sintático que identifica sintagmas nominais, no qual foram implementados os métodos de janela deslizante e sua versão recursiva para melhorar seus resultados. É necessário, então, obter as informações de gênero, número e grau de cada uma das palavras contidas nos sintagmas nominais identificados.

Algoritmo 4

function JanelaDeslRec(frase :Texto, min, max, inicio, fim: Inteiro): Lista

```
1: if fim < 1 then
2:    return []
3: end if
4: tamJanela ⇐ max
5: while tamJanela ≥ min do
6:    inicioJanela ⇐ inicio
7:    while inicioJanela ≥ (tamanho(frase) − tamJanela − 1) do
8:       subFrase ⇐ frase[inicioJanela : (inicioJanela + tamJanela)]
9:       SN ⇐ ISN(subFrase)
10:      if (SN ≠ ''erro'') then
11:         saida ⇐ saida ++ JanelaDeslRec(frase, min, max, inicio, fim)
12:         saida ⇐ saida ++ SN
13:         aux ⇐ inicio + inicioJanela
14:         saida ⇐ saida ++ JanelaDeslRec(frase, min, max, aux, fim)
15:         return saida
16:      end if
17:      inicioJanela ⇐ inicioJanela + 1
18:   end while
19:   tamJanela ⇐ tamJanela − 1
20: end while
21: return ''erro''
```

4.5 IGNG – Identificador de Gênero, Número e Grau

As palavras que constituem um sintagma nominal podem sofrer variações de gênero, número e grau, e essas informações podem ser importantes em programas que fazem uso dos sintagmas. Para recuperar esses dados dos sintagmas nominais extraídos foi desenvolvido o módulo IGNG – Identificador de Gênero, Número e Grau de Palavras do Português –, o qual foi integrado ao SISNOP. O programa recebe como entrada uma palavra, ou um conjunto de palavras, e informa na saída o gênero, o número e o grau de cada uma delas.

O algoritmo desenvolvido no IGNG foi fundamentado na redução de palavras ao radical mais simples (*stem*), por meio da retirada dos sufixos e prefixos, permanecendo apenas a raiz da palavra, processo chamado como *stemming* [Jones e Willet 1997]. As palavras: *"apresentação"*, *"apresentar"* e *"apresentando"* podem ser reduzidas ao *stem "apresent"*.

Um *stemmer* para língua portuguesa, [Orengo e Huyck 2001], foi desenvolvido inspirado no conhecido algoritmo *Porter Stemmer* para língua inglesa, [Porter 1980]. O *Porter Stemmer* é um algoritmo *sufix-stripping* (análise e retirada do sufixo) baseado em um conjunto de regras sem utilização de nenhum dicionário. Ao ser aplicado ao português, adicionou-se uma lista de

exceções a essas regras.

O *stemmer* é composto por oito etapas que precisam ser executadas em uma certa ordem. A primeira delas verifica se uma palavra está no plural, para assim retirar os sufixos por meio da regra de redução de plural. A próxima regra é a de redução de feminino, que parte do mesmo princípio, varrer uma palavra pelo sufixo e verificar se ela se encaixa em uma regra; se sim, a redução é feita. A terceira regra é a de redução de grau. As regras seguintes são de redução de advérbio, nome, verbo, vogal temática e acentos. O algoritmo IGNG implementa somente as regras de interesse, ou seja, as referentes às três primeiras etapas do *stemmer*, que podem ser vistas na Figura 4.7. Essa três etapas são detalhadas a seguir:

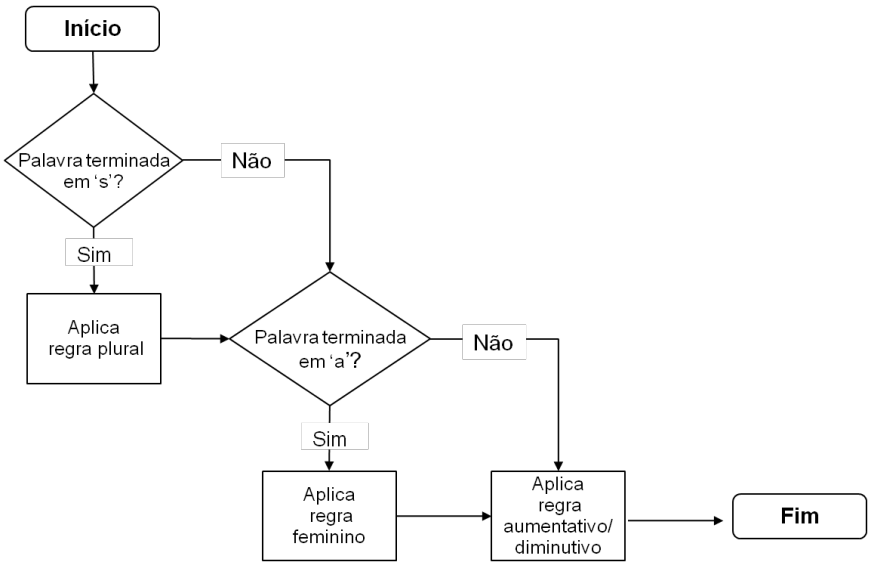


Figura 4.7: Etapas do algoritmo *stemmer* utilizadas no sistema IGNG

- Etapa 1: Redução de plural

  Essa etapa, a primeira a ser executada como pode ser vista na Figura 4.7, consiste basicamente em remover o *"s"* das palavras que não estão listadas como exceções. O plural da língua portuguesa normalmente termina com *"s"*, salvo raras exceções, como a palavra *"lápis"*. Ainda podem ser exigidas algumas modificações extras como palavras terminadas em *"ns"*, que devem ter esse sufixo substituído por *"m"*. Exemplo: *"bons"* - *"bom"*.

- Etapa 2: Redução de feminino

  Essa etapa consiste em transformar a forma feminina para sua correspondente masculina. Apenas palavras terminadas em *"a"* são testadas nessa etapa, mas nem todas são convertidas. Em geral a regra troca *"a"* por *"o"* mas podem existir casos como o da palavra *"chinesa"*, que deve ser modificada para *"chinês"*.

- Etapa 3: Redução de aumentativo e diminutivo

  Substantivos e adjetivos em português apresentam muito mais formas de variação que em inglês. Palavras têm aumentativo, diminutivo e superlativo. Exemplo: *"casinha"*, em que *"inha"* no sufixo indica diminutivo.

Algoritmo 5

function IGNG(P :Lista): Lista

Pré-condição: Uma ou mais palavras de um sintagma nominal

```
1: regras ⇐ conjuntoRegras //contidas em um arquivo
2: for all palavra in P do
3:    saida ⇐ saida ++ stemmerGNG(palavra, regras)
4: end for
5: return saida
```

O Algoritmo 5 carrega o conjunto de regras definidas, junto com sua lista de exceções. A seguir, para cada uma das palavras recebidas como entrada, é chamada a função *stemmerGNG*, Algoritmo 6. Primeiramente, a função verifica se a palavra se encaixa na regra de redução de plural. Em caso positivo, aplica a regra. Faz o mesmo processo para a redução de número e grau. Logo depois, retorna as informações obtidas para a função principal.

Algoritmo 6

function stemmerGNG(palavra: Texto, conjRegras: Lista): Lista

```
1: genero ⇐ ''masculino''
2: numero ⇐ ''singular''
3: grau ⇐ ''normal''
4: regra ⇐ verificaRegra(palavra, conjRegras, ''plural'')
5: if regra0 then
6:    palavra ⇐ aplicaRegra (palavra, regra)
7:    numero ⇐ ''plural''
8: end if
9: regra ⇐ verificaRegra(palavra, conjRegras, ''feminino'')
10: if regra0 then
11:    palavra ⇐ aplicaRegra (palavra, regra)
12:    genero ⇐ ''feminino''
13: end if
14: regra ⇐ verificaRegra(palavra, conjRegras, ''grau'')
15: if regra0 then
16:    palavra ⇐ aplicaRegra (palavra, regra, grau)
17: end if
18: return genero ++ numero ++ grau
```

A execução do Algoritmo 6 para a palavra *"pobrezinhas"*, por exemplo, se dará da seguinte maneira: como a palavra termina em *"s"* e não está na lista de exceções, a regra é aplicada e o

"s" é retirado. A seguir, a palavra *"pobrezinha"* (sem o *"s"*) é testada nas regras de feminino. Como ela é terminada em *"a"* e não está em nenhuma lista de exceção (como as palavras "mapa" e "clima"), é então aplicada a regra e obtém-se a palavra *"pobrezinho"*. Por fim, a regra de grau identifica o final *"-inho"* como indicativo de diminutivo, e o resultado da função é então retornado: *"pobrezinhas: plural, feminino e diminutivo"*.

4.6 Processamento Paralelo no SISNOP

Dentre os pré-requisitos que sistemas de processamento de linguagem natural devem ter para serem aplicáveis em recuperação de informação está a habilidade de processar grandes conjuntos de textos. O tamanho das bases de dados processadas é tipicamente da ordem de *gigabytes*. Isso requer que sistemas que processam linguagem natural sejam eficientes nos quesitos tempo e espaço em memória. É impraticável, por exemplo, um *parser* que demore um segundo para processar uma ou duas sentenças.

A tarefa de maior custo computacional no SISNOP é a identificação dos sintagmas nominais, efetuada pelo analisador sintático, ISN, que necessita de grande esforço, mesmo adotando medidas para otimizar o processo, como não usar regras recursivas à direita e não implementar regras ambíguas. Devido à complexidade da gramática e do consequente número de regras para abranger a grande quantidade de casos, o ISN torna-se lento para conjuntos muito grandes de dados.

Para viabilizar a utilização prática em grandes bases de dados foi adotado o processamento paralelo de parte do SISNOP. Não houve necessidade, no início, de paralelizar o etiquetador morfológico, EM, e o identificador de gênero, número e grau, IGNG, uma vez que já eram suficientemente eficientes.

Processamento paralelo consiste em dividir uma tarefa em partes independentes e executar cada uma dessas partes em processadores diferentes. Para isso, é necessário paralelizar os algoritmos, utilizar um mecanismo para distribuir o processamento pelos diversos processadores e um mecanismo para a troca de mensagens entre os diferentes processos.

Existem diferentes métodos para implementar o processamento paralelo. Como, por exemplo, dividir os processos tendo como base as tarefas a serem realizadas (cada processador poderia realizar tarefas diferentes) ou dividir os processos utilizando como base a quantidade de dados do domínio (cada processador realiza parte da mesma tarefa com um subconjunto de dados do domínio) [Wilkinson e Allen 1999].

Como não existe interação entre as frases de um texto, de forma que cada uma pode ser processada separadamente, o algoritmo do ISN torna-se paralelizável utilizando a técnica de divisão do domínio. A partir de bibliotecas da linguagem Python (*subprocess* e *threading*) foi possível fazer a distribuição e a troca de informações dos processos entre os diversos computadores.

Algoritmo 7

function ISNParalelo (T :Texto, listMaquinas: Lista) : Lista
Pré-condição: Texto etiquetado morfologicamente
Pós-condição: Sintagmas Nominais

1: $conjFrase \Leftarrow divideFrase(T)$
2: $qtdFrase \Leftarrow count(conjFrase)/count(listMaquina)$
3: $inicio \Leftarrow 1$
4: **for all** *maquina* **in** $listaMaquina$ **do**
5: $\quad frases \Leftarrow conjFrase(inicio, inicio + qtdFrase)$
6: $\quad SN \Leftarrow SN +\!+ send(frase, maquina)$
7: $\quad inicio \Leftarrow inicio + qtdFrase$
8: **end for**
9: **return** *SN*

O Algoritmo 7 mostra como foi implementado o ISN paralelo. Como entrada, o algoritmo recebe um texto de palavras etiquetadas e uma lista de nomes de máquinas que devem estar conectadas em rede. Em seguida, a máquina principal, que está rodando o programa, faz a divisão do texto em frases e divide, igualmente, uma quantidade de frases a serem processadas para cada máquina disponível. Depois chama uma sub-rotina, passando o conjunto de frases de cada máquina e finalmente reúne os resultados recebidos.

Há desvantagens em paralelizar um sistema quando ocorre muito *overhead*, ou seja, causas que possam reduzir a eficiência e que sejam específicas do esforço de paralelização. Um sistema paralelo em que seja necessário transferir dados entre os processadores vai gastar tempo na comunicação. O desequilíbrio na distribuição de esforço computacional entre os processadores é, também, uma fonte de *overhead*. Pode ainda existir uma parte sequencial no programa e portanto reservada a um só processador, ficando os outros processadores parados à espera que o primeiro acabe.

Paralelizar deixa de ser, assim, uma boa uma opção para um conjunto de dados pequenos, devido, principalmente, ao *overhead* de comunicação entre os computadores na rede. No caso específico de paralelizar o ISN, os demais problemas de *overhead* citados não ocorrem porque as frases são distribuídas em quantidades iguais, e as tarefas de cada processador também são as mesmas.

4.7 Adaptação para utilização do *parser* PALAVRAS

No desenvolvimento do SISNOP foi escolhido o programa FORMA como etiquetador morfológico por ser uma opção de *software* atual, com bons resultados de precisão, por dispensar licença para utilização e estar disponível em código aberto, sendo possível, assim, fazer modificações caso seja necessário. Todavia, existem outros etiquetadores, e torna-se interessante a possibilidade de utilizá-los, como, por exemplo, o *parser* PALAVRAS [Bick 2000], que tem alta precisão e é citado em diversos trabalhos da área de processamento do português.

Uma adaptação no identificador de sintagmas – ISN – se fez necessária para ser possível a utilização de textos categorizados por outro etiquetador sem haver alteração nas regras gramaticais. Assim, ao se executar o ISN, pode-se informar por parâmetro que o texto passado está etiquetado pelo *software* PALAVRAS ou pelo FORMA.

O SISNOP possui um conjunto de etiquetas que se relacionam com as classes gramaticais do etiquetador para possibilitar o uso de mais de um etiquetador, igualando as categorias e fazendo as adaptações necessárias. A Tabela 4.3 mostra as etiquetas do SISNOP e as respectivas classes gramaticais utilizadas em cada uma pelos programas FORMA e PALAVRAS.

Etiqueta SISNOP	**Etiqueta FORMA**	**Etiqueta PALAVRAS**
adjetivo	adjetivo (AJ) verbo no particípio (AP)	adjetivo (ADJ) verbo no particípio (V PCP)
advérbio	advérbio (AV)	advérbio (ADV)
artigo	definido (AD) e indefinido (AI)	determinante (DET-artd)
conjunção	coordenativa (CC) subordinativa (CS)	coordenativa (KC) subordinativa (KS)
nome	substantivo (SU)	substantivo (N) nome próprio(PROP)
verbo	verbos auxiliar (VA) regulares e irregulares (VB)	verbo(V)
preposição	preposição (PR)	preposição (PRP)
prn_pessoal	pronome pessoal (PP)	pronome pessoal (PERS)
prn_demonstrativo	pronome demonstrativo (PD)	demonstrativo (SPEC-dem)
prn_indefinido	pronome indefinido (PI)	quantitavivo (SPEC-quant)
prn_possessivo	pronome possessivo (PS)	possessivo (SPEC-poss)
prn_interrogativo	pronome relativo (PL)	interrogativo (SPEC-interr)
virgula	vírgula, parênteses e traços (VG)	-
ponto	pontuações (PN)	pontuação (PU)

Tabela 4.3: Relacionamento das classes gramaticais do FORMA e PALAVRAS com o SISNOP

Os etiquetadores têm pequenas diferenças de classes gramaticais, por isso foi possível uma adaptação do SISNOP sem alteração de regras. O PALAVRAS, por exemplo, ao contrário do

FORMA, identifica nomes próprios e, ao utilizá-lo no SISNOP, eles foram tratados junto com os substantivos como uma única etiqueta: *"nome"*.

Outra característica do PALAVRAS é utilizar a etiqueta *"SPEC"* para definir os pronomes de maneira agrupada, diferenciando os tipos em demonstrativo (*"SPEC-dem"*), possessivo (*"SPEC-poss"*), interrogativo (*"SPEC-interr"*) e indefinido (*"SPEC-quant"*). Os números ordinais não são identificados, somente os cardinais; no caso dos advérbios, são reconhecidos os "primários" e os terminados com o sufixo *"-mente"*.

Os artigos, por sua vez, são retornados pelo programa PALAVRAS com a marcação *"DET"*, chamada de determinantes, e não são discriminados em definido e indefinido. Os verbos regulares, irregulares e auxiliares também são tratados de maneira igual, por uma única etiqueta. O mesmo ocorre com vírgulas, parênteses e traços, todos são trabalhados como pontuação ("PU"). Essas diferenças entre o FORMA e o PALAVRAS não ocasionaram problemas no SISNOP devido ao fato de as regras não necessitarem de um nível grande de detalhamento (como a necessidade de saber o tipo do verbo). No entanto, para usá-los em tarefas mais específicas, com regras mais detalhadas, deve-se realizar outro tipo de tratamento para conseguir as informações necessárias.

Como o objetivo no SISNOP não era fazer uma comparação entre etiquetadores morfológicos, a adaptação foi limitada ao *software* PALAVRAS. Com essa modificação no ISN para aceitar textos etiquetados pelo PALAVRAS, foi possível testar as bases de dados marcadas pelo *parser* de forma mais prática.

4.8 Considerações Finais

Este capítulo apresentou os algoritmos desenvolvidos para a construção do SISNOP – Sistema Identificador de Sintagmas Nominais no Português. Foi discutido sobre cada um dos três módulos que compõem o sistema: EM (Etiquetador Morfológico), ISN (Identificador de Sintagmas Nominais) e IGNG (Identificador de Gênero, Número e Grau).

No etiquetador morfológico foi utilizado o sistema FORMA, que faz a classificação gramatical. No ISN foi implementado um conjunto de regras e propostos dois métodos de analisar uma frase, janela deslizante e sua versão recursiva. Com eles foi possível identificar e extrair sintagmas nominais nas frases. Com o IGNG, aos sintagmas extraídos foram adicionadas informações sobre as flexões de gênero, número e grau. A paralelização do ISN tornou possível utilizar o SISNOP em grandes bases de dados em tempo hábil. E a adaptação para utilizar outros etiquetadores morfológicos tornou prático o uso de textos marcados pelo *software* PALAVRAS

pelo SISNOP.

5 *Experimentações e Avaliação dos Resultados*

"*É preciso ter um desprezo saudável pelo impossível. É preciso tentar coisas que a maioria das pessoas não tentaria.*"

Frase ouvida na faculdade por Larry Page, co-fundador da Google

Este capítulo apresenta os experimentos realizados no sistema desenvolvido e os resultados obtidos.

5.1 Introdução

Este capítulo apresenta os testes e as avaliações dos resultados obtidos pelo SISNOP. Buscando uma avaliação mais detalhada foi feito um teste individual de cada um dos módulos, EM – Etiquetador Morfológico –, ISN – Identificador de Sintagmas Nominais – e IGNG – Identificador de Gênero, Número e Grau. Para isso, utilizou-se uma base de dados formada por um conjunto de 186 frases e uma análise manual dos resultados obtidos.

Foram realizados testes em corpora maiores, como o CETEMFolha, composto por mais de 340 mil textos somando 24 milhões de palavras em português brasileiro e o CETEMPúblico, com aproximadamente 180 milhões de palavras em português europeu.

O ambiente computacional utilizado na realização dos experimentos foi o laboratório NINFA, pertencente à Universidade Federal do Espírito Santo. Foram usadas 10 máquinas com processador AMD Turion X2 TL-58, 1.9GHZ com 512Mb de memória RAM. Para a compilação dos programas foram usados Bison versão 2.3, compilador C GCC versão 4.3 e Python 2.6 instalados sob o sistema operacional Debian GNU/Linux 5.0 - Lenny.

5.2 Avaliação SISNOP

O SISNOP é composto de três módulos, EM, ISN e IGNG, que, juntos, realizam a função do sistema, que é identificar e extrair sintagmas nominais de textos. Os testes individuais de cada um permitem, além de uma avaliação do módulo, um estudo para que não aconteçam erros em cadeia. Se, por exemplo, o etiquetador morfológico comete muitos erros de classificação, isso levará o identificador de SN a cometer também vários erros.

O corpus utilizado foi o conjunto de testes usado na LXGram, uma gramática computacional para o processamento linguístico profundo do português, também utilizado no trabalho de Costa [Costa 2007], composto por 186 frases, sendo 38 delas falso-positivas, disponível no Anexo B com os sintagmas nominais destacados em negrito em cada frase, assim como seu núcleo.

5.2.1 Avaliação do módulo EM

O etiquetador morfológico utilizado no SISNOP é o *software* FORMA, que apresenta uma precisão de 95% [Gonzalez, Lima e Lima 2006]. A precisão do etiquetador é de grande importância na qualidade dos resultados do SISNOP. Caso os componentes da frase não sejam

identificados corretamente, as regras gramaticais aplicadas ocasionam erros, e todo o sistema fica comprometido.

Ao submeter as 186 frases do conjunto de testes ao FORMA, foram identificados 1241 *tokens*, incluindo sinais de pontuação, como mostra a Tabela 5.1. Após uma análise manual do resultado, observou-se que 1203 palavras estavam classificadas corretamente, ocorrendo, assim, 38 erros. Com isso, obteve-se uma precisão de 96,93% com relação à quantidade de palavras. Do total de frases, 33 foram prejudicadas com os erros de classificação. Sendo assim, 82,53% das 186 frases foram classificadas como completamente corretas.

	Resultados FORMA
Total de palavras (incluindo pontuação)	1.241
Total de palavras corretas	1.203
% de corretas	96,93%
Total de frases testadas	186
Total de frases corretas	153
% de corretas	82,25%

Tabela 5.1: Resultados dos experimentos do módulo EM

"Atacaram um falso inspetor" (5.1)

"Chegou um falso médico chinês" (5.2)

Algumas das classificações incorretas do etiquetador foram de palavras que podem ser substantivos ou adjetivos, dependendo do contexto. Isso ocorre no Exemplo 5.1: a palavra *"falso"*, que é um adjetivo, foi classificada como um substantivo. O mesmo fato ocorre em três palavras da frase do Exemplo 5.2, em que os adjetivos *"falso"* e *"chinês"* e o substantivo *"médico"* foram classificados erroneamente, como mostra a saída do programa FORMA:

```
Chegou _VB   # Verbo
um     _AI   # Artigo Indefinido
falso  _SU   # Substantivo
médico _AJ   # Adjetivo
chinês _SU   # Substantivo
.      _PN   # Ponto
```

Outro erro ocorreu com a palavra *"certo"* no Exemplo 5.4. O FORMA a classificou, tanto

no Exemplo 5.3 como no 5.4, como pronome indefinido, sendo que no Exemplo 5.4 a palavra está modificando o substantivo *"capítulos"*, logo tem função adjetiva.

"Estão aqui certos dois capítulos." (5.3)

"Estão aqui dois certos capítulos." (5.4)

O FORMA também considera quebra de linha como fim de frase. Isso pode ser vantajoso em textos que utilizam muito enumeração e tabela, como os científicos, para identificar títulos e subtítulos, etc. As frases classificadas foram, então, submetidas ao identificador de SN, e os resultados dos testes são mostrados na próxima seção.

5.2.2 Avaliação do módulo ISN

As frases classificadas pelo etiquetador morfológico são enviadas ao ISN que, por meio de aplicação das regras gramaticais implementadas, busca identificar os sintagmas nominais nas frases e recuperá-los.

Como métrica de avaliação, serão utilizados precisão, abrangência e F_1. A precisão (Equação 5.5) faz a razão entre os sintagmas nominais identificados corretamente e o total de sintagmas extraídos com o ISN. A abrangência, em inglês chamado de *recall*, visa a avaliar, do total de sintagmas do corpus, quantos foram identificados de forma correta pelo sistema; a Equação 5.6 mostra como ela é calculada.

Buscando um equilíbrio entre precisão e abrangência, e não somente avaliar uma das medidas isoladamente, utiliza-se a métrica F_1, cuja fórmula pode ser vista na Equação 5.7. Avalia-se a média harmônica ponderada em que β é considerado o grau relativo de importância atribuída à precisão e à abrangência. Ao utilizar $\beta = 1$ dá-se a mesma importância para as duas medidas, de forma que quanto maior o valor de F_1, melhor o resultado.

$$Precisao = \frac{SNs\ corretos}{total\ de\ SNs\ identificados} \tag{5.5}$$

$$Abrangencia = \frac{SNs\ corretos}{total\ de\ SNs\ existentes\ no\ corpus} \tag{5.6}$$

$$F_{\beta=1} = \frac{(\beta^2+1) * Precisao * Abrangencia}{\beta^2 * Precisao + Abrangencia} \tag{5.7}$$

O conjunto de testes foi o mesmo utilizado ao avaliar o EM, o corpus disponível no Anexo B

de [Costa 2007]. As palavras categorizadas pelo etiquetador foram fornecidas como entrada ao ISN. Na primeira parte da Tabela 5.2, foram exibidas algumas características do corpus de teste utilizado no identificador de sintagmas. Das 186 frases, 38 são as chamadas falso-positivas, frases testadas no sistema mas que não estão corretas gramaticalmente. O corpus tem um total de 1.031 palavras, levando a uma média de 5.54 palavras por frase. Ele contém 224 sintagmas nominais, fornecendo uma média de 1.50 por frase, sendo que 3 é a quantidade máxima de SN em uma frase, e a menor quantidade é 1.

Dados do conjunto de teste	
Total de frases	186
Total de frases corretas	149
Total de palavras	1.031
Total de SN	224
Média de palavras por frase	5,54
Média de SN por frase	1,50
Qtd máxima de SN na frase	3
Resultados ISN	
Total de SN identificados	188
Total de SN ident. corretos	155
Precisão	82,45%
Abrangência	69,20%
F_1	75,24%

Tabela 5.2: Resultados dos experimentos do módulo ISN

No conjunto de teste considerado, o ISN conseguiu identificar 188 sintagmas nominais dos 224 presentes. Analisando manualmente cada um desses 188 sintagmas, observou-se que em 14 deles ocorreu alguma falha. Sendo assim, 174 foram extraídos corretamente de acordo com a gramática implementada. Com essas informações pode-se calcular a precisão, 82.45%, a abrangência, 69.20% e o F_1, 75.24% (Tabela 5.2).

Os erros encontrados foram de sintagmas parcialmente identificados como mostra o Exemplo 5.8, que deveria ter recuperado o SN *"os dois primeiros filmes"*. Os Exemplos 5.9 e 5.10 são frases presentes no experimento em que os sintagmas nominais foram identificados corretamente.

*"Os dois [primeiros **filmes**] passaram aqui."* (5.8)

*"[Os primeiros dois **filmes**] passaram aqui."* (5.9)

*"[Todos os **seres humanos**] são [**livres**]"* (5.10)

Em 36 das 186 frases testadas não foi possível identificar sintagmas nominais, como nos casos dos Exemplos 5.11 e 5.12. A sua regra de formação, responsável por dizer que a frase é válida de acordo com a linguagem, não estava contida na gramática implementada. Outro motivo para a ocorrência de falha ao processar as frases pode ter se dado por um erro na classificação gramatical das palavras pelo módulo EM.

"Chegou uma encomenda tua." (5.11)

"Aquele carro ali estava aqui ontem." (5.12)

Das 38 frases falso-positivas testadas no ISN, somente uma não foi considerada correta, ao contrário do LXGram, que considerou todas essas frases como incorretas. Quando o SISNOP acerta as frases falso-positivas, tem-se a vantagem de, mesmo com a frase fornecida de maneira incorreta, ser possível recuperar os SN. O objetivo não é analisar as frases completas e verificar se estão corretas, mas sim identificar sintagmas nominais.

Os erros nessas frases estão ligados normalmente a um erro na ordem dos elementos, ou à falta de alguns deles. No Exemplo 5.13, falta um substantivo depois do artigo indefinido *"um"*. Já no Exemplo 5.14 há um pronome *"seu"* a mais na frase. No Exemplo 5.15, a formação da frase está correta, mas o sentido não. Ao trocar o adjetivo *"mero"* pelo também adjetivo *"oficial"*, a frase ficará correta: *"Atacaram um inspetor oficial"*.

*"Isso era um [com uma **antena**]."* (5.13)

*"O seu [seu **irmão**] está aqui."* (5.14)

*"Atacaram um [**inspetor** mero]."* (5.15)

Mesmo com os erros listados, o ISN obteve bons valores nas métricas utilizadas. Buscando melhorar esses resultados, dois métodos foram desenvolvidos para identificar os sintagmas nominais nas frases: janela deslizante e janela deslizante recursiva. Ao testá-los nas frases anteriormente utilizadas não houve melhora ou mudança dos resultados. Isso ocorreu devido à pequena quantidade de palavras da frase e à baixa complexidade delas. A seguir, esses dois métodos são testados em um conjunto de frases com maior quantidade de palavras e, consequentemente, maior quantidade de sintagmas nominais.

ISN – Janela deslizante e janela deslizante recursiva

As regras gramaticais foram implementadas para identificar sintagmas nominais em frases compostas por, no máximo, duas orações. A implementação de todas as possibilidades de frases, que são compostas de um grande número de orações, que, por sua vez, são constituídas de diversos sintagmas, se tornaria tão complexa que não seria possível utilizá-la com tempo hábil de resposta.

Para frases mais complexas, com grande número de orações, são aplicados os métodos de janela deslizante nas suas versões recursiva e não recursiva. A fim de facilitar entendimento, os métodos serão chamados de SISNOP-Janela e SISNOP-Recursivo, e o método que não utiliza o princípio de janela deslizante será chamado de SISNOP. Para avaliação comparativa entre os métodos de janela deslizante foi utilizado um outro conjunto de testes composto por frases mais complexas, porém em menor número, considerando que a análise é feita de maneira manual.

O corpus foi obtido do Floresta Sintática[1], que é um conjunto de frases analisadas morfos-sintaticamente. O projeto Floresta Sintática é uma colaboração entre a Linguateca e o projeto VISL[2] e contém textos em português (do Brasil e de Portugal) analisados automaticamente e revistos manualmente por linguistas.

As frases foram retiradas de uma das quatro partes do Floresta Sintática, chamada Bosque, que é a parte totalmente revista por linguistas. O Bosque é composto por 9.368 frases e, desde 2007, vem passando por um novo processo de revisão, durante o qual foram corrigidas algumas pequenas inconsistências. Por ser o corpus mais correto do Floresta, é o mais aconselhado para pesquisas em que não se prioriza tanto a quantidade, mas sim a precisão dos resultados, como neste caso.

O conjunto contém 24 frases, totalizando 889 palavras e 215 sintagmas nominais, como mostra a primeira parte da Tabela 5.3. As frases têm uma média de 37.00 palavras e 8.95 de sintagmas nominais, valores bem superiores a 5.54 e a 1.50, respectivamente, dos testes anteriores. A frase a seguir é a que contém o maior número de sintagmas nominais: 18.

*É um pouco [a **versão**] de [uma **espécie**] de [outro **lado**] de [a **noite**], a [meio **caminho**] entre [os **devaneios**] de [uma **fauna** periférica], seja de [**Lisboa**], [**Londres**], [**Dublin**] ou [**Faro**] e [**Portimão**], e [a **postura** circunspecta] de [os **fiéis**] de [a **casa**], que de [**ela**] esperam [a **música**] [**geracionista**] dos 60 ou dos 70.*

[1] http://www.linguateca.pt/Floresta/
[2] http://visl.sdu.dk/visl/pt/

Dados do conjunto de teste		
Total de frases	24	
Total de palavras	889	
Total de SN	215	
Média de palavras por frase	37	
Média de SN por frase	8,95	
Qtd máxima de SN na frase	18	
	Janela Deslizante	**Recursiva**
Tempo de processamento	0.098s	0.119s
Total de SN identificados	42	99
Total de SN ident. corretos	38	89
Precisão	90,48 %	89,90 %
Abrangência	17,67 %	41,40 %
$F_1\beta = 1$	29,57 %	56,69 %
$F_1\beta = 0.5$	38,13 %	64,65 %
$F_1\beta = 0.25$	49,61 %	72,83 %

Tabela 5.3: Resultados comparativos dos métodos Janela Deslizante e Janela Deslizante Recursiva

O teste sem a utilização dos métodos de janela deslizante não conseguiu identificar nenhuma das 24 frases. Isso porque a gramática não foi implementada com a complexidade exigida pelo corpus testado. O tamanho da janela foi variando do valor mínino estabelecido, 3, ao valor máximo, fixado como o tamanho de cada frase. Os resultados são exibidos na Tabela 5.3.

Dos 215 sintagmas nominais contidos nos corpus, 42 foram identificados pelo SISNOP-Janela e 99 pelo SISNOP-Recursivo e, destes, 38 e 89 estavam corretos, respectivamente. Isso fornece valores muito bons de precisão – 90,48 % e 89,90 % – mas valores nem tão bons de abrangência – 17,67% e 41,40%, respectivamente.

O gráfico da Figura 5.1 mostra um comparativo da porcentagem de acertos de cada um dos métodos para cada uma das 24 frases do corpus de teste. Observa-se que o SISNOP-Recursivo apresenta boa melhora em relação ao SISNOP-Janela, sendo 100,0% o melhor resultado obtido pelo método recursivo.

A seguir uma das três frases em que o número de sintagmas identificados foi igual nos dois métodos, ou seja, o método recursivo não conseguiu encontrar nenhuma frase válida ao ser aplicado no restante da frase que identificou. Os métodos encontram um sintagma nominal (*"a proposta"*) dos quatro sintagmas presentes e vistos abaixo:

*"[**Junqueiro**] foi ainda confrontado com [o **fato**] de não ter falado com [o **ministro**] antes de avançar com [a **proposta**]."*

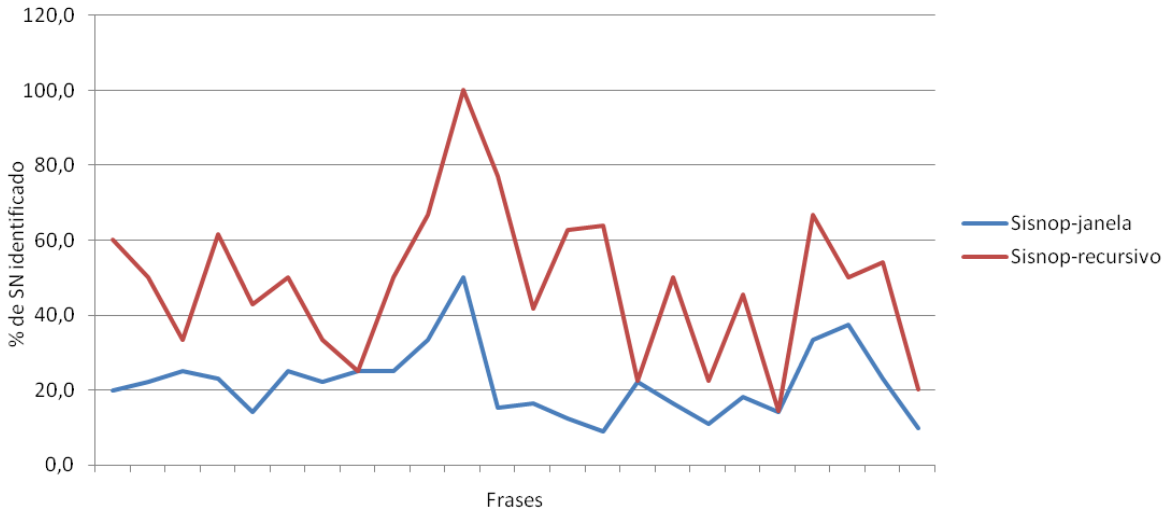


Figura 5.1: Gráfico comparativo entre o método SISNOP-Janela e o SINOP-Recursivo

Como esse experimento foi composto por um conjunto com número pequeno de frases, a diferença de tempo de processamento entre os dois métodos, SISNOP-Janela e SISNOP-Recursivo é mínima, tornando então o método recursivo mais vantajoso, como mostraram a Tabela 5.3 e o Gráfico 5.1. Mais à frente tem-se uma avaliação com corpus maiores em que é possível analisar o tempo de processamento de cada um dos métodos em conjuntos de dados maiores. A seguir o módulo que identifica gênero, número e grau das palavras recuperadas no ISN é avaliado.

5.2.3 Avaliação do módulo IGNG

Os sintagmas extraídos do módulo ISN foram passados ao IGNG de forma a obter as informações sobre gênero, número e grau. Considerando somente as palavras distintas, houve um total de 131 palavras. Dessas, 126 foram classificadas corretamente, gerando um total de 96,18%.

	Resultado IGNG
Total de palavras	131
Total de palavras corretas	126
% Correta	96,18 %
Total de palavras erradas	5
Total de erros de gênero	2
Total de erros de número	0
Total de erros de grau	3

Tabela 5.4: Resultados dos experimentos do módulo IGNG

A Tabela 5.4 mostra também que, dos erros encontrados, dois foram relacionados a encontrar gênero e três relativos ao grau. Isso normalmente ocorre devido ao fato de a lista de exceções não estar completa.

Abaixo há a saída do sistema IGNG para três termos fornecidos como entrada. Ocorre erro na identificação das palavras *"irmão"* e *"invasão"*, pois foram consideradas aumentativo (ex.:*"grandão"* e *"casarão"*), devido ao fato de não estarem presentes na lista de exceção da regra de palavras terminadas *"ão"*. A palavra *"chão"*, por exemplo, está na lista e pode ser classificada corretamente.

```
> invasão +masc +sing +aum
> irmão   +masc +plu +aum
> chão    +masc +sing +norm
```

O IGNG acertou casos como os das palavras *"galopante"*, *"pianista"* e *"participante"*, que são substantivos de dois gêneros, pois tanto podem ser feminino (*"a galopante"* , *"a pianista"*) como masculino (*"o pianista"* , *"o participante"*).

5.3 Avaliação em corpora

Os experimentos anteriores utilizaram um conjunto de testes de maneira que fosse possível avaliar a qualidade dos sintagmas identificados, sendo esse um trabalho executado manualmente por um especialista. É necessário verificar o comportamento do SISNOP para corpus maiores, com isso, a avaliação não será manual, e será utilizado como critério a quantidade de frases e de palavras identificadas. Nesse experimento utilizou-se o método SISNOP-Recursivo, pois ele demonstrou melhores resultados em conjunto de frases semelhantes às que serão utilizadas nessa avaliação.

Foram utilizados três conjuntos de testes para a avaliar a extração automática de sintagmas nominais. O primeiro deles é composto pelos 15 textos utilizados por Kuramoto no escopo de sua tese de doutorado [Kuramoto 1999] e disponível nela como anexo. Souza [Souza 2005], na avaliação do seu trabalho de identificação de sintagmas, também utilizou os textos de Kuramoto.

O segundo experimento foi realizado no CETENFolha – Corpus de Extratos de Textos Eletrônicos NILC/Folha de São Paulo –, que é um corpus em português brasileiro, criado pelo projeto Processamento Computacional do Português, que depois deu origem à Linguateca, com base nos textos do jornal Folha de São Paulo. O corpus inclui os textos das 365 edições da Folha

de São Paulo, num total aproximado de 24 milhões de palavras e destina-se primariamente a todos que desenvolvem programas que processam a língua portuguesa.

O CETENFolha está dividido em 340.947 textos, classificados por semestre e cadernos do jornal do qual provém. Como o texto está dividido em parágrafos e frases, e os títulos e os autores dos artigos estão assinalados, foi necessário construir um *script* (desenvolvido na linguagem Python) para retirar as marcações dos textos. Assim, extraindo o conteúdo principal de cada notícia, sem as informações adicionais, e escolhendo somente um subconjunto do corpus, chegou-se a um total aproximado de 4 milhões de palavras.

Utilizou-se também no experimento o CETEMPúblico – Corpus de Extratos de Textos Eletrônicos MCT/Público – um corpus de aproximadamente 200 milhões de palavras em português europeu, criado também pelo projeto que deu origem à Linguateca, retirado de O PÚBLICO, um jornal diário português de grande circulação. Os trabalhos de [Rocha e Santos 2000] e [Santos e Rocha 2001] descrevem as características particulares desse corpus que contém, em sua maioria, textos em português europeu, embora haja alguns textos de autores brasileiros e africanos.

Santos, em [Santos 2005], apresenta um método de identificação de sintagmas nominais por meio de Aprendizado de Máquina, e utilizou para teste o corpus Mac-Morpho[3], um corpus fechado e anotado, formado por artigos jornalísticos retirados da Folha de São Paulo, ano 1994. Como os textos encontrados em Mac-Morpho estão contidos na coleção CETENFolha, este corpus não foi utilizado no conjunto de testes aqui descrito.

	Corpora		
	Kuramoto	CETENFolha	CETEMPúblico
Total de frases	1.001	209.643	7.918.525
Total de frases reconhecidas	958	205.720	7.490.263
%	**95,70%**	**98,12%**	**94,59%**
Total de palavras	32.499	4.240.068	203.530.238
Total de palavras reconhecidas	12.322	2.350.035	87.218.446
%	**37,91%**	**55,42%**	**42,85%**
Total de SN identificados	3.447	601.905	23.682.200

Tabela 5.5: Resultados de experimentos em corpora

Assim, dispondo dos três corpus: Kuramoto, CETENFolha e CETEMPúblico, os experimentos foram realizados e os resultados podem ser vistos na Tabela 5.5. Kuramoto, com aproximadamente mil frases, reconheceu 95.70% delas, o CETENFolha, com 200 mil, 98,12% e

[3]http://www.nilc.icmc.usp.br/lacioweb/corpora.htm

o CETEMPúblico, com quase 8 milhões, teve uma abrangência de reconhecimento de 94.59% das frases.

Como o método utilizado foi o SISNOP-Recursivo, avaliou-se também a quantidade de palavras reconhecidas em frases válidas, permitindo a recuperação de sintagmas nominais. Assim, tem-se que Kuramoto possui 32 mil palavras, CETENFolha, 4 milhões e CETEMPúblico apresenta 203 milhões de palavras (cerca de 50 vezes mais que o CETENFolha), como mostra a Tabela 5.5. Desses totais, obtiveram-se as seguintes porcentagens de palavras reconhecidas: 37.91% em Kuramoto, 55.42% em CETENFolha e 42.85% em CETEMPúblico.

Com esses dados, percebe-se, portanto, que praticamente nenhuma frase fica sem ser reconhecida pelo sistema, devido aos bons valores de porcentagem de frases reconhecidas. E como o método é o de janela deslizante, essa frase pode não ter sido inteiramente encaixada em uma regra gramatical, mas apenas parte dela. Ao utilizar a versão recursiva, tenta-se recuperar a maior quantidade de sintagmas possível fazendo a análise também no restante da frase em que o melhor resultado foi obtido, no caso, 55.42% de palavras processadas e contidas nas regras implementadas. Assim, pode-se recuperar mais de 3 mil sintagmas nominais no corpus Kuramoto, 600 mil no CETENFolha e mais de 23 milhões no CETEMPúblico.

Uma análise com esses corpora relativa ao tempo de processamento comparado ao poder de reconhecimento de cada método implementado (SISNOP, SINOP-Janela e SISNOP-Recursivo) é apresentada mais adiante. A seguir, é avaliada uma alternativa ao sistema SISNOP, utilizando no módulo EM outro *software* como etiquetador morfológico.

5.4 Avaliação utilizando o *parser* PALAVRAS

O SISNOP é separado por módulos, sendo o Etiquetador Morfológico o primeiro deles e responsável por categorizar as palavras em classes gramaticais. Foi feita uma adaptação no sistema para também aceitar o *parser* PALAVRAS para realizar a mesma função. O conjunto de frases utilizado neste trabalho para testar o EM, retirado de Costa [Costa 2007], foi submetido ao PALAVRAS e não apresentou nenhum erro na classificação das palavras. O FORMA, por sua vez, dos 1241 itens encontrados, apresentou erro em 38 deles.

As frases etiquetadas pelo PALAVRAS foram submetidas ao SISNOP, e a comparação com o resultado do FORMA é exibida na Tabela 5.6. Dos 224 sintagmas nominais contidos no teste, 193 foram identificados com o uso do PALAVRAS, 8 a menos que o sistema usando o etiquetador FORMA. Do conjunto de sintagmas identificados corretamente, o PALAVRAS acertou 20 sintagmas a mais em relação ao FORMA. Pode-se perceber, portanto, que um etiquetador com

uma menor quantidade de erros pode aumentar o valor das métricas precisão, abrangência e, consequentemente, o F_1 em aproximadamente 8%.

Dados	**Sisnop-FORMA**	**Sisnop-PALAVRAS**	**Diferença**
Total de SN	224	224	-
Total de SN identificados	188	193	8
Total de SN ident. corretos	155	175	20
Precisão	82,45%	90,67%	8,22%
Abrangência	69,20%	78,12%	8,92%
F_1	75,24%	83,93%	8,66%

Tabela 5.6: Resultados comparativos entre FORMA e PALAVRAS como etiquetador morfológico do SISNOP

No Exemplo 5.16, a palavra *"pneu"* foi classificada erroneamente como pronome pessoal pelo FORMA, e de forma correta, como um substantivo, pelo PALAVRAS, o que ocasionou um erro na identificação da frase e dos sintagmas presentes nela. A frase do Exemplo 5.17 foi utilizada como exemplo de erro pelo FORMA. Ao ser classificada corretamente pelo PALAVRAS conseguiu-se recuperar o sintagma nominal *"um falso médico chinês"*.

"A minha bicicleta tem um pneu vermelho." (5.16)

"Chegou um falso médico chinês" (5.17)

O PALAVRAS tem algumas características que o diferem do FORMA, como a classificação das palavras *"todos os"* como uma única categoria, um pronome indefinido representado como *"todos=os"*. O FORMA, por sua vez, classifica separadamente como pronome indefinido e artigo, o que torna o analisador sintático por vezes mais complexo. O PALAVRAS classifica nomes próprios diferenciando-os dos substantivos comuns, além de recuperar as informações de gênero, número e grau.

Mesmo sendo um poderoso *parser* e não somente um etiquetador morfológico, não foi possível realizar maiores experimentos utilizando o PALAVRAS, ou até utilizá-lo como programa no módulo Etiquetador Morfológico do SISNOP, por não possuir uma licença para utilização deste *software*. Além disso, sua versão *online* aceita somente arquivos de tamanho de até 2Mb. É possível encontrar alguns corpus categorizados pelo PALAVRAS, como o CETENFolha e o CETEMPúblico, mas, para corpus maiores ainda não etiquetados, a utilização do PALAVRAS torna-se inviável.

5.5 Avaliação de tempo de processamento

Para obter eficácia no processamento de linguagem natural e ser possível utilizar em aplicações de recuperação de informação, por exemplo, os sistemas devem ter a habilidade de processar uma grande quantidade de documentos, sendo eficientes nas complexidades de tempo e espaço [Chengxiang, Evans e Zhai 1996].

A Tabela 5.7 mostra o tamanho de cada corpus utilizado no teste anterior, expresso pela quantidade de frases e de palavras, e os tempos de processamento para cada um dos métodos do SISNOP.

Corpora	**Dados do corpora**		**Tempo de processamento**		
	Qtd Frase	Qtd Palavras	SISNOP	SISNOP-Janela	SISNOP-Recursivo
Kuramoto	1.001	32.499	0.14s	2.90s	3.24s
CETENFolha	209.643	4mi	12.55s	204.817s	232.96s
CETEMPúblico	7mi	203mi	3m33s	90m23s	110m10s

Tabela 5.7: Resultados comparativos do tempo de processamento de cada método em diferentes corpora

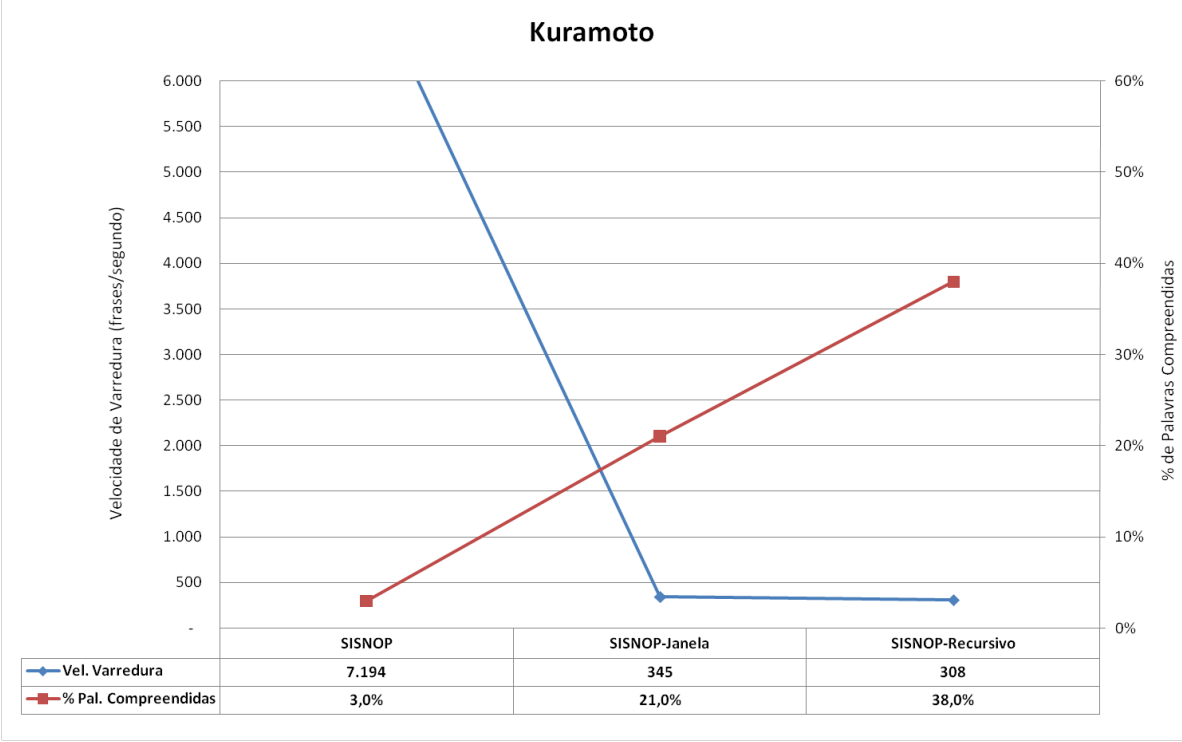


Figura 5.2: Gráfico comparativo entre os métodos, considerando velocidade de varredura e porcentagem de acertos, a partir do corpus Kuramoto

Como esperado, à medida que aumenta o tamanho do corpus, aumenta o tempo de processamento. Além disso, em todos corpora, o método recursivo consome mais tempo que os

demais. Mas a diferença entre o SISNOP-Janela e o SISNOP-Recursivo é pequena, devido à natureza semelhante dos dois de varrer cada frase diversas vezes com a janela para encontrar um encaixe e recuperar os sintagmas. Assim, os dois métodos têm tempo muito superior ao SISNOP sem aplicação da técnica de janela deslizante.

O corpus CETEMPúblico, que possui mais de 7 milhões de frases, gastou 110 minutos no SISNOP-Recursivo, 90 minutos no SISNOP-Janela e 3 minutos no SISNOP, rodando em paralelo com duas máquinas. Os outros dois conjuntos de testes não foram paralelizados, pois a diferença era mínima, já que a latência da rede e o tempo de comunição e envio de dados consomem tempo de processamento.

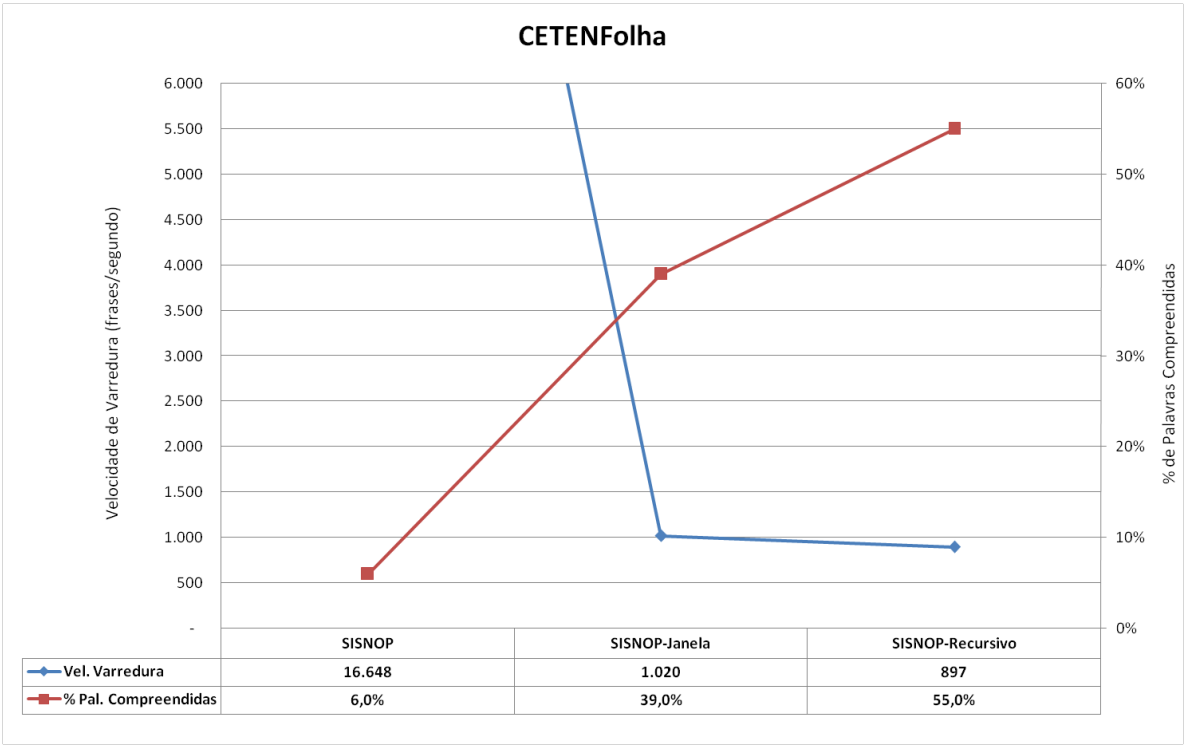


Figura 5.3: Gráfico comparativo entre os métodos, considerando velocidade de varredura e porcentagem de acertos, a partir do corpus CETENFolha

Ao processar corpora grandes como esse, busca-se alcançar um bom custo-benefício entre o tempo gasto de processamento e a porcentagem de palavras compreendidas pelo sistema. Para tentar avaliar melhor essa relação e facilitar uma possível escolha do método de acordo com a necessidade da aplicação, os dados foram disponibilizados em gráficos. Vale ressaltar que os testes efetuados anteriormente utilizaram o método SISNOP-Recursivo e que os valores dos gráficos a seguir foram arredondados para uma melhor análise gráfica.

A partir dos dados de tempo e do número de frases de cada corpus, foi calculada a velocidade de varredura aplicada em cada um dos métodos. A velocidade de varredura está explicitada

na unidade frases/segundo, significando, em um valor médio, quantas frases foram processadas por segundo. Além da velocidade, foi exibida a porcentagem de palavras processadas de cada método, considerando esse valor como sendo a quantidade de palavras contidas nas frases reconhecidas pelo programa. Assim, foi criado, para cada um dos corpus, um gráfico em que são expostas a velocidade de varredura e a porcentagem de palavras compreendidas para cada um dos três métodos desenvolvidos.

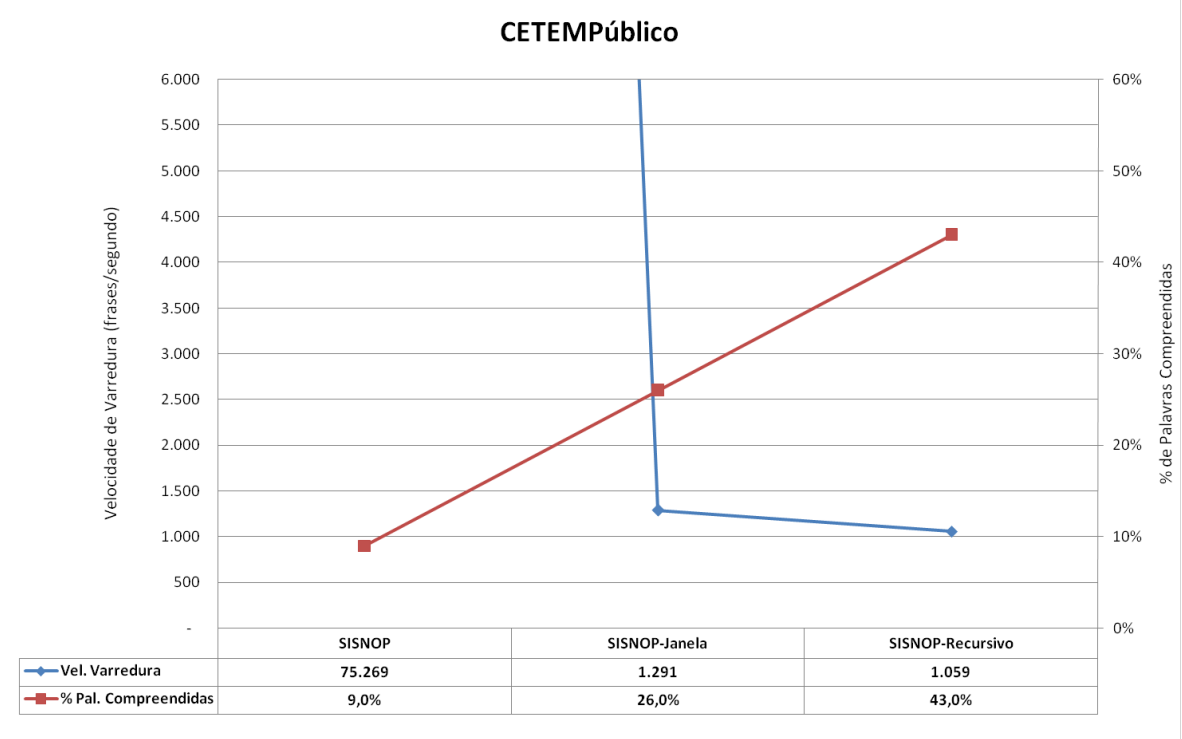
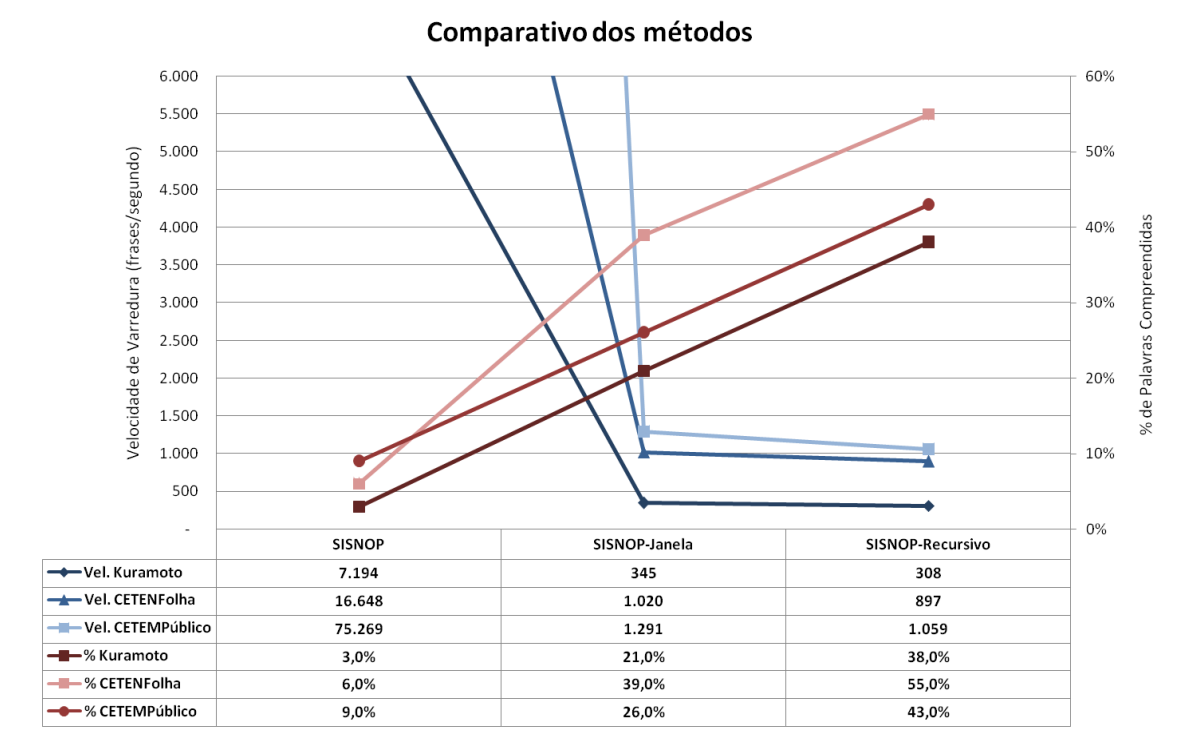


Figura 5.4: Gráfico comparativo entre os métodos, considerando velocidade de varredura e porcentagem de acertos, a partir do corpus CETEMPúblico

Os dados do corpus Kuramoto estão no Gráfico 5.2, que apresenta no lado esquerdo do eixo y os valores de velocidade de varredura e, no lado direito, a porcentagem de palavras compreendidas. O gráfico exibe, assim, duas curvas, uma comparando a velocidade com cada método e outra a porcentagem de palavras compreendidas. Assim, no Gráfico 5.2, a velocidade para o corpus Kuramoto, utilizando o SISNOP, é maior que 7 mil frases por segundo, valor esse muito maior que nos outros dois métodos. Por outro lado, a porcentagem de palavras compreendidas, 3%, é bem pequena em relação ao método SISNOP-Janela, com 21,0%, e SISNOP-Recursivo, com 38%.

O CETENFolha é um conjunto de dados bem maior que o Kuramoto. O Gráfico 5.3 mostra também as linhas de velocidade e porcentagem de palavras compreendidas. Nesse teste o método SISNOP-Janela obteve o seu melhor resultado, 39% de palavras compreendidas, mas, mesmo assim não conseguiu superar o método SISNOP-Recursivo, que obteve o valor de 55%,



	SISNOP	SISNOP-Janela	SISNOP-Recursivo
Vel. Kuramoto	7.194	345	308
Vel. CETENFolha	16.648	1.020	897
Vel. CETEMPúblico	75.269	1.291	1.059
% Kuramoto	3,0%	21,0%	38,0%
% CETENFolha	6,0%	39,0%	55,0%
% CETEMPúblico	9,0%	26,0%	43,0%

Figura 5.5: Gráfico comparativo entre os métodos, considerando os corpora, velocidade de varredura e porcentagem de acertos

sendo esse também o seu melhor valor em relação aos outros corpus.

As curvas mostradas no Gráfico 5.4 do corpus CETEMPúblico mostram a mesma tendência dos outros dois gráficos. O valor da velocidade de varredura do SISNOP é muito grande, mas o resultado é bem inferior aos outros dois métodos.

O Gráfico 5.5 mostra todas as linhas dos três corpora descritos anteriormente em um mesmo gráfico. Assim, pode-se comparar o comportamento dos três métodos nos corpora mostrados. As retas em tons de azul representam a velocidade de varredura e, as retas em tons de rosa, a porcentagem de palavras compreendidas. Percebe-se, então, sem maiores análises, que elas apresentam um mesmo comportamento: velocidade muito maior no método SISNOP, mas uma porcentagem de palavras compreendidas bem pequena. Por outro lado, os métodos SISNOP-Janela e SISNOP-Recursivo possuem uma velocidade menor, e bem próximas entre si, e uma porcentagem de palavras compreendidas bem maior que o SISNOP.

Pelos gráficos e pelos dados disponibilizados abaixo dele, pode-se observar que o método de janela deslizante aplicado em SISNOP-Janela e SISNOP-Recursivo melhora bastante a abrangência nas palavras do corpus e, consequentemente, dos sintagmas nominais recuperados. O seu uso, contudo, diminui a velocidade fazendo com que o tempo de processamento aumente com a utilização de corpora maiores.

Como a porcentagem de palavras compreendidas do SISNOP sem os métodos de janela foram muito baixas, pode-se afirmar que na maioria das aplicações será mais útil utilizar os outros dois métodos. Como eles, entre si, apresentam velocidades de varredura próximas, assim como valores de porcentagem em alguns casos não tão divergentes, fica a cargo do utilizador a escolha do melhor método a ser usado. Vale destacar também que diferenças de tempo podem ser reduzidas com a utilização de processamento paralelo, como foi feito para os testes aqui mostrados.

6 *Conclusões e trabalhos futuros*

"*É claro que meus filhos terão computadores, mas antes terão livros.*"
Bill Gates, fundador da Microsoft

Neste capítulo são apresentadas as conclusões deste trabalho e algumas propostas de trabalhos futuros.

Processar uma linguagem natural é permitir que os seres humanos comuniquem-se com os computadores da forma mais natural possível, utilizando a linguagem com a qual as pessoas estão acostumadas para se comunicar. O processamento da linguagem natural é assim, um campo de pesquisa instigante e desafiador. O objetivo do processamento de linguagem natural é fornecer aos computadores a capacidade de entender e compor textos escritos e falados. Entender um texto significa reconhecer o contexto, realizar as análises sintática, semântica, léxica e morfológica, criar resumos, extrair informação, interpretar os sentidos e até aprender conceitos com os textos processados.

Os sintagmas nominais são unidades de sentido com função sintática dentro de uma frase e representam interesse no processamento de linguagem natural devido ao grande valor semântico que possuem. Esta dissertação apresentou uma metodologia de identificação de sintagmas nominais do português. Mostrou também os métodos que trabalhos anteriores desenvolveram para extrair sintagmas nominais no inglês (*noun phrase*) e no português. Apresentou estudos da aplicação de sintagmas nominais em recuperação de informação (descritores de textos, índices, palavras-chave), na resolução de anáforas, na geração automática de ontologia e também em trabalhos utilizados na área médica.

A partir de um comparativo entre os conceitos de Perini e Liberato sobre a definição dos sintagmas nominais, em que o primeiro define sintagmas de forma sintática e o segundo de maneira mais semântica, foi proposto um método e desenvolvido um sistema que identifica sintagmas nominais com as informações de flexões de gênero, número e grau. O sistema é composto de três módulos, um responsável pela etiquetagem morfológica, outro que extrai os sintagmas das frases por meio de regras descritas em uma gramática e um terceiro que encontra as informações de gênero, número e grau de cada uma das palavras contidas no sintagma.

Os resultados permitiram observar bons valores de precisão e abrangência ao submeter um conjunto de 186 frases e analisá-lo manualmente. Ao testar conjuntos de dados maiores, como as coleções CETEMFolha com 209 mil frases e CETEMPúblico com 7 milhões, fez-se uma análise sobre a relação custo-benefício entre o tempo gasto de processamento e a porcentagem de palavras compreendidas pelo sistema. Pode-se também observar que o sistema pode ser usado para português de Portugal, já que o corpus CETEMPúblico possui grande parte dos seus textos nessa língua.

Dois métodos foram desenvolvidos com o intuito de melhorar o processo de análise das frases e consequentemente os resultados de identificação: janela deslizante e janela deslizante recursiva. Eles consistem em fazer uma varredura nas frases tentando encontrar uma regra gramatical implementada que se adeque à frase definida pela largura da janela. Esse método

foi de grande ganho para o sistema, pois fez os resultados de identificação para frases muito grandes e complexas, presentes na maioria dos textos em português, serem consideravelmente melhores. Como a análise sintática necessária para identificar os sintagmas nominais é muito custosa em termos de tempo de processamento, foi desenvolvida também a paralelização dessa tarefa.

Assim, com um conjunto de regras de pouca complexidade, com o método de janela deslizante e com um etiquetador morfológico, foi possível fazer identificação de sintagmas nominais em textos, independente do domínio, podendo ainda fazer uso de processamento paralelo para diminuir o tempo de gasto de execução do sistema para corpora de tamanhos maiores. Com isso, permitiu a utilização de sintagmas nominais em diferentes aplicações, com o intuito de prover melhor entendimento da linguagem natural por sistemas computacionais.

A metodologia proposta apresentou bons resultados, mas pode ser melhorada. O etiquetador morfológico FORMA, por exemplo, pode ser adaptado para corrigir os erros que foram listados neste trabalho. Como foi mostrado, com algumas poucas modificações no código, é possível trocar de etiquetador morfológico. Além disso, com a obrigação das novas regras ortográficas da língua portuguesa a partir de 2013, é interessante que o sistema se adeque para tal. Em relação às regras gramaticais, alguns casos não tratados podem ser adicionados e os existentes podem ser otimizados, evitando sempre a ambiguidade. O identificador de gênero, número e grau pode ter um número maior de casos nas listas de exceções, tornando-o mais específico.

Os métodos janela deslizante e janela deslizante recursivo podem ter um estudo mais aprofundado. Utilizando o princípio de que existem orações dentro de orações, eles conseguiram obter uma melhora muito boa nos resultados. O mesmo princípio pode ser aplicado em trabalhos futuros para identificar sintagmas dentro de sintagmas. O método recursivo, em especial, pode estabelecer condições à chamada de recursão, diminuindo assim o número de varredura nas frases. Além disso, podem ser realizados estudos sobre a variação da qualidade dos resultados, de acordo com o tamanho da janela, e comparativos com o tempo gasto de processamento.

Os códigos do sistema SISNOP, desenvolvido nesta dissertação, estão disponíveis sob licença GNU/GPL em [SISNOP]. Isso possibilita o desenvolvimento de uma versão *web* para o SISNOP, disponibilizado via navegador ou por *web-service*, facilitando assim a utilização por outros interessados no assunto. Pode-se também fazer um sistema auxiliar que receba arquivos em diferentes formatos, converta-os e depois os envie para o SISNOP, até mesmo com a possibilidade de enviar um *link* da internet e identificar sintagmas das páginas referentes a ele, facilitando o uso na geração de ontologias em web semântica.

Referências Bibliográficas

[Angheluta et al. 2004]ANGHELUTA, R. et al. Clustering algorithms for noun phrase coreference resolution. In: PURNELLE, G.; FAIRON, C.; DISTER, A. (Ed.). *Le Poids Des Mots. Actes Des 7émes Journées Internationales D'analyse Statistique Des Données Textuelles*. [S.l.]: Presses Universitaires UCL, 2004. p. 60–70. 28

[Azeredo 1990]AZEREDO, J. C. *Iniciação a sintaxe do português*. Rio de Janeiro: Letras, 1990. 47

[Baeza-Yates e Ribeiro-Neto 1999]BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern Information Retrieval*. 1st. ed. [S.l.]: Addison Wesley, 1999. Paperback. ISBN 020139829X. 20

[Bastos 2007]BASTOS, E. C. *SISNBT - Sistema Identificador de Sintagmas Nominais Baseado em Tagger*. [S.l.], Julho 2007. 25

[Bechara 2009]BECHARA, E. *Moderna Gramca Portuguesa*. 1. ed. [S.l.]: Lucerna, 2009. 40

[Bender e Osório 2003]BENDER, T. C.; OSÓRIO, F. S. Reconhecimento e recuperaç ao de imagens utilizando redes neurais artificiais do tipo mlp. *Anais do IV ENIA - Encontro Nacional de Inteligência Artificial*, Campinas - SP, v. 1, p. 1–10, 2003. 62

[Bennett et al. 2004]BENNETT, N. A. et al. Extracting noun phrases for all of medline. *canis*, 2004. 31

[Bick 2000]BICK, E. *The Parsing System PALAVRAS: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework.* Tese (Doutorado) — Arthus University, Arthus, 2000. 71

[Bison 2009]BISON. 2009. Disponível em: <http://www.gnu.org/software/bison/>. 55

[Brants 2003]BRANTS, T. Natural language processing in information retrieval. In: *CLIN - Computational Linguistics in the Netherlands*. University of Antwerp - Belgium: [s.n.], 2003. 26

[Cardie e Pierce 1998]CARDIE, C.; PIERCE, D. Error-driven pruning of treebank grammars for base noun phrase identification. In: *Proceedings of the 17th international conference on Computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 1998. p. 218–224. 21

[Chengxiang, Evans e Zhai 1996]CHENGXIANG, D. E.; EVANS, D. A.; ZHAI, C. Noun-phrase analysis in unrestricted text for information retrieval. In: *In Proceedings of ACL*. [S.l.: s.n.], 1996. p. 17–24. 26, 87

[Chomsky 1956]CHOMSKY, N. Three models for the description of language. *IRE Transactions on Information Theory*, v. 2, p. 113–124, 1956. 42

[CLARIT]CLARIT. Disponível em: <http://www.clarit.com/tk$_o$*verview.htm*>. 31

[Costa 2007]COSTA, F. *Deep Linguistic Processing of Portuguese Noun Phrases*. Dissertação (Mestrado) — Department of Informatics, University of Lisbon, November 2007. DI/FCUL TR-07-34. 25, 75, 78, 85, 104

[Duque 2006]DUQUE, C. G. Uma abordagem ontológica para a indexação de documentos eletrônicos através da utilização de linguística computacional. In: *XI Simpósio Nacional de Letras e Linguística e I Simpósio Internacional de de Letras e Linguística*. [S.l.: s.n.], 2006. 30

[Freitas 2005]FREITAS, S. A. A. *Interpretação Automatizada de Textos: Processamento de anáforas*. Tese (Doutorado) — Universidade Federal do Espírito Santo, Vitória, Novembro 2005. 14, 29

[Gonzalez, Lima e Lima 2006]GONZALEZ, M. A. I.; LIMA, V. L. S. de; LIMA, J. V. de. Tools for nominalization: An alternative for lexical normalization. In: VIEIRA, R. et al. (Ed.). *PROPOR*. [S.l.]: Springer, 2006. (Lecture Notes in Computer Science, v. 3960), p. 100–109. ISBN 3-540-34045-9. 52, 75

[Hopcroft Rajeev Motwani 2002]HOPCROFT RAJEEV MOTWANI, J. D. U. J. E. *Introduç ao ´a teoria de autômatos, linguagens e computação*. [S.l.]: Campus, 2002. 53

[Huang et al. 2005]HUANG, Y. et al. Improved identification of noun phrases in clinical radiology reports using a high-performance statistical natural language parser augmented with the umls specialist lexicon. *Journal of the American Medical Informatics Association*, v. 12, n. 3, May/Jun 2005. 31

[Jones e Willet 1997]JONES, K. S.; WILLET, P. (Ed.). *Readings in Information Retrieval*. [S.l.]: Morgan Kaufmann Publishersand Inc., San Francisco, California, 1997. 66

[Kuramoto 1995]KURAMOTO, H. Uma abordagem alternativa para o tratamento e a recupera de informa textual : os sintagmas nominais. *Ciência da Informação*, v. 25, n. 2, 1995. 22

[Kuramoto 1999]KURAMOTO, H. *Uma abordagem alternativa para o tratamento e a recuperação de informação textual: os sintagmas nominais.* Tese (Doutorado) — Universidade Lumiere, Lyon, Fran 1999. 22, 23, 24, 83

[Kuramoto 2002]KURAMOTO, H. Sintagmas nominais: uma nova proposta para a recuperação de informação. *DataGramaZero - Revista de Ciência da Informação*, v. 3, n. 1, Fevereiro 2002. 23

[Lee, Lin e Chen 2001]LEE, C.-H.; LIN, C.-R.; CHEN, M.-S. Sliding-window filtering: an efficient algorithm for incremental mining. In: *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*. New York, NY, USA: ACM, 2001. p. 263–270. ISBN 1-58113-436-3. 62

[Liberato 1997]LIBERATO, Y. G. *A estrutura do SN em português - Doutorado em Letras - estudos lingísticos*. Tese (Doutorado) — UFMG, Minas Gerais, 1997. 15, 44, 45, 46, 48

[Lopes et al. 2009]LOPES, L. et al. Extraçautomática de termos compostos para construção de ontologias: um experimento na área da saúde. In: . Rio de Janeiro: Revista Eletrônica de Comunicação, Informaçõ e Inovaçõ em Saúde, 2009. p. 60–70. 30

[Louden]LOUDEN, K. C. *Compiladores: principios e praticas*. [S.l.: s.n.]. 42

[Miorelli 2001]MIORELLI, S. T. *ED-CER: Extração do Sintagma Nominal em Sentenças em Português*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul, Julho 2001. , 21, 23, 44, 46, 47

[Moraes e Lima 2007]MORAES, S. M. W.; LIMA, V. L. S. de. Um estudo sobre categorização hierárquica de uma grande coleção de textos em língua portuguesa. In: *V TIL - Workshop em Tecnologia da Informação e da Linguagem Humana*. [S.l.: s.n.], 2007. 52

[Moraes e Lima 2008]MORAES, S. M. W.; LIMA, V. L. S. de. Abordagem nupervisionada para extração de conceitos a partir de textos. In: *VI TIL - Workshop em Tecnologia da Informação e da Linguagem Humana*. [S.l.: s.n.], 2008. 27, 52

[Morellato 2007]MORELLATO, L. V. *SIDSN - Sistema Identificador de Sintagmas Nominais*. [S.l.], Julho 2007. 24, 25

[Moura et al. 2008]MOURA, M. F. et al. *Uma Abordagem Completa para a Construção de Taxonomias de Tópicos em um Domínio*. Sarlos - SP, 2008. Disponível em: <http://www.icmc.usp.br/b̃iblio/BIBLIOTECA/rel_tec/RT_329.pdf>. 52

[Nicola 2008]NICOLA, J. de. *Português - Ensino Médio*. 1. ed. Saulo: Scipione, 2008. 13, 34, 35

[Oliveira e Freitas 2006]OLIVEIRA, C.; FREITAS, M. C. de. Um modelo de sintagma nominal lexical na recuperação de informações. In: *XI Simpósio Nacional e I Simpósio Internacional de Letras e Linguística (XI SILEL)*. [S.l.: s.n.], 2006. p. 778–786. 27

[Oliveira et al. 2006]OLIVEIRA, C. et al. A set of np-extraction rules for portuguese: Defining, learning and pruning. In: VIEIRA, R. et al. (Ed.). *PROPOR*. [S.l.]: Springer, 2006. (Lecture Notes in Computer Science, v. 3960), p. 150–159. ISBN 3-540-34045-9. 24

[Oliveira e Quental 2003]OLIVEIRA, C. M. G. M. de; QUENTAL, V. de S. T. D. B. Aplicações do processamento automático de linguagem natural na recuperação de informações. *Congresso Internacional da ABRALIN*, Anais do III ABRALIN, Rio de Janeiro, RJ, Brasil, p. 949–955, 2003. 13

[Orengo e Huyck 2001]ORENGO, V. M.; HUYCK, C. A Stemming Algorithm for Portuguese Language. In: *Proc. of Eigth Symposium on String Processing and Information Retrieval (SPIRE 2001) - Chile*. [S.l.: s.n.], 2001. p. 186–193. 66

[Penn Treebank Wall Street Journal]PENN Treebank Wall Street Journal. Disponível em: <http://www.cis.upenn.edu/ treebank/>. 21

[Pereira 2009]PEREIRA, F. S. do C. *Uma Metodologia para a utilização do processamento de Linguagem Natural na busca de informações em documentos digitais*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, Vitória, agosto 2009. 14, 29

[Pereira, Morellato e Freitas 2009]PEREIRA, F. S. do C.; MORELLATO, L.; FREITAS, S. A. A. de. Evaluation of an information retrieval model based in anaphora resolution. In: *IADIS International Conference WWW/Internet*. Rome, Italy: [s.n.], 2009. 29

[Pereira, Seibel Júnior e Freitas 2009]PEREIRA, F. S. do C.; Seibel Júnior, H.; FREITAS, S. A. A. de. An anaphora based information retrieval model extension. In: *CSIE*. Los Angeles, LA, USA: [s.n.], 2009. 29

[Perini 1995]PERINI, M. A. *Gramática Descritiva do Português*. [S.l.]: Ática, 1995. 23, 46

[Perini 2003]PERINI, M. A. *Gramática Descritiva do Português*. 4. ed. Saulo: Ática, 2003. 15, 44, 45, 46, 47, 48, 64

[Porter 1980]PORTER, M. F. An algorithm for suffix stripping. *Program*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, v. 14, n. 3, p. 130–137, 1980. Disponível em: <http://portal.acm.org/citation.cfm?id=275705>. 66

[Protégé]PROTÉGÉ. Disponível em: <http://protege.stanford.edu/>. 30

[Ricarte 2008]RICARTE, I. *Introdução à compilação*. 1. ed. [S.l.]: Campus-Elsevier, 2008. 43

[Rich e Knight 1993]RICH, E.; KNIGHT, K. *Inteligência Articial*. [S.l.]: Makron, 1993. 39

[Rocha e Santos 2000]ROCHA, P. A.; SANTOS, D. Cetempúblico: Um corpus de grandes dimenses de linguagem jornalística portuguesa. In: *V Encontro para o processamento computacional da língua portuguesa escrita e falada (PROPOR'2000)*. [S.l.]: Graunes, 2000. p. 131–140. 84

[Santos 2005]SANTOS, C. N. dos. *Aprendizado de máquina na identificação de sintagmas nominais: o caso do português brasileiro*. Dissertação (Mestrado) — Instituto Militar de Engenharia, 2005. 24, 84

[Santos e Rocha 2001]SANTOS, D.; ROCHA, P. Evaluating cetempúblico, a free resource for portuguese. In: *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2001. p. 450–457. 84

[Seibel Júnior 2007]Seibel Júnior, H. *Recuperação de informações relevantes em documentos digitais baseada na resolução de anáforas*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, Vitória, julho 2007. 14, 29

[SISNOP]SISNOP. Disponível em: <http://inf.ufes.br/ lvmorellato/sisnop>. 94

[Soon et al. 2001]SOON, W. M. et al. *A Machine Learning Approach to Coreference Resolution of Noun Phrases*. 2001. 28

[Souza 2005]SOUZA, R. R. *Uma proposta de metodologia para escolha automática de descritores utilizando sintagmas nominais.* Tese (Doutorado) — Escola de Ciência da Informação, Belo Horizonte, MG, 2005. 23, 24, 44, 47, 83

[Spackman e Hersh 1996]SPACKMAN, K. A.; HERSH, W. R. Recognizing noun phrases in medical discharge summaries: An evaluation of two natural language parsers. In: . Portland, USA: Biomedical Information Communication Center Oregon Health Sciences University, 1996. 31

[Tanenbaum 2003]TANENBAUM, A. S. *Redes de Computadores*. trad. 4 ed. Rio de Janeiro: Elsevier, 2003. 62

[Vieira et al. 2000]VIEIRA, R. et al. Extração de sintagmas nominais para o processamento de co-referência. In: *V Encontro para o processamento computacional da língua portuguesa escrita e falada (PROPOR'2000)*. [S.l.]: Graunes, 2000. p. 131–140. 27

[Vieira et al. 2006]VIEIRA, R. et al. (Ed.). *Computational Processing of the Portuguese Language, 7th International Workshop, PROPOR 2006, Itatiaia, Brazil, May 13-17, 2006, Proceedings*, v. 3960 de *Lecture Notes in Computer Science*, (Lecture Notes in Computer Science, v. 3960). [S.l.]: Springer, 2006. ISBN 3-540-34045-9.

[Voutilainen 1995]VOUTILAINEN, A. Nptool, a detector of english noun phrases. *CoRR*, cmp-lg/9502010, 1995. 21, 31

[Wilkinson e Allen 1999]WILKINSON, B.; ALLEN, M. *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*. [S.l.: s.n.], 1999. 69

ANEXO A – Regras gramaticais de identificação dos sintagmas

A seguir são apresentadas as regras que definem um sintagma nominal como combinação de classes gramaticais e de palavras específicas ("qualquer um", "cada um", "mesmo", "próprio"e "outro"). A regra nome_composto utilizada abaixo representa um nome (substantivo) ou uma composição de nomes (como no exemplo: "República Federativa").

- PRONOME_PESSOAL
- ARTIGO ADJETIVO nome_composto ADJETIVO
- ARTIGO ADJETIVO nome_composto
- ARTIGO nome_composto ADJETIVO
- ARTIGO nome_composto
- ADJETIVO nome_composto ADJETIVO
- ADJETIVO nome_composto
- nome_composto ADJETIVO
- nome_composto
- ARTIGO ADJETIVO nome_composto ADJETIVO ADVERBIO
- ARTIGO ADJETIVO nome_composto ADVERBIO
- ARTIGO nome_composto ADJETIVO ADVERBIO
- ARTIGO nome_composto ADVERBIO
- ADJETIVO nome_composto ADJETIVO ADVERBIO

- ADJETIVO nome_composto ADVERBIO
- nome_composto ADJETIVO ADVERBIO
- nome_composto ADVERBIO
- ARTIGO ADJETIVO nome_composto ADJETIVO PREPOSICAO ADVERBIO
- ARTIGO ADJETIVO nome_composto PREPOSICAO ADVERBIO
- ARTIGO nome_composto ADJETIVO PREPOSICAO ADVERBIO
- ARTIGO nome_composto ADVERBIO
- ADJETIVO nome_composto ADJETIVO PREPOSICAO ADVERBIO
- ADJETIVO nome_composto PREPOSICAO ADVERBIO
- nome_composto ADJETIVO PREPOSICAO ADVERBIO
- nome_composto PREPOSICAO ADVERBIO
- ARTIGO ADJETIVO nome_composto ADJETIVO PREPOSICAO ADVERBIO ADVERBIO
- ARTIGO ADJETIVO nome_composto PREPOSICAO ADVERBIO ADVERBIO
- ARTIGO nome_composto ADJETIVO PREPOSICAO ADVERBIO ADVERBIO
- ARTIGO nome_composto ADVERBIO
- ADJETIVO nome_composto ADJETIVO PREPOSICAO ADVERBIO ADVERBIO
- ADJETIVO nome_composto PREPOSICAO ADVERBIO ADVERBIO
- nome_composto ADJETIVO PREPOSICAO ADVERBIO ADVERBIO
- nome_composto PREPOSICAO ADVERBIO ADVERBIO
- ARTIGO PRONOME_POSSESSIVO ADJETIVO nome_composto ADJETIVO
- ARTIGO PRONOME_POSSESSIVO ADJETIVO nome_composto
- ARTIGO PRONOME_POSSESSIVO nome_composto ADJETIVO
- ARTIGO PRONOME_POSSESSIVO nome_composto

- ARTIGO "mesmo"ADJETIVO nome_composto ADJETIVO
- ARTIGO "mesmo"ADJETIVO nome_composto
- ARTIGO "mesmo"nome_composto ADJETIVO
- ARTIGO "mesmo"nome_composto
- ARTIGO "próprio"ADJETIVO nome_composto ADJETIVO
- ARTIGO "próprio"ADJETIVO nome_composto
- ARTIGO "próprio"nome_composto ADJETIVO
- ARTIGO "próprio"nome_composto
- ARTIGO "outro"ADJETIVO nome_composto ADJETIVO
- ARTIGO "outro"ADJETIVO nome_composto
- ARTIGO "outro"nome_composto ADJETIVO
- ARTIGO "outro"nome_composto
- PRONOME_POSSESSIVO ADJETIVO nome_composto ADJETIVO
- PRONOME_POSSESSIVO ADJETIVO nome_composto
- PRONOME_POSSESSIVO nome_composto ADJETIVO
- PRONOME_POSSESSIVO nome_composto
- "mesmo"ADJETIVO nome_composto ADJETIVO
- "mesmo"ADJETIVO nome_composto
- "mesmo"nome_composto ADJETIVO
- "mesmo"nome_composto
- "próprio"ADJETIVO nome_composto ADJETIVO
- "próprio"ADJETIVO nome_composto
- "próprio"nome_composto ADJETIVO
- "próprio"nome_composto

- "outro"ADJETIVO nome_composto ADJETIVO
- "outro"ADJETIVO nome_composto
- "outro"nome_composto ADJETIVO
- "outro"nome_composto
- PRONOME_INDEFINIDO ADJETIVO nome_composto ADJETIVO
- PRONOME_INDEFINIDO ADJETIVO nome_composto
- PRONOME_INDEFINIDO nome_composto ADJETIVO
- PRONOME_INDEFINIDO nome_composto
- "qualquer um"
- "cada um"
- "qualquer"ADJETIVO nome_composto ADJETIVO
- "qualquer"ADJETIVO nome_composto
- "qualquer"nome_composto ADJETIVO
- "qualquer"nome_composto
- "qualquer outro"ADJETIVO nome_composto ADJETIVO
- "qualquer outro"ADJETIVO nome_composto
- "qualquer outro"nome_composto ADJETIVO
- "qualquer outro"nome_composto
- PRONOME_DEMONSTRATIVO ADJETIVO nome_composto ADJETIVO
- PRONOME_DEMONSTRATIVO ADJETIVO nome_composto
- PRONOME_DEMONSTRATIVO nome_composto ADJETIVO
- PRONOME_DEMONSTRATIVO nome_composto
- ADJETIVO

ANEXO B – Conjunto de teste

Um conjunto de testes composto por 186 frases, sendo 38 delas falso-positivas, extraído da dissertação de Costa [Costa 2007] que o utilizou para avaliação da LXGram, uma gramática computacional para o processamento linguístico profundo do português, desenvolvida na Universidade de Lisboa. Os sintagmas nominais estão limitados pelo símbolo colchetes e o núcleo destacado em negrito.

** Frases incorretas gramaticalmente

1.[**Eles**] avariaram.

2.[Esses **carros**] avariaram.

3.[Os meus **carros**] avariaram.

4.[Aqueles meus dois **carros**] avariaram.

5.Chegou [um falso **médico** chinês].

6.Chegou [um falso **médico**] que é [**chinês**].

7.[O meu **carro**] está [na **oficina**].

8.[Todos os **homens**] são [**mortais**].

9.Chegou [a tua **encomenda**].

10.Chegou [uma **encomenda** tua].

11.Chegaram [as duas **encomendas**].

12.Chegaram [duas **encomendas**].

13.[O primeiro **lugar**] está [**vago**].

14.[Todos os **homens**] leram [um certo **livro**].

15.[Todos os **homens**] batem [num pobre **burro**].

16.[Todos os **homens**] batem [num **burro** cinzento].

17.[O pai do **Rui**] chegou ontem.

18.Era [um **cão**] [com três **pernas**].

19.Os que chegarem primeiro esperam.

20.Não existem [com **cinco**].

21.Isso sai [com **benzina**].

22.Isso [com **benzina**] sai.

23.Era [um **chapéu**] [com uma **antena**].

24.** Isso era um com uma antena chapéu.

25.São [os quatro **naipes**].

26.Muitas [espécies de **sapos** da Amazônia] já estão [**extintas**].

27.Bastantes [espécies de **sapos** da Amazônia] já estão [**extintas**].

28.As muitas [espécies de **sapos** da Amazônia] já estão [**extintas**].

29.** As bastantes espécies de sapos da Amazônia já estão extintas.

30.[Todas as **pessoas**] leram [um certo **livro**].

31.[Todas as **pessoas**] leram [um **livro**].

32.Foi [a **invasão** americana] [do **Iraque**].

33.[Os primeiros dois **filmes**] passaram aqui.

34.[Os dois primeiros **filmes**] passaram aqui.

35.[Os **adeptos**] sentiram [**entusiasmo**] depois de [duas grandes **vitórias**] [do **clube**].

36.** Os adeptos sentiram entusiasmo depois de grandes duas vitórias do clube.

37.[Os **seres humanos**] são [**livres**].

38.[Todos os **seres humanos**] são [**livres**].

39.[Todas as **pessoas**] são [**livres**].

40.[Todas aquelas **pessoas**] são [**livres**].

41.[Todas **pessoas**] são [**livres**].

42.[As **pessoas** todas] são [**livres**].

43.Chegou [um falso **médico** chinês].

44.Atacaram [um mero **inspetor**].

45.** Atacaram um inspetor mero.

46.** Atacaram um japonês inspetor.

47.Atacaram [um **inspetor** japonês].

48.Atacaram [um falso **inspetor**].

49.Atacaram [um **inspetor** falso].

50.Era [um **grande**], [grande **filme**].

51.Era [um **filme** chato], [**chato**].

52.Viram [a **alunagem** americana] [na **televisão**].

53.Viram [a **invasão** americana] [do **Iraque**].

54.** Viram a invasão do Iraque americana.

55.Viram [a **alunagem** americana] [de **1969**].

56.** Viram a alunagem de 1969 americana.

57.** Viram a invasão americana iraquiana.

58.Viram o [**consumo** galopante] [de **petróleo**].

59.Viram [o **consumo de petróleo** galopante].

60.Viram [o **consumo de petróleo**] que continua a crescer.

61.[A minha **bicicleta**] tem [um **pneu** vermelho].

62.[Uma **bicicleta** minha] tem [um **pneu** vermelho].

63.[Aquela tua **bicicleta**] tem [um **pneu** vermelho].

64.[Aquela **bicicleta** tua] tem [um **pneu** vermelho].

65.[**Ele**] é teu [**irmão**]?

66.[As minhas duas **bicicletas**] estão aqui.

67.** Minhas as duas bicicletas estão aqui.

68.** As duas minhas bicicletas estão aqui.

69.[**Ele**] é [**pianista**].

70.** Ele viu pianista.

71.[Minha **bicicleta**] tem [um **pneu** vermelho].

72.[O **irmão da Ana**] está aqui.

73.[O seu **irmão**] está aqui.

74.** O seu seu irmão está aqui.

75.[Os meus dois **irmãos**] estão aqui.

76.[O teu **livro**] está aqui.

77.[Os primeiros dois **capítulos**] são [**cômicos**].

78.[Os dois primeiros **capítulos**] são [**cômicos**].

79.[Um certo primeiro **capítulo**] é [**cômico**].

80.** Um primeiro certo capítulo é cômico.

81.[Dois certos **capítulos**] são [**cômicos**].

82.[Certos dois **capítulos**] são [**cômicos**].

83.** Os dois três carros avariaram.

84.** O segundo primeiro lugar está vago.

85.[Um certo **carro**] avariou.

86.[Um determinado **carro**] avariou.

87.** Um determinado certo carro avariou.

88.** Os dois primeiros três lugares estão vagos.

89.** Os primeiros dois segundos pratos estão aqui.

90.** Certos dois certos carros avariaram.

91.[Os vários **participantes**] passeiam [as **folhas**] [pela **sala**].

92.** Os vários vinte participantes estão aqui.

93.** Os vinte vários participantes estão aqui.

94.[Os vários primeiros **lugares**] estão aqui.

95.[Os primeiros vários **lugares**] estão aqui.

96.Viu [vários certos **participantes**].

97.Viu certos [vários **participantes**].

98.** Viu os vários vários participantes.

99.** Viu os vários vinte vários participantes.

100.** Verá os próximos primeiros capítulos.

101.** Verá os primeiros próximos capítulos.

102.Verá [os três próximos **capítulos**].

103.Verá [os próximos três **capítulos**].

104.Verá [os dois melhores **capítulos**].

105.** Verá os melhores dois capítulos.

106.[Todas as **pessoas**] leram [um certo **livro**].

107.[Todas as **pessoas**] leram [um **livro**].

108.** Todos os determinados homens leram um livro.

109.** Os determinados homens leram um livro.

110.** Esses determinados homens leram um livro.

111.[Todos os **filhos**] [da **Ana**] leram [um certo **livro**].

112.Estão aqui [dois certos **capítulos**].

113.Estão aqui [certos dois **capítulos**].

114.Está aqui [um **DVD**] com [dois primeiros **episódios**] [dessa **série**].

115.** Está aqui um DVD com primeiros dois episódios dessa série.

116.[Algumas **cartas**] chegaram.

117.[Duas **cartas**] chegaram.

118.[**João**] não viu [uma **mancha**] [no **chão**].

119.[**João**] não viu [**manchas**] [no **chão**].

120.[Todas as pessoas] leram [um **livro**] [sobre **girafas**].

121.[Todas as **pessoas**] leram [**livros**] [sobre **girafas**].

122.[**Pedro**] quer encontrar [um **policial**].

123.[**Pedro**] quer encontrar [**policiais**].

124.[**João**] não viu [duas **manchas**] [no **chão**].

125.[O **João**] não viu [certa **mancha**] [no **chão**].

126.[Todas as **pessoas**] leram [certo **livro**] [sobre **girafas**].

127.[**Pedro**] quer encontrar [certo **polícia**].

128.[Aquele **carro**] ali estava aqui ontem.

129.** Aquele ali carro estava aqui ontem.

130.Vi [**carros**] sem [**assentos** vermelhos].

131.Vi [os dois **carros**] [da **Ana**].

132.Saíram com [a **Ana**].

133.Chegou [um falso **médico**] que é [**Chinês**].

134.Todos [os exatamente três **filmes**] que lá vi eram [**maus**].

135.[A **bicicleta**] essa é [**verde**].

136.Chegaram várias [**cartas** tuas].

137.Desapareceram [as **cartas** todas].

138.** Uma bicicleta essa é verde.

139.** Essa bicicleta essa é verde.

140.** Esta bicicleta essa é verde.

141.** A bicicleta essa essa é verde.

142.[Esta **bicicleta**] aqui é [**verde**].

143.[Essa **bicicleta**] aí é [**verde**].

144.[Aquela **bicicleta**] ali é [**verde**].

145.[O **carro**] esse é [**verde**].

146.[Esse **carro**] é [**verde**].

147.[Todos esses **carros**] avariaram.

148.Vi [a **casa** azul] e [a **verde**].

149.Vi algumas [**crianças**] com [**chapéus**] e algumas com [**bonés**].

150.Vi [os **pobres**].

151.Vi [os **dois**].

152.Vi [os **sem abrigo**].

153.Os que podem ajudar nunca ajudam.

154.Vi [os **homens** bastante velhos] e [os especialmente **novos**]. ««

155.Vi alguns.

156.Vi os seus [**dois**].

157.Vi a [**verde**].

158.Vi alguns [**jovens**] com [**chapéus**].

159.Comprei [**maçãs**].

160.Todas estavam [**podres**].

161.Todos são [**livres**].

162.[As **pessoas**] chegaram.

163.[Aquelas **pessoas**] chegaram.

164.[Todas as **pessoas**] chegaram.

165.[Todas aquelas **pessoas**] chegaram.

166.[A minha **irmã**] está aqui.

167.[Uma **irmã** minha] está aqui.

168.[A minha **bicicleta**] está aqui.

169.[Os dois **carros**] avariaram.

170.[Dois **carros**] avariaram.

171.[Os dois primeiros **capítulos**] estão aqui.

172.[Os primeiros dois **capítulos**] estão aqui.

173.[Certos dois **carros**] avariaram.

174.[Dois certos **carros**] avariaram.

175.[Todos os **homens**] leram [um **livro**].

176.[Todos os **homens**] leram [um certo **livro**].

177.[A **irmã** mais velha do Rui] chegou.

178.[A **irmã** do Rui mais velha] chegou.

179.Mora [numa **casa**] com [**janelas** azuis].

180.** Mora numa com janelas azuis casa.

181.[As duas grandes **guerras**] que abalaram [o **mundo**] foram [**más**].

182.[O **filme**] esse é [**mau**].

183.[A minha **bicicleta**] é [**azul**].

184.** Uma minha bicicleta é azul.

185.Alguns eram [extremamente **secos**].

186.[Os muito **ricos**] sempre abusaram [dos muito **pobres**].