Fernando Medeiros do Nascimento

Como Treinar seu Parser: Classificação Sintática automatizada de textos em português

Salvador

Brasil

2019

Fernando Medeiros do Nascimento

Como Treinar seu Parser: Classificação Sintática automatizada de textos em português

Projeto de trabalho de conclusão do curso de Bacharelado em Ciência da Computação, da Universidade Federal da Bahia.

Universidade Federal da Bahia – UFBA

Instituto de Matemática e Estatística

Departamento de Ciência da Computação

Orientador: Prof. Dr. Marlo Vieira dos Santos e Souza

Salvador
Brasil
2019

Dedico esse trabalho a todos que já pensaram em desistir de tudo e recomeçar. Vocês provavelmente tinham razão de pensar assim. Se tiver oportunidade, jogue tudo pra cima.

Agradecimentos

Este trabalho merecia a menção de um incontável número de pessoas. O que é impossível. Meus agradecimentos, então, vão para:

Minha mãe. Obrigado por todo o amor. Eu vou fazer todo sacrificio valer a pena.

Minha irmã e meu pai. Por todo o crescimento, obrigado.

Minha companheira. Companheira de emoções, aventuras, e vida. Viver é mais fácil com você.

Onix e Mago. Obrigado pela paciência.

Meus amigos. Vocês não me ajudaram em nada (risos), mas este trabalho seria impossível sem vocês.

Meu orientador. Obrigado por não desistir (até quando eu já tinha desistido).

A UFBA. Resistir é preciso, se renovar também.

"Eu amo prazos. Adoro prazos. Adoro o barulho de vento que eles fazem quando os dias vão passando"

Douglas Adams

Resumo

Classificadores sintáticos automatizados (também conhecidos como *parsers*) são um recurso computacional estudado na área de Processamento de Linguagem Natural. São dispositivos capazes de realizar a classificação morfossintática de sentenças escritas em linguagem natural. Apesar de serem estudados há bastante tempo, são poucos os *parsers* disponíveis desenvolvidos para o processamento da língua portuguesa.

Existem diversos métodos de *parsing* baseados em regras pré-definidas na literatura. Porém, atualmente os mais investigados se baseiam em métodos estatísticos. Tais métodos necessitam de um conjunto de dados de entrada pré-classificados para serem treinados. Estes conjuntos de dados são chamados de bancos de árvores (*treebanks*).

Dada a existência prévia de *treebanks* próprios para o processamento da língua portuguesa; e *parsers* com boa performance, mas que não estão adaptados para esta mesma língua; propõe-se, então, o treinamento de um *parser* conhecido, desenvolvido com foco na língua inglesa, com dados de treino da língua portuguesa. Para tal, será realizada a *transdução* dos dados nos seus respectivos formatos originais para o formato aceito pelo *parser* escolhido, sem perda de informação lexical.

Palavras-chaves: *parsers*, classificador sintático, Língua Portuguesa, transdução.

Abstract

Automated syntactic classifiers (also known as parsers) are a computational resource studied in the area of Natural Language Processing. They are devices capable of performing the morphosyntactic classification of sentences written in natural language. Despite being studied for a long time, there are few available parsers developed for the Portuguese language processing.

There are several parsing methods based on predefined rules in the literature. However, currently, the most investigated methods are based on statistical methods. Such methods require a set of pre-classified input data to be trained. These data sets are called treebanks.

Given the specific treebanks previous existence for processing the Portuguese language; and parsers with good performance, but that are not adapted to the same language, it is proposed then, the training of a known parser, developed with a focus on the English language, with training data from the Portuguese language. To this end, the data will be transduced in their respective original formats to the format accepted by the chosen parser, without loss of lexical information.

Key-words: parsers, syntactic classifier, Portuguese language, transduction.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

CD	*Corpus* de Destino. *Corpora* cujas estruturas serão reproduzidas no processo de transdução
CETEM	*Corpus de Extractos de Textos Electrónicos NILC*, Corpus criados pelo projeto Linguateca. Possui as variantes CETEMFolha e CETEMPublico.
CFG	*Context Free Grammars*, Gramáticas Livres de Contexto
CLI	*Command Line Interface*, *softwares* que permitem a interação por meio do terminal de comandos do sistema.
CO	*Corpus* de Origem. *Corpora* que passarão pelo processo de transdução para se assemelharem a algum CD.
FAQ	*Frequently Asked Questions*, perguntas mais frequentes
IA	Inteligência Artificial
PCFG	*Probabilistic Context Free Grammars*, Gramáticas Livres de Contexto Probabilísticas
PLN	Processamento de Linguagem Natural
POS	*Part of Speech*, Parte do Discurso
PTB	*Penn TreeBank*
SP	*Stanford Parser*
UD	*Universal Dependencies*, projeto para criar *tagsets* unificados para *treebanks* de diversas línguas

Sumário

1 Introdução

"O mundo é mediado pela linguagem" (OLIVEIRA, 2019) , no sentido que, interações humanas são realizadas a partir da linguagem, ou interpretadas com base nela. Pode-se pensar, então, em situações nas quais a linguagem "não é" diretamente usada, como ao chutar uma pedra. Porém, toda a situação é interpretada se valendo da linguagem para fazê-lo.

Fanon (2008, p 33) afirma: "Falar é estar em condições de empregar uma certa sintaxe, possuir a morfologia de tal ou qual língua, mas é sobretudo assumir uma cultura, suportar o peso de uma civilização [...]. Um homem que possui a linguagem possui, em contrapartida, o mundo que essa linguagem expressa e que lhe é implícito". Dominar a língua, ser capaz de interpretá-la e articulá-la, dá ao indivíduo uma série de possibilidades. Não só o poder de influência, mas o poder do reconhecimento como pessoa, (*Ibid., p 33*) "Uma vez que falar é existir absolutamente para o outro".

Desde a origem da computação, existe o desejo em fazer com que computadores sejam capazes de processar linguagem. Quando TURING (1950, p 433), nos pede que consideremos a questão "Podem as máquinas pensar?" [1] , a forma sugerida para que o famoso Jogo da Imitação seja operado é, essencialmente, uma troca de mensagens, ou seja, o uso livre da linguagem.

A interpretação da linguagem por meios automáticos é uma das várias áreas que a Inteligência Artificial (I.A.) estuda, chamada de PLN (Processamento de Linguagem Natural, ou NLP em inglês). Rodrigues (2017) resume:

> "O objetivo do PLN é fornecer aos computadores a capacidade de entender e compor textos. 'Entender' um texto significa reconhecer o contexto, fazer análise sintática, semântica, léxica e morfológica, criar resumos, extrair informação, interpretar os sentidos, analisar sentimentos e até aprender conceitos com os textos processados."

PLN é muito estudada como ciência de base, ou até mesmo por empresas para a criação de produtos como *chatbots* e Assistentes Pessoais, para citar exemplos cotidianos.

Uma das formas de utilizar a linguagem computacionalmente é a Classificação Sintática Automatizada. Como definido por Charniak (1997, p 33), "Classificação sintática automatizada é o processo de atribuir um marcador de sintagma a uma sentença" [2] . Ou

[1] "*Can machines think?*". Tradução própria.

[2] No original: "*Syntactic parsing is the process of assigning a phrase marker to a sentence*". Tradução própria.

seja, realizar a análise sintática de sentenças como "João ganhou a bola" é marcar, por exemplo, a palavra "João" como um substantivo, "ganhou" como verbo etc. Classificação automatizada diz respeito à construção de sistemas computacionais que realizem tal tarefa.

Ainda se pesquisam métodos baseados em lógica para realizar tal tarefa. Porém, a tendência atual são os métodos estatísticos. Como definido em Charniak (1997, p 37),

> "Classificadores estatísticos funcionam atribuindo probabilidades para possíveis classificações (árvores) de uma sentença, localizando a árvore mais provável, e então apresentando tal árvore como resposta. Também, para construir um *parser* estatístico, deve-se descobrir como (1) encontrar possíveis árvores, (2) atribuir probabilidades para elas, e (3) devolver a mais provável." [3]

Para que tenham bom funcionamento, *parsers* precisam ser treinados com um conjunto de sentenças pré-classificadas. A essas sentenças damos o nome de *árvores*, e ao seu conjunto, o nome de Banco de Árvores, ou *treebanks* (como será melhor explicado em 2.4).

Ao leitor pode parecer contra-intuitivo, utilizar estatística ao invés de regras pré-definidas. Nas palavras do próprio Charniak, em (MOOR, 2006, p 89):

> "Estatística dominou o processamento de linguagem natural porque funciona" [4]

Uma busca simples no Google Acadêmico[5] por "*parser*" nos retorna aproximadamente 320.000 resultados. Pesquisar por "*natural language processing*" retornará aproximadamente 3.290.000. *Parsers* são quase 10% deste total.

Fazendo um processo semelhante, buscando por "*parser portuguese*", obtemos aproximadamente 10.400 resultados. Quase 3%. Já "classificador sintático português" retorna 15.100 resultados. A pesquisa por "*parser english*", porém, retorna 106.000 resultados. O triplo de resultados em comparação com o verbete anterior.

É compreensível que o número de pesquisas na área para a língua inglesa seja maior do que para a língua portuguesa. É importante que se diga, também, que existem *treebanks* robustos para o português. Sente-se falta da conexão entre ambos, *parsers* e dados disponíveis. Pode-se criar a hipótese, portanto, de que mesmo com dados e métodos disponíveis, o desenvolvimento de *parsers* não é trivial. Isso explicaria a tendência (natural)

[3] No original: "*Statistical parsers work by assigning probabilities to possible parses of a sentence, locating the most probable parse, and then presenting the parse as the answer. Thus, to construct a statistical parser, one must figure out how to (1) find possible parses, (2) assign probabilities to them, and (3) pull out the most probable one*". Tradução própria.

[4] "*Statistics has taken over natural language processing because it works*". Tradução própria.

[5] scholar.google.com.br

dos pesquisadores de dedicarem maior esforço para os métodos já desenvolvidos, além de focar na língua com maior expressão global.

Dada a complexidade da tarefa, surge a inquietação que motiva este trabalho: Seria possível desenvolver um *parser out-of-the-box*, ou seja, tentando utilizar apenas materiais já desenvolvidos (a saber, *parsers* já existentes, bem como corpus de treino já existentes), de modo que ele seja eficiente?

Sobre *out-of-the-box parser*, seguimos a linha de Silva et al. (2010, p 2), que se propõe a estender pacotes de *software* já disponíveis, que permitam treinar um *parser* robusto a partir de um *treebank*, que seja independente de linguagem e suporte uma aplicação para o português sem grandes problemas.

Os primeiros testes empíricos foram um fracasso. Os *corpus* disponíveis não eram recebidos pelos *parsers* localizados neste estudo. Levantou-se, então, uma nova possibilidade: realizar a transdução dos dados de entrada.

"Transdutores" foram explicados por Mohri (2004, p 1):

> "[Transdutores] São autômatos os quais cada transição, em adição à sua etiqueta de entrada normal, é aumentado com uma etiqueta de saída de um possível novo alfabeto, e carrega algum elemento de peso de um semianel. Transdutores podem ser usados para definir um mapeamento entre dois tipos diferentes de fontes de informação, por exemplo palavras e sequências de fenômenos" [6]

Em suma, neste trabalho faremos a transdução de conjuntos de dados, construídos num certo formato, para um novo formato, e faremos a avaliação deste experimento. Isto nos leva a um segundo produto: o desenvolvimento de uma metodologia de adaptação inter-corpora.

1.1 Contexto do Trabalho

O estudo de *parsers* é bastante conhecido do meio acadêmico, tendo muito uso tanto na NLP, como na compilação de linguagens de programação. Também, existe uma produção crescente para o desenvolvimento de tecnologias em NLP para o português, principalmente para as variantes europeia e brasileira. Foi sentida, porém, a falta de mais estudos focados na língua portuguesa.

Pesquisas prévias mostraram que existem, sim, *parsers* projetados pensando na língua portuguesa (serão mais abordados na sessão 2.4.2). Porém, eram de difícil acesso, ou

[6] "*[Transductors] are automata in which each transition in addition to its usual input label is augmented with an output label from a possibly new alphabet, and carries some weight element of a semiring. Transducers can be used to define a mapping between two different types of information sources, e.g., word and phoneme sequences*". Tradução própria.

não era possível utilizá-los. Foram encontrados, também, bancos de dados[7] com informações em língua portuguesa, variantes europeia e brasileira.

Sabendo-se da existência de tecnologias tanto de *parsing* como de textos em português pré-classificados, pensou-se na possibilidade do desenvolvimento *Out-of-the-box*, adaptando tecnologias pré-existentes, e avaliando seus resultados. Não apenas desejando verificar a eficiência de tal método, como analisar quanto estudo técnico é necessário para tal.

Notou-se, contudo, que tal processo de adaptação exige um método ele mesmo. Pois os *parsers* exigem um formato de entrada, que nem sempre é o mesmo formato dos dados disponíveis. Portanto, foi desenvolvido um método de conversão entre dados de formatos distintos, de modo a que seja possível o uso destes dados no analisador sintático escolhido.

1.2 Objetivo Geral

Este trabalho tem como objetivo o treinamento de *parsers* já existentes para que sejam capazes de processar a língua portuguesa. A partir de tal treino, avaliar a sua eficiência, taxa de erros e afins.

1.3 Objetivos Específicos

- Estudar sobre analisadores sintáticos (*parsers*);
- Fazer o levantamento de florestas sintáticas (*treebanks*) disponíveis na língua portuguesa;
- Fazer o levantamento de *parsers* em português;
- Definir um *parser* convencional a ser utilizado na pesquisa;
- Desenvolver um método de adaptação entre florestas sintáticas;
- Realizar a adaptação de florestas sintáticas para utilizar no *parser* definido;
- Treinar o *parser* definido com as florestas sintáticas adaptadas;
- Avaliar resultados do *parser*;

1.4 Justificativa

Como já citado, a quantidade de pesquisas em *parsers* para a língua portuguesa ainda estão em número reduzido com relação aos estudos mundiais. Por observação empírica,

[7] Bancos de Árvores (*treebanks*, que serão melhor explicados em 2.2)

nota-se que estão muito focados na reprodução e manutenção de materiais já existentes, e antigos. Como visto em (MANNING; SCHÜTZE, 1999, p 371):

> "Uma avaliação completa de classificadores como pré-processadores úteis para tarefas de NLP multilíngues de alto nível só serão possíveis após resultados experimentais suficientes de uma ampla gama de línguas estiver disponível" [8]

1.5 Metodologia

O desenvolvimento deste trabalho foi dividido em 6 etapas.

A Primeira etapa consistiu na revisão bibliográfica. Fez-se necessário o estudo dos classificadores sintáticos, e das florestas sintáticas.

Para o estudo dos classificadores sintáticos, pesquisou-se no Google Acadêmico[9] por *parsing*, *statistical parsing*, *constituency parsing*, *Neural Networks and parsing*, *Treebank*, *parser comparison*. Foram coletados 10 artigos por assunto pesquisado, que tiveram seu resumo/*abstract* lidos, para avaliação de relevância. Como material base de estudo, foi usado (MANNING; SCHÜTZE, 1999). Será melhor abordado em 2.4.

Para a pesquisa de florestas sintáticas para o português, pesquisou-se pelas palavras-chave *portuguese treebank*, *treebank português*, *floresta sintática português* em motores de busca como Google Acadêmico e Google[10]. Encontramos o Bosque (FREITAS; ROCHA; BICK, 2008), do projeto Floresta Sintá(c)tica disponível publicamente, e nos foi cedido o CINTIL (BRANCO et al., 2011). Será explicado em 2.2.

De forma análoga, pesquisou-se por *parsers* na língua portuguesa nos motores de busca supracitados, com as palavras-chave *portuguese parser*, *parser português*, *classificador sintático português*. Encontramos referências ao PALAVRAS (BICK, 2000) e ao LX-Parser (NLX-GRUPO DE FALA E LINGUAGEM NATURAL, 2010). Abordados em 2.4.2.

Na Segunda etapa, buscou-se por um *parser* a ser utilizado no projeto. Utilizando-se os motores de busca supracitados, pesquisou-se por *parser*, *parsing*, *constituency parser*. Devido à sua robustez, fácil utilização sem necessidade de desenvolvimento computacional (uma vez que é possível o uso por comandos de terminal), e por ter amplo reconhecimento na comunidade, optou-se pelo uso do Stanford Parser (SP) (CHEN; MANNING, 2014). Este recebe, como entrada, dados no formato Penn Treebank (PTB) (MARCUS; MARCINKIEWICZ; SANTORINI, 1993). O SP será melhor descrito em 2.4.1.

---

[8] No original: "*A full evaluation of taggers as useful preprocessors for high-level multilingual NLP tasks will only be possible after sufficient experimental results from a wide range of languages are available*". Tradução própria.

[9] <http://www.scholar.google.com>

[10] <www.google.com>

Percebeu-se, então, a necessidade de realizar uma adaptação entre bancos de árvores, iniciando-se a Terceira etapa, de referencial teórico para a transdução.

Para a adaptação entre bancos de árvores, estudou-se a estrutura dos bancos originais, por observação de exemplos. De acordo com as necessidades observadas, estudamos os manuais de cada um, que conste: O Manual de *bracketing* do Penn Treebann (BIES et al., 1995), Manual de classificação do Penn Treebank (MARCUS; MARCINKIEWICZ; SANTORINI, 1993), Manual da Bíblia Florestal (FREITAS; AFONSO, 2007) e Manual de LxParser (NLX-GRUPO DE FALA E LINGUAGEM NATURAL, 2010). Com a necessidade de aprofundamento, estudou-se também o manual do formato Árvores Deitadas, utilizado no BOSQUE (FREITAS, 2006) e o Manual do Cintil (BRANCO et al., 2011). Neste contexto, foi desenvolvida a metodologia de transdução de *parsers*, que será abordada neste trabalho.

Na Quarta etapa, foi feita a transdução de bancos de árvores propriamente dita. O *transdutor* foi desenvolvido na linguagem Python, pela afinidade do autor com a mesma. O desenvolvimento teve como objetivo realizar a transdução de modo a conservar a informação lexical dos dados originais, adaptando apenas a sua estrutura para que possam ser consumidos pelo supra-citado Stanford Parser. Como dito anteriormente, este *parser* recebe como entrada árvores no formato Penn Treebank (PTB). Portanto, os bancos de árvores selecionados na etapa Um foram transduzidos para o formato PTB. A discussão aprofundada do método de transdução será realizada no Capítulo 3.

A Quinta etapa envolveu o treino do Stanford Parser propriamente dito. Em nenhum momento houve implementação sobre o *parser*. Foi realizado apenas o processo de transdução, treinamento e avaliação. Para o uso do *parser*, foi utilizada a sua interface de terminal (*command line interface*, CLI). O treino foi feito utilizando o método *10-fold validation*, para possibilitar uma média de avaliação. Os procedimentos de treino serão comentados nas seções 3.2 e 3.3.

Por fim, na Sexta etapa foi feito o cruzamento dos dados obtidos, suas análises e considerações, que serão demonstradas na seção 4.

Parte I

Referenciais teóricos

2 Revisão de Literatura

Primeiramente, apresentamos conceitos linguísticos na seção 2.1. Em seguida, na seção 2.2, estudamos os *treebanks*: o que são, suas estruturas, e os *treebanks* utilizados neste trabalho, tanto os que serão transduzidos, como o *treebank* de referência. Em 2.3, explicamos o conceito por trás dos classificadores morfossintáticos. Em 2.4, mostramos *parsers* na prática: seu funcionamento, o *parser* que utilizaremos para o estudo, alguns *parsers* da língua portuguesa. Por fim, em 2.5, será explanada a medida de avaliação que será utilizada neste trabalho

2.1 Revisão de conceitos linguísticos

Para o bom entendimento do trabalho, é necessário o domínio de algumas características linguísticas que serão trabalhadas em seu desenvolvimento.

2.1.1 Análise de Constituinte e Sintagma

Em qualquer linguagem, palavras não são ditas ao acaso. Existe uma ordem, e regras, para que elas sejam bem emitidas, e bem recebidas. Um conceito fundamental sobre tal ordenação é a ideia que palavras se agrupam de forma hierárquica. Esta estrutura pode ser visualizada na Figura 1,

Para entender esta hierarquia, é essencial entender o conceito de sintagma.

O **sintagma** (ou *phrase*, em inglês) é, de acordo com Castilho (2010, p 55),

> "a quarta unidade gramatical na hierarquia descritivista [1]. Trata-se de uma associação de palavras articuladas à volta de cinco dentre elas: o verbo, o substantivo, o adjetivo, o advérbio e a preposição. [...] A classe de palavras que nucleariza o sintagma dá-lhe o nome, e assim teremos o sintagma nominal (SN), o sintagma verbal (SV), o sintagma adjetival (SAdj), o sintagma adverbial (SAdv) e o sintagma preposicionado (SP)." [2]

[1] Os anteriores são: o Fonema, a Sílaba, o Morfema

[2] Este trabalho baseou-se no *tagset* do *Penn Treebank*, que será melhor explicado na seção 2.2.1. Salvo exceções, serão utilizados neste trabalho os rótulos para sintagmas (*phrase tag*) equivalentes ao do Penn. Que conste: *noun phrase* (NP), *verbal phrase* (VP), *adjective phrase* (ADJP), *adverb phrase* (ADVP), *prepositional phrase* (PP).

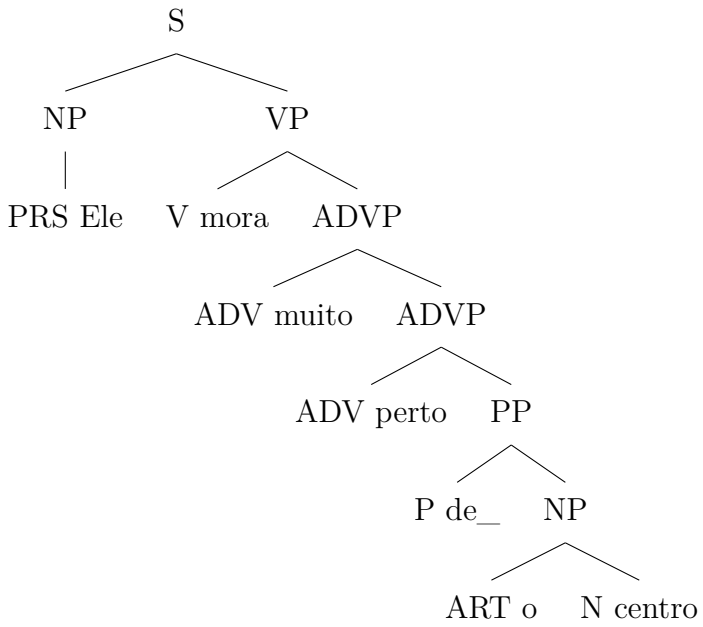


Figura 1 – Exemplo de árvore de Constituência. Adaptado da sentença aTSTS-002/80, do CINTIL

E, "Os sintagmas exemplificam a propriedade de 'constituência', isto é, a capacidade linguística de organizar expressões dotadas de uma margem esquerda, um núcleo e uma margem direita" (*Ibid., p 55*).

(RUDER, 2019) destaca que "Análise de Constituência visa extrair uma árvore baseada em constituência de uma sentença que represente sua estrutura sintática de acordo com uma gramática de estrutura frasal" [3], ou seja, uma árvore de constituintes (ou árvore de constituência - *constituency parse*). Esta árvore demonstra a relação hierárquica da sentença e dos seus elementos. Como destacam Manning e Schütze (1999, p 93):

> "Uma ideia fundamental é que certos agrupamentos de palavras se comportam como constituintes. Constituintes podem ser detectados por serem capazes de ocorrer em várias posições, e mostrar possibilidades sintáticas uniformes para expansão." [4]

Sobre a mudança de posição, podemos exemplificar como:

- Eu fui no mercado comprar maçãs
- Comprar maçãs, no mercado, eu fui

[3] No original: "*Constituency parsing aims to extract a constituency-based parse tree from a sentence that represents its syntactic structure according to a phrase structure grammar*". Tradução própria.

[4] No original: "*One fundamental idea is that certain groupings of words behave as constituents. Constituents can be detected by their being able to occur in various positions, and showing uniform syntactic possibilities for expansion*". Tradução própria.

- Eu fui comprar maçãs no mercado

Sobre a expansão de um constituinte, seria como na Tabela 1:

Ele	**COMPROU**	isso
Ontem ele	**COMPROU**	maçã
Ele foi ontem e	**COMPROU**	uma maçã
Ele foi na loja e	**COMPROU**	uma torta de maçã

Tabela 1 – Exemplo de expansão de constituinte

Elementos que podem ser substituídos por outro numa posição sintática apresentam um *relacionamento paradigmático*. Duas palavras que podem formar um sintagma possuem *relação sintagmática*.

2.1.2 Estrutura Frasal

Manning e Schütze (1999, p 93) dizem "Sintaxe é o estudo das regularidades e restrições da ordem das palavras e estrutura frasal" [5]. Isso é importante, pois as palavras não estão posicionadas aleatoriamente numa frase. Sua ordem, sua morfologia, tudo influencia para o significado que ela expressa.

Já foi abordado em 2.1.1 o conceito de constituintes, que é fundamental para esse estudo. Será feita agora uma breve revisão sobre estruturas.

Gramáticas como a inglesa tem a estrutura mais estática, ou seja, as palavras têm posições bem definidas, que implicam no seu significado. No exemplo citado por Manning e Schütze (1999, p 85),

- *Mary gave Peter a book*
- *Peter gave Mary a book*

O posicionamento informa exatamente quem deu o livro a quem. A ordem das palavras determina a categoria da frase. No caso:

- Forma base: Sujeito - Verbo - Objeto
  - [The children][should][eat spinach]
- Interrogativo: Verbo - Sujeito - Objeto

[5] "*Syntax is the study of the regularities and constraints of word order and phrase structure*". Tradução própria.

  - [Should][the children][eat spinach]?

- Imperativo: Verbo - Objeto

  - [Eat][spinach]!

Já línguas como o Latim ou o Russo podem posicionar as palavras em qualquer posição, sem perda de significado. São chamadas "Free Word Order Language" [6]. Tais linguagens utilizam outros marcadores para definir o significado, e utilizam a ordem para indicar estrutura de discurso. Exemplo:

- Pedro deu o livro a Maria
- Maria deu o livro a Pedro
- A Maria, Pedro deu o Livro

Para reproduzir o padrão de sentenças, podemos nos valer das regras de reescrita. (MANNING; SCHÜTZE, 1999, p 96) "Uma regra de reescrita tem a forma 'categoria → categoria*' e declara que o símbolo do lado esquerdo pode ser reescrito com a sequência de símbolos no lado direito" [7]. Ou seja, para quem já está familiarizado com o estudo de teoria dos autômatos, seriam as regras de produção ou reescrita. Produção é definida em (HOPCROFT; MOTWANI; ULLMAN, 2003, p 171) como:

- "Uma variável que está sendo (parcialmente) definida pela produção. Esta variável é geralmente chamada 'núcleo' da produção." [8]
- "O símbolo da produção →." [9]
- "Uma cadeia de caracteres de zero ou mais terminais e variáveis. Essa cadeia, chamada o corpo da produção, representa o modo de formar cadeias na linguagem da variável do núcleo." [10]

As produções dependem apenas da variável a ser reescrita (no nosso caso, a categoria sintática). Portanto, temos uma Gramática Livre de Contexto (GLC, ou CFG em Inglês).

[6] "Linguagem de livre ordem das palavras". Tradução própria

[7] No original: "*A rewrite rule has the form 'category → category*' and states that the symbol on the left side can be rewritten as the sequence of symbols on the right side*". Tradução própria.

[8] No original: "*A variable that is being (partially) defined by the production. This variable is often called the head of the production.*" Tradução própria.

[9] No original: "*The production symbol →*". Tradução própria.

[10] "*A string of zero or more terminals and variables. This string, called the body of the production, represents one way to form strings in the language of the variable of the head*". Tradução própria.

2.1.3 Etiquetas Morfosintáticas - Part of Speech Tags

Sintaticamente falando, palavras podem ser agrupadas em classes (ou grupos) que demonstrem seu comportamento sintático. Tais classes podem ser chamadas de Categorias Sintáticas, ou Categorias Gramaticais. Até mesmo Classes. Exemplos de classes são *verbo*, *substantivo*, *adjetivo*, *artigo* etc. Outra forma de se referir a tais classes, é chamando-as de "*Part of Speech*"[11] (POS).

(MANNING; SCHÜTZE, 1999, p 81) "O teste mais básico para palavras pertencentes à mesma classe é o teste de substituição" [12]. Podemos ver um exemplo na Figura 2.

$$\text{A} \left\{ \begin{array}{c} \text{casa} \\ \text{pessoa} \\ \text{mulher} \\ \text{criança} \\ \text{Bolsa de Valores} \\ \text{pena} \end{array} \right\} \text{caiu.}$$

Figura 2 – Teste de Substituição

Na Figura 2, demonstramos o teste da substituição para substantivos. Note-se que palavras podem ter mais de uma POS. Por exemplo, *casa* pode ser um substantivo (NN), ou a terceira pessoa do singular do verbo "casar" (VBZ [13]). Sistemas que fazem análise sintática de sentenças atribuem um rótulo a cada palavra que represente a sua classe gramatical. Esta marca é chamada de Etiqueta Morfossintática (ou *POS tag*, no inglês).

Palavras podem ser de duas categorias principais, citadas por Manning e Schütze (1999, p 82):

> "As categorias abertas, ou lexicais, são aquelas como substantivos, verbos e adjetivos, que tem um grande número de membros, e nos quais novas palavras são comumente adicionadas. As fechadas, ou funcionais, são categorias como preposições e artigos (contendo palavras como *de, em, o, um*) que tem apenas poucos membros, e cujos membros normalmente tem um claro uso gramatical" [14]

[11] Parte do Discurso

[12] No original: "*The most basic test for words belonging to the same class is the substitution test*". Tradução própria.

[13] Essa é uma tag do Penn Treebank, usada de maneira direta. Em 2.2.1 explicamos melhor este *treebank*, e suas possíveis traduções no Capítulo 3.

[14] No original: "*The open or lexical categories are ones like nouns, verbs and adjectives which have a large number of members, and to which new words are commonly added. The closed or functional categories*

Podemos complementar essa ideia com a Tabela 2, apresentada por Castilho (2010, p 55):

Abertas	Fechadas
Substantivos	Preposições
Verbos	Conjunções
Adjetivos	Determinantes
Interjeições	Pronomes
Advérbios	Quantificadores

Tabela 2 – Classes de Palavras no Português. Adaptada de Castilho (2010, p 55)

Algumas POS tags serão descritas com mais detalhes quando forem abordados o Penn Treebank (2.2.1), e *treebanks* para o Português (2.2.2).

2.2 *Treebanks*

Em Processamento de Linguagem Natural, não é possível avaliar textos em seu contexto original o tempo todo. Ao invés disso, faz-se uma coleção de textos, que servirão como amostra para a análise. Esse corpo de textos é chamado de *corpus*, e um conjunto de corpus é chamado *corpora*, como destacado em (MANNING; SCHÜTZE, 1999, p 6):

> "Adotando tal abordagem baseada em corpus, pessoas apontaram para as primeiras defesas das ideias empiricistas pelo linguista britânico J. R. Flirth, que forjou o slogan 'Você deve conhecer uma palavra pela companhia que esta mantém'". [15]

Em 1951, Zelling Harris tenta descobrir procedimentos no qual a estrutura de uma linguagem possa ser encontrado automaticamente. Como destacado em (*Ibid., p 6*):

> "Enquanto este trabalho não pensava numa implementação computacional, e é de certa forma ingênuo computacionalmente, encontramos aqui também a ideia de que uma boa descrição gramatical é uma que provenha de uma representação compacta de um corpus de textos" [16].

are categories such as prepositions and determiners (containing words like of, on, the, a) which have only a few members, and the members of which normally have a clear grammatical use". Tradução própria.

[15] No original: "*Adopting such a corpus-based approach, people have pointed to the earlier advocacy of empiricist ideas by the British linguist J.R. Firth, who coined the slogan 'You shall know a word by the company it keeps'*". Tradução própria.

[16] No original: "*While this work had no thoughts to computer implementation, and is perhaps somewhat computationally naive, we find here also the idea that a good grammatical description is one that provides a compact representation of a corpus of texts*". Tradução própria.

Diversas técnicas de *parsing* utilizam de aprendizado supervisionado. Portanto, precisamos de uma fonte de dados que sirva para o treino e para o teste destes sistemas. No nosso caso, usamos bancos de árvores, ou *treebanks*. *Treebanks* são, como descrito em (MANNING; SCHÜTZE, 1999, p 412), "Alguns exemplos dos tipo de análise em árvore desejados. Uma coleção de tais árvores de exemplo são denominados *treebank*" [17].

Freitas, Rocha e Bick (2008, p 142) resumem como:

> "Uma floresta sintática [18] – tradução do inglês *treebank* – é um conjunto de itens (frases) analisados sintaticamente. A cada frase é atribuída uma estrutura sintática hierárquica, e por isso uma frase (sintaticamente analisada) pode ser vista como uma árvore, donde uma floresta nada mais é que um conjunto de frases analisadas sintaticamente e com informação relativa aos níveis de constituintes. Florestas sintáticas costumam ser utilizadas, de maneira geral, tanto em estudos da língua baseados em corpus como no treino de analisadores sintáticos".

Ou seja, *treebanks* são *corpora* de sentenças pré-analisadas sintaticamente, de maneira automatizada, semi-automatizada, ou cuja análise foi totalmente feita por humanos.

Uma grande quantidade de grupos criou seus próprios *treebanks* ao longo da história. Manning e Schütze (1999, p 412) destacam que o mais utilizado, refletindo seu tamanho (robustez) e legibilidade, é o *Penn Treebank*.

2.2.1 *Penn Treebank*

Vários *treebanks* foram construídos ao longo da história [19]. Dentre eles, um que recebeu destaque foi o Brown Corpus. (MARCUS; MARCINKIEWICZ; SANTORINI, 1993, p 314):

> "O conjunto de etiquetas morfossintáticas usados para anotar grandes corpora no passado são, tradicionalmente, bem extensivos. O pioneiro, Brown Corpus, distingue 87 etiquetas simples permite a formatação de tags compostas." [20]

A ideia do *Penn Treebank* (PTB) é, indo na contramão, fazer um *corpus* com o *tagset* simplificado. *Tagset* é o conjunto de etiquetas, ou seja, as marcações morfossintáticas

[17] No original: "*some examples of the kinds of parse trees that are wanted. A collection of such example parses is referred to as a treebank*". Tradução própria.

[18] Ao longo do trabalho serão utilizados os termos "banco de árvores", ou "treebanks".

[19] Marcus, Marcinkiewicz e Santorini (1993, p 314) citam, por exemplo, Lancaster-Oslo / Bergen, o Lancaster UCREL, e o London-Lund Corpus of Spoken English.

[20] No original: "*The POS tagsets used to annotate large corpora in the past have traditionally been fairly extensive. The pioneering Brown Corpus distinguishes 87 simple tags and allows the formation of compound tags*". Tradução própria.

que serão aplicadas a palavras e estruturas. Algumas estratégias foram tomadas para que essa redução fosse possível, uma vez que, é preciso lembrar, a extensa quantidade de *tags* tem razão para acontecer. É uma forma de alcançar o que é descrito em Garside, Sampson e Leech (1988 apud MARCUS; MARCINKIEWICZ; SANTORINI, 1993, p 314), "O ideal de prover códigos distintos para todas as classes de palavras possuindo comportamentos gramaticais distintos" [21]. Uma das primeiras estratégias citadas foi reduzir a redundância de *tags* cuja distinção pode ser obtida pelo léxico da palavra. Por exemplo, (*Ibid., p 314*) "O Brown Corpus [...] distingue tres formas de 'do' - a forma base (DO), o tempo passado (DOD), e a terceira pessoa do presente do singular (DOZ)" [22]. Todas estas diferenças podem ser capturadas lexicalmente no momento da análise.

Além da recuperabilidade lexical, também houve a eliminação de *POS tags* cuja distinção é recuperável com referência à estrutura sintática. Como dito em (*Ibid., p 315*):

> "Por exemplo, o conjunto de etiquetas do Penn Treebank não distingue pronomes sujeitos de pronomes objeto mesmo em casos onde a distinção não é recuperável pela forma pronominal, tal como '*you*', uma vez que a distinção é recuperável na base a posição do pronome na árvore de derivação na versão classificada do corpus" [23]

Tomar essa medida não só minimiza redundâncias, como também aumenta a consistência do corpus, uma vez que um número reduzido de *tags* reduz a possibilidade de erros / inconsistências.

O PTB foca, também, em marcar a palavra de acordo com sua característica sintática. Por exemplo, a palavra *One*. (MARCUS; MARCINKIEWICZ; SANTORINI, 1993, p 315-316):

> "Por exemplo, no sintagma 'the one', 'one' sempre é marcado como CD (número cardinal), enquanto que no sintagma plural correspondente 'the ones', 'ones' é sempre marcado como NNS (substantivo comum plural), independente da função paralela de 'one' e 'ones' como núcleos do sintagma nominal.// Por contraste, [...] nós codificamos uma função sintática de palavra no seu *POS tag* sempre que possível. Portanto, '*one*' é marcado como NN (substantivo comum singular) ao invés de CD [...] quando é núcleo de um sintagma nominal." [24]

[21] No original: "*the ideal of providing distinct codings for all classes of words having distinct grammatical behaviour*". Tradução própria.

[22] No original: "*The Brown Corpus [...] distinguishes three forms of do–the base form (DO), the past tense (DOD), and the third person singular present (DOZ)*". Tradução própria.

[23] No original: "*For instance, the Penn Treebank tagset does not distinguish subject pronouns from object pronouns even in cases where the distinction is not recoverable from the pronoun's form, as with you, since the distinction is recoverable on the basis of the pronoun's position in the parse tree in the parsed version of the corpus*". Tradução própria.

[24] No original: "*For instance, in the phrase the one, one is always tagged as CD (cardinal number),*

Um diferencial entre o PTB e boa parte dos *treebanks* existentes é a questão da indeterminação. Quando há tanto a ambiguidade de *POS* no texto, como incerteza do [25]. Em diversos momentos, o contexto linguístico é capaz de resolver tais diferenças. Nem sempre é possível atribuir uma única *tag* a uma palavra. Para resolver isto, o PTB possibilita ao anotador que atribua mais de uma *tag* a uma palavra, se necessário. Existe a liberdade de atribuir quantas *tags* forem necessárias, "mas na prática, multiplas tags se restringem a um pequeno número de combinações de duas *tags* recorrentes" [26], como visto em (MARCUS; MARCINKIEWICZ; SANTORINI, 1993). Para fazer a anotação do PTB, foi usado um processo em duas partes: primeiro, automatizado, e depois um revisão manual.

Por fim, temos a tabela 2.2.1, de *POS tags* relativas ao PTB.

Tabela 3 – Tabela de *POS tags* do *Penn Treebank*, com anotações. Adaptada de Santorini (1990b)

Tag	**Legenda original**	**Tradução da legenda**
CC	Coordinating conjunction	Conjunção coordenada
CD	Cardinal number	Número cardinal
DT	Determiner	Determinante/artigo
EX	Existential there	There existencial
FW	Foreign word	Palavra estrangeira
IN	Preposition/subordinating conjunction	Preposição/conjunção subordinada
JJ	Adjective	Adjetivo
JJR	Adjective, comparative	Adjetivo, comparativo
JJS	Adjective, superlative	Adjetivo, superlativo
LS	List item marker	Marcador de item de lista
MD	Modal	Modal
NN	Noun, singular or mass	Substantivo, singular ou conjunto
NNS	Noun, plural	Substantivo, plural
NNP	Proper noun, singular	Substantivo próprio, singular
NNPS	Proper noun, plural	Substantivo próprio, plural
PDT	Predeterminer	Predeterminante

Continua na próxima página

whereas in the corresponding plural phrase the ones, ones is always tagged as NNS (plural common noun), despite the parallel function of one and ones as heads of the noun phrase.// By contrast, [...], we encode a word's syntactic function in its POS tag whenever possible. Thus, one is tagged as NN (singular common noun) rather than as CD [...] when it is the head of a noun phrase". Tradução própria.

[25] Anotador é o sistema que atribui *POS tags* a cada palavra.

[26] No original: "*but in practice, multiple tags are restricted to a small number of recurring two-tag combinations.* Tradução própria."

Tabela 3 – *Continuação da página anterior*

Tag	**Legenda original**	**Tradução da legenda**
POS	Possessive ending	Encerramento possessivo ('s)
PRP	Personal pronoun	Pronome pessoal
PRP$	Possessive pronoun	Pronome possessivo
RB	Adverb	Advérbio
RBR	Adverb, comparative	Advérbio comparativo
RBS	Adverb, superlative	Advérbio superlativo
RP	Particle	Particula
SYM	Symbol (mathematical or scientific)	Símbolo (matematico ou científico)
TO	to	to (para)
UH	Interjection	Interjeição
VB	Verb, base form	Verbo, infinitivo
VBD	Verb, past tense	Verbo, passado
VBG	Verb, gerund/present participle	Verbo, gerûndio/presente particípio
VBN	Verb, past participle	Verbo, passado particípio
VBP	Verb, non-3rd ps. sing. present	Verbo, presente singular não-3ª pessoa
VBZ	Verb, 3rd ps. sing. present	Verbo, presente singular 3ª pessoa
WDT	wh-determiner	Determinante com WH (What, Which)
WP	wh-pronoun	Pronome com WH (who, whose, which, what)
WP$	Possessive wh-pronoun	Pronome possessivo com WH (whose)
WRB	wh-adverb	Advérbio com WH (when, where, how, why)

2.2.2 *Treebanks* para Língua Portuguesa

Serão descritos nessa Seção alguns dos *treebanks* existentes para a língua portuguesa. Estes são os *treebanks* que usaremos no nosso processo de transdução.

2.2.2.1 FLORESTA SINTÁ(C)TICA

A Floresta Sintá(c)tica é um projeto colaborativo entre o Linguateca[27] e o VISL[28]. Consta de um conjunto de diversos *treebanks*, em diversos estágios de construção, e diversos usos. O projeto, atualmente com 4 partes distintas. Como visto em (LINGUATECA, 2010):

> "Atualmente, a Floresta Sintá(c)tica tem quatro partes, que diferem quanto ao gênero textual, quanto ao modo (escrito vs falado) e quanto ao grau de revisão linguística: o Bosque, totalmente revisto por linguistas; a Selva, parcialmente revista, a Floresta Virgem e a Amazônia, não revistos. Junto, todo esse material soma cerca de 261 mil frases (6,7 milhões de palavras) sintaticamente analisadas."

Neste trabalho, utilizamos o Bosque. Ele está disponível no site da Linguateca, em sua versão 8.0, que data de 2008.

Cabe notar que o Bosque está, atualmente, no projeto Universal Dependencies (UD). Decidimos manter o uso da versão 8.0 por ser disponibilizado, também, em formato com PTB, o que facilitaria (em tese) nosso estudo. O formato disponibilizado pelo UD segue o padrão CoNLL (NIVRE et al., 2007).

```
informação textual
nº frase: texto
A1
NÓ RAIZ<
=NÓ 1
==NÓ 1.1.
===NÓ 1.1.1
====NÓ 1.1.1....n
==NÓ 1.2.
===NÓ 1.2.1.
====NÓ 1.2.1....
```

Figura 3 – Formato Árvores Deitadas. Adaptado de Freitas (2006, p 6)

O Bosque segue o formato chamado Árvores Deitadas (FREITAS, 2006, p 6), que se apresenta como na Figura 3. Como descrito em (FREITAS; AFONSO, 2007):

> "A cada frase está associada informação textual, isto é, informação relativa ao extracto a que a frase pertence, o número da frase no Bosque e o texto (frase

[27] https://www.linguateca.pt/
[28] https://visl.sdu.dk/

> *per se*). Cada frase é iniciada por A1 (análise 1 da frase). A mesma árvore pode ter mais do que uma análise distinta que são indicadas por A2, A3, etc. [...]
> NÓ RAIZ é o nó mais alto da árvore correspondente à sua raiz, por isso é único, isto é, não existem mais nós ao mesmo nível. Assim, o nó raiz não exibe descontinuidade nem pode estar coordenado.
> Todos os outros nós constituintes da árvore (NÓ 1 e nós dependentes e os dependentes dos nós dependentes (NÓ 1.1. ou NÓ 1.2. a NÓ 1.1.1....N ou NÓ 1.2.1....N) estão por isso abaixo da raiz da árvore."

=>N:art('o' <artd> M P) Os
=H:n('promotor' M P) promotores

Figura 4 – Exemplo de nós no formato AD. Como descrito em Freitas e Afonso (2007), "A funçao de 'os' é de dependente de um núcleo nominal (N) que está a sua direita (>), por isso a marcação >N; a forma de 'os' é artigo. Tem-se entao o par de função e forma >N:art. 'promotores', por sua vez, é o núcleo (H) do sintagma, e sua forma é substantivo/nome (n). O par função e forma é portanto H:n". Adaptado de Freitas e Afonso (2007)

.

Cabe notar que cada nó segue o formato (F:f), ou seja, **F**unção e **f**orma. Como descrito em (FREITAS; AFONSO, 2007):

> "A função corresponde à função sintáctica (sujeito, predicador, etc.) que cada constituinte possui em cada oração ou sintagma que compõe a frase. A forma corresponde à estrutura interna dos constituintes, isto é, sintagmas e orações para os nós não terminais, e, para os nós terminais, é usada uma classificação muito próxima das classes de PoS (advérbio, adjectivo, etc.)."

Como podemos ver na Figura 4, cada nó é rico em informações morfossintáticas. O tratamento para isto será melhor descrito na Seção 3.3. As *tags* utilizadas no Bosque, bem como seu nome, podem ser vistos na Tabela 8, ou no anexo 1 da Bíblia Florestal (FREITAS; AFONSO, 2007). A Bíblia é o manual de anotação, que descreve toda estrutura do Projeto (FREITAS, 2006) .

2.2.2.2 CINTIL

O CINTIL é um *dataset* desenvolvido pela *Natural Language and Speech Group* (NLX-Group[29]) da Universidade de Lisboa[30]. Seu objetivo é permitir o estudo linguístico

[29] <http://nlx.di.fc.ul.pt/>
[30] <https://www.ulisboa.pt>

da língua portuguesa, variante europeia.

Por (CARVALHEIRO, 2012, p 1), ele é um corpus de árvores sintáticas de constituência, de textos em português, constituído por 10039 sentenças e 110166 tokens, tirados de diversas fontes de domínios: notícias (8861 sentenças, 101430 tokens), romances (339 sentenças, 3082 tokens). Além disso, há também 779 sentenças (5654 tokens) que são usadas para testes de regressão de gramáticas computacionais que apoiaram a anotação. A Tabela 4 demonstra a distribuição do corpus.

Sub-corpus	id	Sentenças	*Tokens*	Domínio
Sentenças para testes de regressão	aTSTS	779	5654	Teste
CINTIL-*Corpus* Internacional do Português	bCINT	1219	13516	Notícias
	cCINT	399	3082	Romances
CETEMPublico	eCTMP	7541	86905	Notícias
Penn TreeBank (tradução)	dPENN	101	1012	Notícias
Total		10039	110166	

Tabela 4 – Distribuição de sentenças e *tags* pelo CINTIL. Adaptado de (CARVALHEIRO, 2012, p 1).

Foi usada uma classificação semi-automática na criação do *treebank*. Como visto no Iness[31] (ROSÉN et al., 2012), num primeiro momento, uma *deep computational grammar* (BRANCO; COSTA, 2008) é usada para gerar todas as possíveis árvores em uma sentença. Na sequência, é feita uma desambiguação manual, no qual dois anotadores escolhem a melhor árvore. Em caso de empate, um terceiro especialista servirá de árbitro.

O CINTIL é distribuído em formato XML, e as árvores classificadas tem formato semelhante ao PTB, fazendo separações usando parênteses, e com classes muito parecidas. A Tabela 6 demonstra as *tags* originais, e a frequência de uso no banco.

Os padrões de classificação do CINTIL atual está catalogado no *CINTIL TreeBank handbook* (BRANCO et al., 2011).

O CINTIL tem uma versão de 2005, e uma mais atual distribuída em 2012, pelo Metashare [32]. Existe o site de divulgação oficial, cintil.ul.pt[33], porém ele não é atualizado há algum tempo. Isso se nota pois o *tagset* disponibilizado por ele está desatualizado, sendo referente à versão anterior. O mais atual segue as diretrizes do supracitado *Handbook*.

[31] <http://clarino.uib.no/iness/page?page-id=port-descr>
[32] <http://shorturl.at/yCGIO>
[33] <http://cintil.ul.pt/pt/cintilwhatsin.html>

2.3 Análise Sintática (*Parsing*)

Análise sintática, ou *parsing*, é uma técnica que possui importância principalmente em duas áreas de conhecimento. A primeira, no campo do processamento de Linguagens de Computação. Na segunda, no campo do Processamento de Linguagens Naturais.

Sobre linguagens de computação, Aho, Sethi e Lam (2008, p 39) definem a análise sintática como "O processo para determinar como uma cadeia de terminais pode ser gerada por uma gramática". Nessa definição, terminais são (*Ibid., p 27*) "os símbolos elementares da linguagem, definidos pela gramática". Sobre gramática, (*Ibid., p 39*) "descreve [...] a estrutura hierárquica da maioria das construções de linguagens [...]".

Já sobre Processamento de Linguagem Natural (que será dado foco neste trabalho), como descrito por Charniak (1997, p 33), *parsing* é "O processo de atribuir marcadores de frase a uma sentença" [34]. Por exemplo, na frase "*o cachorro late*", *o* pode ser marcado como artigo, *cachorro* como substantivo, *late* como verbo. *O cachorro*, juntos, formam um sintagma nominal. O verbo, sozinho, faz parte de um sintagma verbal. E ambos formam a sentença. Como fica mais claro na Figura 5.

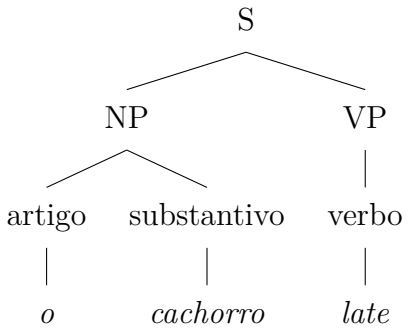


Figura 5 – Exemplo de árvore simples (Adaptado de Charniak (1997, p 34))

A esta estrutura dá-se o nome de *árvore de derivação*, *árvore*, ou *parse*.

Para sentenças simples, este é um trabalho simples. Porém, muito facilmente uma frase pode gerar mais de uma árvore, ou seja, gerar uma ambiguidade sintática[35], como no exemplo usado por Pagani (2009, p 10), na Figura 6.

Além da ambiguidade, existe outro problema: a quantidade de árvores incoerentes. Como destaca Charniak (1997, p 33), o exemplo "*[...] Implica que podemos atribuir*

[34] No original: "*the process of assigning a phrase marker to a sentence*". Tradução própria
[35] Nota: Não haverá foco na ambiguidade lexical neste trabalho.

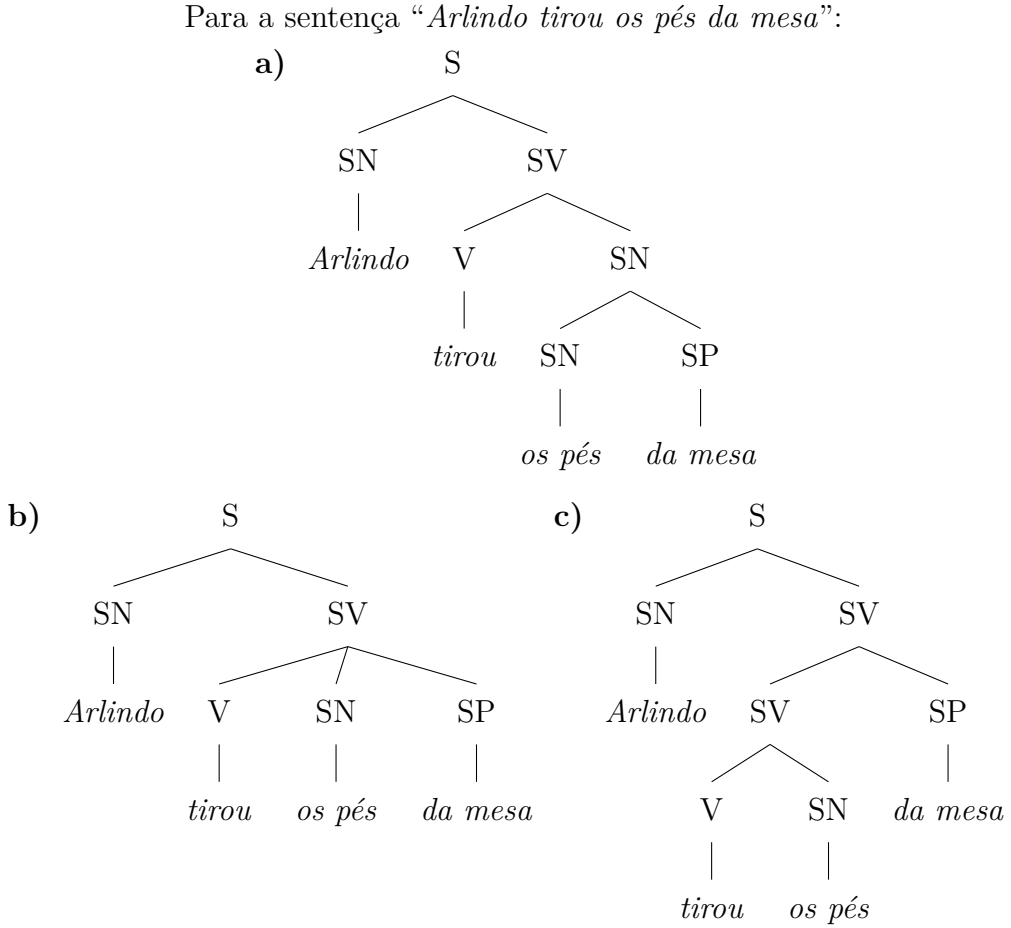


Figura 6 – Exemplo de ambiguidade entre árvores (Adaptado de Pagani (2009, p 10))

pelo menos um significado semi-plausível para toda árvore possível. Para a maioria das gramáticas [...], este não é o caso [36]". Ou seja, é necessário filtrar as árvores com real relevância e que mais se aproximem de uma estrutura correta.

Nesse ponto, existem métodos que resolvam essa ambiguidade, como as Gramáticas Lives de Contexto Probabilísticas e a classificação lexicalizada. Ambos serão abordados, respctivamente, nas seções 2.3.1 e 2.3.2.

Pode-se questionar qual a utilidade de *parsing*, e do seu estudo. Manning e Schütze (1999) dedicam alguns capítulos para demonstrar seu uso e técnicas, como Tradução de Máquina, Agrupamento (*clustering*), Recuperação de Informação e Categorização de Textos[37].

2.3.1 Classificação Estatística (*Statistical Parsing*)

Como dito em 2.3, uma das dificuldades no processo de *parsing* é a resolução de ambiguidades. (CHARNIAK, 1997, 33) "maior parte das árvores que gramáticas com

[36] No original: "[...] implies that we can assign at least a semiplausible meaning to all the possible parses. For most grammars [...], this is not the case." Tradução própria.
[37] (MANNING; SCHÜTZE, 1999, capítulos 13,14,15,16)

ampla cobertura encontram [...] não fazem sentido" [38]. Isto exige uma estratégia para, não só identificar a sentença mais provável, como rejeitar classificações de baixa qualidade.

Uma estratégia possível é o uso da estatística. Como já visto em 2.3 e 2.1.1, a estrutura de um *parse* é uma árvore de derivação, em que os nós terminais (folhas) são as palavras, e os nós não-terminais são referentes aos sintagmas, e compõem a estrutura de derivação e constituência. Pode-se, então, catalogar todas as possíveis derivações de cada sintagma, criando uma *gramática*. Tais derivações são chamadas de regras.

Determinar qual regra utilizar em cada derivação para classificar uma sentença se torna a questão a ser resolvida. Atribuindo-se um valor de probabilidade para cada regra, pode-se, então, selecionar as regras mais prováveis para uma dada sentença. (CHARNIAK, 1997, p 37) "Classificação estatística funciona atribuindo probabilidades à possíveis árvores de uma sentença, localizando a árvore mais provável, e então apresentando tal árvore como resposta" [39].

Com isto, pode-se intuir uma estrutura simples, uma Gramática Livre de Contexto com probabilidades atribuidas às suas regras. (MANNING; SCHÜTZE, 1999, p 382):

> "O modelo probabilístico mais simples para incorporação recursiva é uma PCFG, uma Gramática Livre de Contexto Probabilística (às vezes também chamada Estocástica), que é simplesmente uma GLC com probabilidades adicionadas às regras, indicando o quão prováveis diferentes reescritas são." [40]

Deste modo, dada uma sentença s, e uma possível árvore π, sendo c os constituintes da árvore, e $r(c)$ a regra usada para expandir c, temos a equação 2.1.

$$p(s, \pi) = \prod_{c} p(r(c)) \tag{2.1}$$

Ou seja, a probabilidade de uma dada árvore, para uma sentença específica, é o produto das probabilidades de todas as regras de expansão desta mesma árvore.

Existem várias vantagens nas PCFGs. A estrutura de gramática livre de contexto é bastante conhecida tanto por cientistas da computação, como por linguistas. Também, como visto em (CHARNIAK, 1997, p 38), algoritmos baseados em PCFG são tão eficientes quanto algoritmos não-estatísticos baseados em gramáticas.

[38] No original: "*most of the parses that wide-coverage grammars find are [...] pretty senseless.*" Tradução própria.

[39] No original: "*Statistical parsers work by assigning probabilities to possible parses of a sentence, locating the most probable parse, and then presenting the parse as the answer.*" Tradução própria.

[40] No original: "*The simplest probabilistic model for recursive embedding is a PCFG, a Probabilistic (sometimes also called Stochastic) Context Free Grammar which is simply a CFG with probabilities added to the rules, indicating how likely different rewritings are.*" Tradução própria.

Para se criar uma PCFG, o procedimento é como segue: deve-se ler árvores previamente classificadas (e consideradas corretas). Neste processo, deve-se anotar a quantidade de regras utilizadas. No exemplo de (*Ibid., p 38*):

> "É possível ler todas as regras necessárias dessa maneira. Além disso, podemos atribuir as probabilidades às regras contando com que frequência cada regra é usada. Por exemplo, se a regra $np \rightarrow det\, noun$ é usada, digamos, 1000 vezes, e regras gerais de np são usadas 60.000 vezes, então atribuímos à essa regra a probabilidade $1,000/60,000 = .017$" [41]

Para catalogar as regras necessárias para uma gramática, é necessário um conjunto de sentenças/árvores pré-classificadas. São chamados de bancos de árvores, florestas sintáticas, ou *treebanks*, que serão comentados em 2.2. Um dos mais populares entre eles, o *Penn Treebank* (ou PTB), é abordado em 2.2.1. Preferencialmente, esses conjuntos tem que ter uma quantidade alta de sentenças. O *Penn Treebank*, por exemplo, tem na casa de 50.000 árvores para que, com isto, sejam catalogadas uma quantidade grande de regras. Pois até mesmo o PTB (*Ibid., p 39*) "não é grande o bastante para conter todas" [42].

PCFGs possuem características que foram listadas em (MANNING; SCHÜTZE, 1999, p 386-388):

- Quanto mais as gramáticas crescem, mais ambíguas se tornam;
- PCFG não dá uma boa ideia quanto à plausibilidade das árvores, uma vez que se baseia em fatores estruturais, não lexicais;
- PCFGs são boas para a indução de gramáticas; (GOLD, 1967 apud MANNING; SCHÜTZE, 1999) CFGs não podem ser aprendidas sem exemplos negativos (agramaticais, errados), mas PCFGs conseguem aprender apenas com exemplos positivos;
- Robustez. Pode lidar com sentenças erradas facilmente, desde que se usem sentenças semelhantes para treiná-las (com baixa probabilidade);
- Geram um modelo de linguagem probabilístico para linguagens naturais;
- O poder preditivo das PCFGs tende a ser maior do que gramáticas de estado finitos;
- Na prática, costumam ser piores que modelos *n-gram* (n > 1), uma vez que *n-grams* consideram o contexto; [43]

[41] No original: "*It is possible to read off all the necessary rules in this fashion. Furthermore, we can assign probabilities to the rules by counting how often each rule is used. For example, if the rule* $np \rightarrow det\, noun$ *is used, say, 1,000 times, and overall np rules are used 60,000 times, then we assign this rule the probability* $1,000/60,000 = .017$.". Tradução própria.

[42] No original: "*is not large enough to contain them all*". Tradução própria.

[43] Modelos como Modelos Ocultos de Markov, ou n-gramas, não serão abordados neste trabalho. Recomenda-se (MANNING; SCHÜTZE, 1999, p 191) e Ibid., p 317

- PCFGs não são bons modelos em separado, mas pode ser usado em conjunto com outros modelos;
- PCFGs tem certos vieses que podem ser inapropriados. Por exemplo, costuma dar preferência à árvores menores, pois uma quantidade pequena de expansões costuma ser valorizadas, uma vez que reescritas individuais tem probabilidades maiores.

Por fim, Charniak (1997, p 38) "*PCFGs by themselves do not make particularly good statistical parsers, and many researchers do not use them*" [44]. Uma possível razão é pelo fato de, dada a própria natureza deste tipo de gramática, contexto e estrutura léxica das sentenças são ignoradas. Uma possível alternativa é apresentada em 2.3.2.

2.3.2 *Lexicalized Parsing*

Considerar cada palavra no processo de *parsing* seria muito problemático. Algumas palavras podem aparecer no conjuntos de dados apenas uma vez, ou ser uma conjugação pouco vista de um verbo pouco utilizado, e isso rapidamente se tornaria um problema. Uma alternativa, então, é considerar o núcleo dos constituintes. Por (CHARNIAK, 1997, p 40) *parsers* estatísticos lexicalizados coletam, então, duas estatísticas. Uma relativa ao núcleo do sintagma para a regra usada para expandir este sintagma, denotado $p(r|h)$ (para r regra, e h núcleo. E o núcleo de um sintagma com relação ao núcleo de uma subárvore, ou $p(h|m,t)$ (sendo h núcleo da subárvore, m o núcleo do sintagma mãe, e t o tipo (*POS tag*) da subárvore.

Assim, a Equação 2.1 se torna a Equação 2.2:

$$p(s,\pi) = \prod_{c} p(h(c)|m(c))p(r(c)|h(c)) \tag{2.2}$$

A Equação 2.2 é explicada por (*Ibid., p 40*),

> "Aqui, primeiro encontramos a probabilidade do núcleo do constituinte $h(c)$ dado o núcleo da mãe $m(c)$, e então a probabilidade da regra $r(c)$ dado o núcleo de c." [45]

Essa modificação, por (*Ibid., p 40*), permite que *parsers* alcancem resultados de *precision-recall* [46] de 87%, aproximadamente, contra 75% dos PCFGs básicos.

[44] "PCFGs sozinhas não fazem *parsers* estatísticos muito bons, e muitos pesquisadores não as usam". Tradução própria.

[45] No original: "*Here, we first find the probability of the head of the constituent $h(c)$ given the head of the mother $m(c)$ and then the probability of the rule $r(c)$ given the head of c*". Tradução própria.

[46] Abordado em 2.5

2.4 *Parsers*

Nesta sessão, serão comentados os *parsers* propriamente ditos. Tanto os verificados para a construção deste estudo, como o que foi utilizado nos experimentos, o *Stanford Parser*.

2.4.1 Parser Utilizado - Stanford Parser

Para este trabalho, decidimos utilizar o *Stanford Parser* (SP). Desenvolvido pelo *Stanford NLP Group*[47], consiste de um pacote escrito na linguagem Java com diversos parser *incluídos*, como o neural, lexicalizado, PCFG etc. Possui modelos pré-programados para diversas línguas, como árabe, inglês, alemão, francês, espanhol e chinês. Ele está disponível como biblioteca no Maven, para desenvolvimento. Porém, é possível utilizá-lo por meio de API (as ferramentas necessárias estão inclusas no pacote, mas pode ser visto também em <http://nlp.stanford.edu:8080/parser/>), ou por terminal de comando unix. Existem pacotes baseados no SP para outras linguagens, como Python, Ruby, PHP, .NET etc.

Também disponível pelo Stanford NLP Group, existe o CoreNLP, que consiste em (MANNING et al., 2014, p 55):

> "Um *framework* de *pipeline* de anotações em java (pelo menos, baseado na JVM), que provê a maior parte do núcleo dos passos de processamento de linguagem natural (NLP), da tokenização à resolução de co-referência." [48]

Seu objetivo é tornar a implementação de procedimentos NLP mais simples, com diversas ferramentas disponíveis de maneira compacta.

Neste trabalho, não foi desenvolvido código baseados no SP, nem no CoreNLP, por alguns motivos. Primeiramente a proposta do trabalho é, justamente, verificar o quão simples seria desenvolver um parser alternativo, utilizando ferramentas já disponíveis. Segundo, o aprendizado de um novo *framework* demandaria uma quantidade de tempo não contemplada por este trabalho.

Outra pergunta que pode surgir é: no site, são disponibilizados diversos pacotes de idiomas, porque não usar algum alternativo? Mais uma vez, por dois motivos: o inglês é a língua *default* do SP, não sendo necessário o uso de nenhum outro pacote; e a língua inglesa possui menos inflexões, o que pode facilitar o desenvolvimento. Como visto em (MANNING; SCHÜTZE, 1999, p 371):

[47] <https://nlp.stanford.edu/>

[48] "*a Java (or at least JVM-based) annotation pipeline framework, which provides most of the common core natural language processing (NLP) steps, from tokenization through to coreference resolution*". Tradução própria.

> "Em muitas outras línguas, a ordem das palavras é muito mais livre, e as palavras vizinhas darão menor contribuição sobre morfossintaxe. Porém, na maioria delas, a riqueza de inflexões de uma palavra contribuem com mais informações sobre morfossintaxe do que acontece no Inglês." [49]

Para este trabalho, será utilizada a classe *LexicalizedParser*, por ser a classe padrão para o uso do SP através de comandos de terminal. O treino com esta classe dá como resultado métricas que abrangem 4 *parsers* distintos, que podem ser vistos nas Tabelas 19 e 22. Utilizaremos os resultados do PCFG para nossos estudos, como visto na Seção 4. PCFG e *parser* lexicalizado são abordados nas Seções 2.3.1 e 2.3.2, respectivamente. Os comandos utilizados para treinos e testes serão explicados em suas respectivas sessões.

O funcionamento interno do Stanford Parser foi demonstrado (de maneira simplificada) na Figura 7.

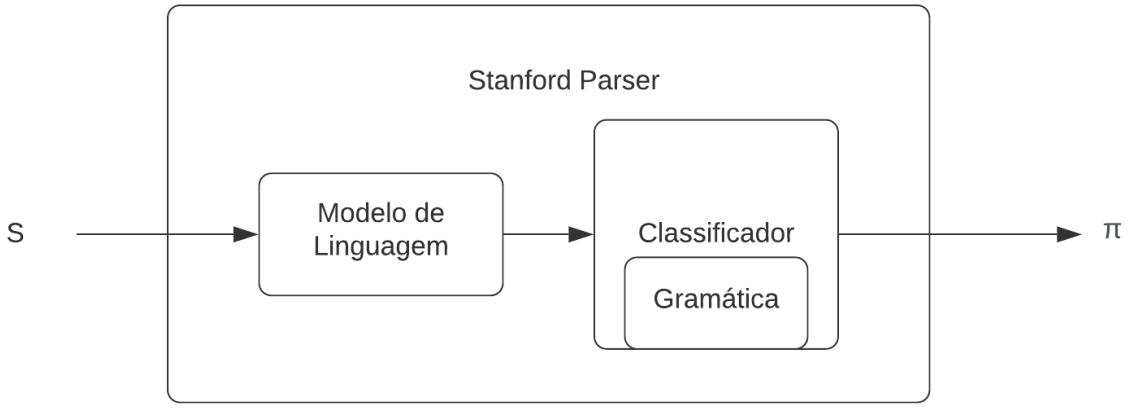


Figura 7 – Fluxograma descrevendo o funcionamento do *Stanford Parser* que já possui uma gramática disponível.

O *parser*, ao receber uma sentença S, faz um pré-processamento utilizando um modelo de linguagem. Ele é responsável por preparação da sentença para a classificação, como por exemplo, a separação de contrações, tais como *doesn't*, que se torna *does n't*. Depois desta etapa, o classificador (para este trabalho, o LexicalizedParser) realiza a classificação. Para tal, se utiliza de uma gramática, que possui a descrição das regras estatísticas de classificação. Esta gramática é gerada previamente, durante o processo

[49] "*In many other languages, word order is much freer, and the surrounding words will contribute much less information about part of speech. However, in most such languages, the rich inflections of a word contribute more information about part of speech than happens in English*". Tradução própria.

de treino. Ao final das operações, gerar-se-á uma árvore π, que descreve a estrutura morfossintática da sentença S.

Para a execução do SP, independente de como será utilizado, é obrigatório que o JDK[50] esteja instalado no seu sistema.

O *Stanford NLP Group* é um grupo de pesquisa baseado na universidade de Stanford (Califórnia, EUA), fazendo parte do *Stanford IA Lab*[51]. Possui membros tanto do departamento de Linguística, quanto de Ciência da Computação. Tem como objetivo o desenvolvimento de algoritmos que permitam a computadores o processamento de linguagem humana.

2.4.2 *Parsers* para Língua Portuguesa

Seguindo a premissa do trabalho, pesquisou-se por *parsers* para a língua portuguesa já disponíveis. Foram encontrados principalmente dois, o PALAVRAS e o LX-Parser, que serão descritos aqui.

2.4.2.1 PALAVRAS

Este *software* foi o produto da dissertação de doutorado de Echkard Bick[52], (BICK, 2000). É um *parser* baseado no paradigma de Gramática de Restrições. Este *parser* foi utilizado no projeto Floresta Sintá(c)tica, para classificação automática de textos, como pode ser visto em 2.2.2.1.

Além do projeto da Linguateca, vemos em (BICK, 2000) que o *parser* também é utilizado no projeto GramTrans[53], que utiliza o CETEMFolha e CETEMPúblico para fazer traduções dinamarquês/português.

Como visto em (LINGUATECA, 2010),

> "O CETENFolha (Corpus de Extractos de Textos Electrónicos NILC/Folha de S. Paulo) é um corpus de cerca de 24 milhões de palavras em português brasileiro, criado pelo projecto Processamento computacional do português (projecto que deu origem à Linguateca) com base nos textos do jornal Folha de S. Paulo que fazem parte do corpus NILC/São Carlos, compilado pelo Núcleo Interinstitucional de Lingüística Computacional (NILC)."

Por sua vez,

[50] Java Development Kit. Disponível em <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
[51] <http://ai.stanford.edu/>
[52] <https://visl.sdu.dk/~eckhard/Artikeloversigt.html>
[53] <https://gramtrans.com/>

> "O CETEMPúblico (Corpus de Extractos de Textos Electrónicos MCT/Público) é um corpus de aproximadamente 180 milhões de palavras em português europeu, criado pelo projecto Processamento computacional do português [...] após a assinatura de um protocolo entre o Ministério da Ciência e da Tecnologia (MCT) português e o jornal PÚBLICO em Abril de 2000."

O PALAVRAS está disponível sob licença proprietária, portanto não foi possível obter acesso para a realização de testes próprios[54].

2.4.2.2 LX-PARSER

Como descrito em (NLX-GRUPO DE FALA E LINGUAGEM NATURAL, 2010), "O LX-Parser é um analisador sintáctico de constituência para o Português baseado numa abordagem estatística". Em (SILVA et al., 2010) podemos ver que tanto o Bosque quanto o CINTIL foram considerados para o seu desenvolvimento. Cabe uma citação: "A maior parte do artigo, na verdade, consiste na descrição das muitas dificuldades que os autores tiveram de lidar quando adaptando o formato de árvore do Bosque para o formato adequado para treinar o *parser*" [55]. Por fim, os autores decidiram por continuar o desenvolvimento utilizando o CINTIL [56]. Foram encontradas dificuldades semelhantes no desenvolvimento desse trabalho.

LX-Parser usa como base o *Stanford Parser*, e conta com um *tagset* que pode ser visto na Tabela 6. O software *standalone* está disponível para download, porém utiliza uma versão muito antiga do SP, não sendo possível utilizá-lo. Mas contam com uma versão online que pode ser acessada em <http://lxcenter.di.fc.ul.pt/services/pt/LXParserPT.html>. Note-se que não é um *webservice*, ou seja, não é uma ferramenta que pode ser acessada remotamente por outro sistema. Exige-se assim, portanto, que o usuário acesse a página da *web* para usufruí-la.

2.5 Avaliação de Parsers de Constituência

Para verificar se um dado parser é capaz de classificar corretamente novas sentenças dadas, existem algumas métricas desenvolvidas. Dentre as mais conhecidas, se destacam as *PARSEVAL Measures*, e sua derivação, a *F-Measure*.

[54] <https://visl.sdu.dk/remoting.html>

[55] *Most of that paper actually consists in the description of the many difficulties that the authors need to cope with when adapting the tree format of Bosque to a format suited for training the parser*. Tradução própria.

[56] Pode-se supor que o fato de a mesma equipe de desenvolvimento do LX-Parser ter desenvolvido o CINTIL tenha tido forte influência nesta escolha.

2.5.1 *PRECISION, RECALL, F1-SCORE*

Considerando um treinamento supervisionado (como no caso dos *parsers* deste trabalho), utiliza-se um conjunto de exemplos corretos (chamados *gold standard*, padrão ouro), que podem ser usados para comparar com os resultados obtidos pelo sistema desenvolvido. Considerando o padrão ouro como objetivo/alvo, pode-se obter resultados semelhantes à Tabela 5. Como descrito por Manning e Schütze (1999, p 368):

> "Os casos considerados *tp* (verdadeiro positivo) e *tn* (verdadeiro negativo) são casos em que nosso sistema acertou. Os casos selecionados erroneamente em *fp* são chamados falso positivos, falsas aceitações, ou erros Tipo II. casos em *fn* que falharam de ser selecionados são chamados falsos negativos, falsas rejeições, ou erros Tipo I" [57]

	Actual	
System	target	¬target
selected	tp	fp
¬selected	fn	tn

Tabela 5 – Tabela de Confusão, ou *contingency matrix*. Adaptado de (MANNING; SCHÜTZE, 1999, p 268)

A partir daí podemos definir duas métricas, *precision* e *recall*[58]. Por (MANNING; SCHÜTZE, 1999, p 268-269) *precision* é definida como a medida da proporção de itens selecionados que o sistema acertou. *Recall* é definida como a medida da proporção dos itens alvo que o sistema selecionou. Ambas podem ser vistas na equação 2.3.

$$precision = \frac{tp}{tp + fp}, \qquad recall = \frac{tp}{tp + fn} \tag{2.3}$$

A medida *F-measure*, também conhecido por *F1-score* (DERCZYNSKI, 2016, p 262) é, por (SASAKI et al., 2007, p 1), a média harmônica entre *precision* (P) e *recall* (R), como pode ser visto na Equação 2.4. Seus valores variam entre 0 e 1.

$$F = 2P\frac{R}{P + R} \tag{2.4}$$

[57] "*The cases accounted for by tp (true positives) and tn (true negatives) are the cases our system got right. The wrongly selected cases in fp are called false positives, false acceptances or Type II errors. The cases in fn that failed to be selected are called fake negatives, false rejections or Type I errors*". Tradução própria.

[58] Precisão e Revocação

2.5.2 *PARSERVAL MEASURES*

As PARSERVAL measures são basicamente três: *precision* (ou *labeled precision*, LP), *recall* (ou *labeled recall*, RB) e *crossing brackets*, que seguem uma lógica semelhante à citada na Seção 2.5.1. Dado um *parser* que gere uma árvore, (MANNING; SCHÜTZE, 1999, p 433-434):

> "Precisão é quantos parênteses na árvore gerada combinam com aqueles da árvore correta, revogação mede quantos parênteses na árvore correta estão na árvore gerada, e parênteses cruzados dá a média de quantos constituintes em uma árvore cruza com as fronteiras na outra árvore" [59]

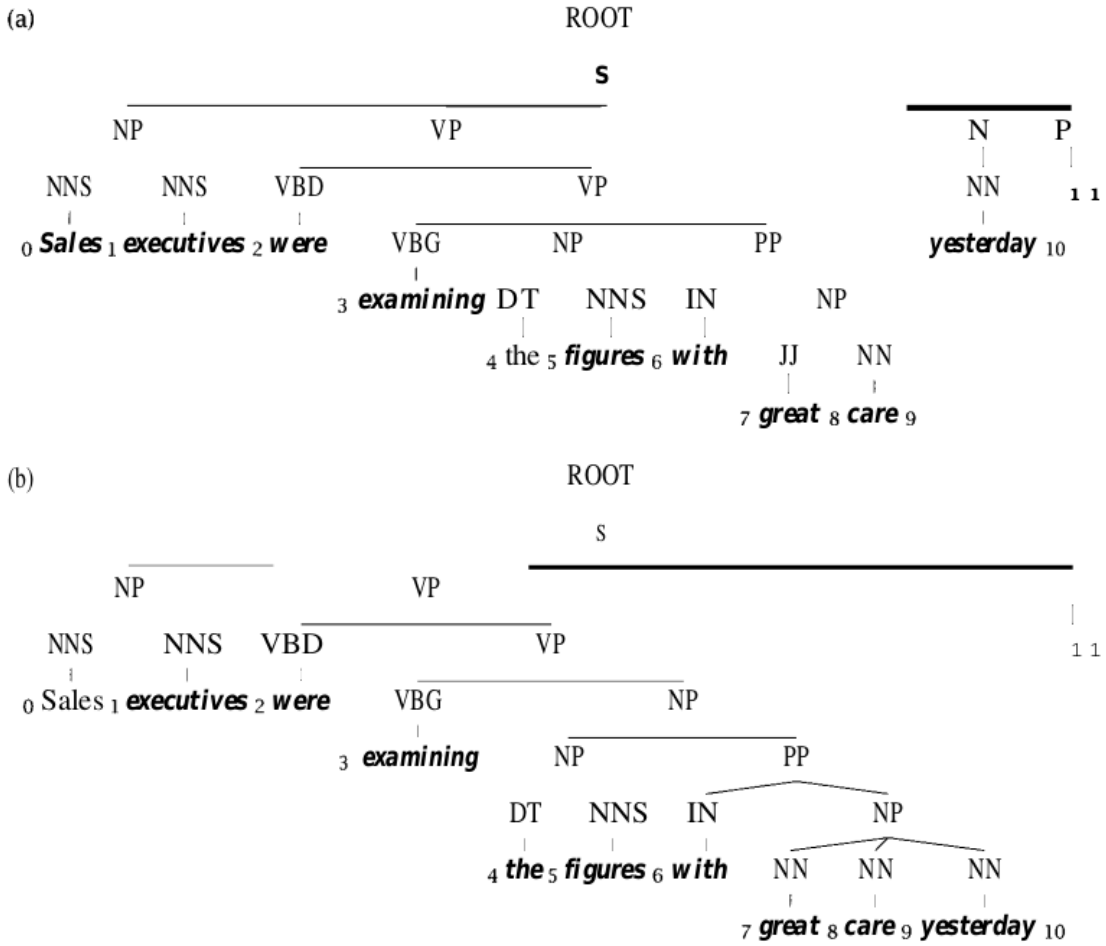
Nesse contexto, o cálculo do F-MEASURE se dá utilizando a mesma Equação 2.4.

Algumas críticas podem ser feitas à esse sistema. Manning (2018) demonstra que essa medida é muito sensível à propagação de erros em cascata. Ou seja, um constituinte mal colocado num nível mais baixo da árvore faz com que todos os nós acima dela também estejam errados, reduzindo muito a pontuação. Romanyshyn (2014) também nota que "O problema do PARSERVAL padrão é que ele conta nós como os mesmos, independente da estrutura subalterna que estes dominam" [60].

Isto pode ser melhor observado na Figura 8. Numa sentença $W = [w_0 \dots w_n]$, para w_j palavras, os nós dá árvore são identificados por $P - (i : f)$, sendo P a POS *tag*, i o inicio da *abrangência* do nó, e f o final. Nós com mesma POS são identificados pela ordem de aparecimento na árvore, e pela abrangência. Por *abrangência*, refere-se ao alcance de todos os seus descendentes. Assim, na árvore 8[a], o primeiro VP é representado por $VP - (2 : 9)$ por englobar a sentença de "*were*" a "*care*". Note que, ao posicionar o último NP, referente à "*yesterday*", a árvore candidata o marca como $NP - (7 : 10)$, em contraste com o "padrão-ouro", que deveria ser $NP - (9 : 10)$. Isto faz com que, não só este nó esteja errado, como todos os nós acima dele, fazendo com que os valores de LP, LR e, por fim, F1 sejam afetados negativamente.

[59] "*Precision is how many brackets in the parse match those in the correct tree, recall measures how many of the brackets in the correct tree are in the parse, and crossing brackets gives the average of how many constituents in one tree cross over constituent boundaries in the other tree*". Tradução própria.

[60] "*The problem of standard Parseval is that it counts nodes as the same regardless of the underlying structure they dominate*". Tradução própria.



(c) Brackets in gold standard tree (a.):
S-(0:11), NP-(0:2), VP-(2:9), VP-(3:9), **NP-(4:6),** PP-(6-9), NP-(7,9), *NP-(9:10)

(d) Brackets in candidate parse (b.):
S-(0:11), NP-(0:2), VP-(2:10), VP-(3:10), NP-(4:10), **NP-(4:6),** PP-(6-10), NP-(7,10)

(e) Precision:	3/8 = 37.5%	Crossing Brackets:	0
Recall:	3/8 = 37.5%	Crossing Accuracy:	100%
Labeled Precision:	3/8 = 37.5%	Tagging Accuracy:	10/11 = 90.9%
Labeled Recall:	3/8 = 37.5%		

Figura 8 – Demonstração do funcionamento do PARSERVAL. Extraido de (MANNING; SCHÜTZE, 1999, p 433). Note que o nó NP-(9:10) (*yesterday*), ao ser posicionado como filho do nó NP-(7:10), torna todos os nós acima dele também errados.

Parte II

Como Treinar seu Parser

3 Desenvolvimento

A partir de agora, será descrito o desenvolvimento do trabalho propriamente dito. Foram feitas as transduções dos *corpora* CINTIL e BOSQUE para o formato PTB, como já mencionado.

O procedimento de transdução *inter-corpora* foi semelhante para ambas conversões mencionadas neste trabalho. Cabe, então, uma descrição desta metologia na sessão 3.1. Depois desta explanação, veremos os pormenores do desenvolvimento da transdução dos *corpora* originais, e os treinamentos realizados a partir de tais transduções, serão pormenorizados nas seções 3.2 e 3.3.

3.1 Transdução inter-corpora

Neste trabalho, deseja-se a transdução entre um *corpus* em dado formato X, noutro num formato Y, de modo que não haja perda lexical, ou seja, sem afetar as palavras presentes no *corpus* original. Podemos considerar, portanto, o transdutor como uma função [1] equivalente à 3.1:

$$t(S) = S', dado\ S = CO, S' \in CD. \tag{3.1}$$

Sendo CD o *Corpus* de Origem (CO), a converter, e (CD) o Corpus de Destino, a ser convertido.

O procedimento de transdução consiste de 4 etapas principais:

1. Escolha dos *corpora*;
2. Estudo das estruturas do CO e do CD;
3. Planejamento de equivalências;
4. Construção do transdutor

[1] Para um debate mais aprofundado sobre o embasamento matemático de transdutores, recomenda-se a leitura de (MOHRI, 2004).

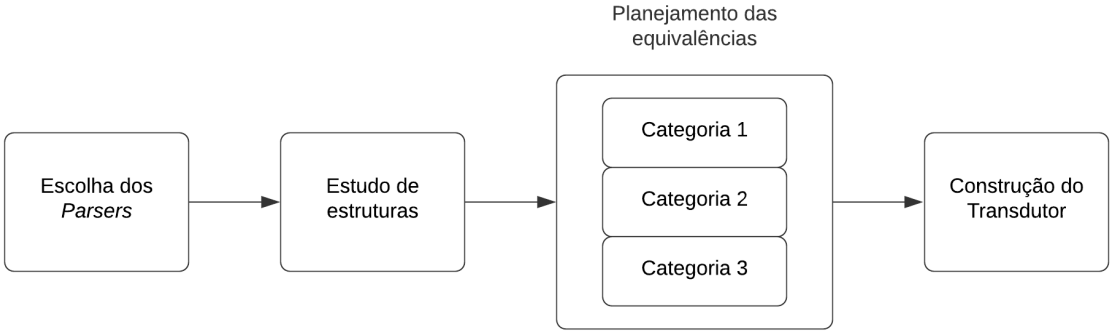


Figura 9 – Fluxograma descrevendo a metodologia inter-*corpora*

3.1.1 Escolha dos *corpora*

A escolha dos *corpora* pode parecer um procedimento trivial num primeiro momento, mas pode fazer com que o construtor de transdutores deixe de analisar coisas importantes.

Primeiramente, é preciso saber qual será a finalidade da transdução. No caso deste trabalho, o objetivo na construção do transdutor é fazer o treinamento do *parser* escolhido. Dado esse ponto de partida, a conversão inter-*corpora* não pode perder informação lexical, ou seja, não serão alteradas as palavras e suas estruturas. Também, é importante que as estruturas internas do produto da transdução sejam análogos ao do *corpus* de destino.

No contexto deste trabalho, a escolha do CD foi ocasionada pela necessidade: desejava-se utilizar o Stanford Parser para análise, e este *parser* está adaptado a entradas no formato Penn Treebank. Portanto, o CD é o PTB.

Os CO demandaram mais pesquisa. Fez-se necessário que, primeiramente, fossem *corpora* da língua portuguesa, principalmente das variantes brasileira e europeia. Também, para facilitar o procedimento de transdução, deu-se preferência à *corpora* cujo formato se assemelhasse ao do PTB. Por fim, é necessário que exista uma documentação bem desenvolvida, tanto de CO como de CD, pois com base nelas serão feitas as próximas etapas.

Nesse sentido, optar por CINTIL e BOSQUE fica mais evidente. O CINTIL possui estruturas internas semelhantes às do PTB, além de um *tagset* mais simplificado. Também, possui um número grande (10.140) de sentenças classificadas. O BOSQUE, por sua vez, possui uma documentação melhor resolvida (pelas análises realizadas neste trabalho), um *tagset* expressivo (com menos redundâncias), e está disponível gratuitamente.

3.1.2 Estudo das estruturas do CO e do CD

O estudo das estruturas internas dos *corpora* é peça chave na transdução inter-*corpora*. É fundamental saber como as sentenças em cada *corpus* se estrutura. Existe um

ponto ainda mais fundamental: Sabendo-se que os *corpora* são, em resumo, conjuntos de dados em forma de texto, qual o formato do arquivo a ser lido para ser transduzido? Como tal arquivo é escrito? O que o transdutor deve considerar ou ignorar em cada análise? Para utilizar o BOSQUE como exemplo, foi utilizado o formato de arquivo que emulava o PTB, portanto eram feitas as separações (bracketing) com o uso de parênteses. Porém, se fosse utilizado o formato original, estilo Árvores Deitadas (AD, ver Seção 2.2.2.1), seria necessário considerar os símbolos de = no processo de reconstrução da estrutura de árvore. O transdutor construído corre a sequencia de caracteres procurando por símbolos "abre parênteses" para iniciar a construção de uma nova sub-árvore, e o seu par "fecha parênteses" para concluir a reconstrução da mesma sub-árvore. Se fosse utilizado o arquivo em AD, seria necessária outra estratégia de reconstrução "arbórea". Por exemplo, poderia-se contar a quantidade de "=" no começo de cada linha e, se a quantidade aumenta, implicaria num novo nível da árvore; a redução implicaria no final da sub-árvore.

Deve-se dar atenção, também, ao *tagset* de cada *corpora*. Este tópico será mais explorado na Seção 3.1.3.1, mas deve-se ter em mente: quais as *tags* que cada *corpus* possui? Quais semelhanças e diferenças?

Por fim, faz-se necessário o estudo de estruturas internas. Como será visto com detalhes nas Seções 3.2 e 3.3, diversos arranjos internos exigem uma adaptação específica, e é importante tê-las em mente ao desenvolver o transdutor. Porém, a experiência da prática demonstra que muitas vezes, o *parser* que apontará tais estruturas para o desenvolvedor, quando houver erros de execução durante os treinamentos.

3.1.3 Planejamento das equivalências

Esta etapa está intrinsecamente ligada à etapa anterior (3.1.2). Aqui, serão planejadas as conversões propriamente ditas. O Planejamento de Equivalências pode ser subdividido em 3 categorias principais:

1. Identificação de *tags* correlacionadas;
2. Identificação de *tags* conceitualmente semelhantes;
3. Identificação de *tags* que exigem modificações estruturais.

O desenvolvedor deve criar uma tabela de equivalências, onde de um lado estarão as *tags* do CO, e do outro, as *tags* do CD. A coluna de *tags* do CO será preenchida a cada uma das etapas acima. Exemplos destas tabelas podem ser vistos nas Seções 3.2 e 3.3.

3.1.3.1 Identificação de *tags* correlacionadas

Nesta etapa, serão identificadas as *tags* que podem ser transduzidas diretamente do CO para o CD, sem perda de significado. Tais *tags*, em geral, não possuem grande dificuldade de identificação: a tradução dos seus nomes já costuma ser esse indicativo. Por exemplo: Para o CINTIL, como demonstrado na Tabela 4 da Seção 3.2, a leitura de seu manual (BRANCO et al., 2011, p 4) aponta que "*adjectives*" recebem a *tag* A. O Manual de Marcação do Penn Treebank (SANTORINI, 1990a, p 1), por sua vez, define "*adjectives*" como JJ. A tabela de correlações pode, portanto, ser preenchida com a correlação A/JJ. O BOSQUE define suas *tags* em português. Nestes casos uma tradução simples resolverá a questão: Se por um lado o BOSQUE não tem uma tag para "adjectives", por outro existe a tag "adjectivos". Portanto, na tabela de correlações BOSQUE/PTB, será preenchida a linha adj/JJ.

3.1.3.2 Identificação de *tags* conceitualmente semelhantes

Existem *tags* que, apesar de sua nomenclatura não indicar uma conversão direta, uma análise de definição, ou a observação de como a *tag* se comporta no *treebank*, pode apontar o caminho correto. Por exemplo, o *tagset* do BOSQUE, possui a *tag* fcl, que é definida por "Forma Oracional Finita". O *tagset* do PTB não possui algo semelhante. A definição de fcl em (FREITAS, 2006, p 12) informa que:

> "A oração finita (fcl) contém um verbo de forma finita. [...] A estrutura interna das orações finitas e não-finitas inclui um predicador (P) e argumentos ou adjuntos verbais."

Tal estrutura pode ser classificada como um Sintagma Verbal. E, por (MARCUS; MARCINKIEWICZ; SANTORINI, 1993, p 321), tal sintagma é marcado por VP. Assim, a tabela recebe a linha fcl/VP.

3.1.3.3 Identificação de *tags* que exigem modificações estruturais

Por fim, o modelo de *tags* que demandam mais trabalho de pesquisa e adaptação são aquelas que não permitem transduções diretas. Um exemplo marcante são as *tags* de coordenação de ambos os CO escolhidos, $CONJP$ para o CINTIL e CU para o BOSQUE. Ambas serão abordadas em suas respectivas Seções, 3.2.1 e 3.3.4.

Nestes casos, faz-se necessária, em primeiro lugar, o entendimento do que tal estrutura representa gramaticalmente na linguagem. O livro base para este trabalho foi (CASTILHO, 2010), onde foram realizadas a maior parte das pesquisas teóricas sobre linguagem e gramática. Também foi bastante utilizado (MIOTO; SILVA; LOPES, 2013),

para dúvidas linguísticas e estruturais. Em seguida, é necessário o estudo e domínio da forma como tal estrutura é descrita/implementada no CO, e no CD.

3.1.4 Construção do Transdutor

Neste trabalho, o transdutor $t(x)$ é um algoritmo de Busca em Profundidade, que varre a cadeia de caracteres de entrada S, $S = [w_1 w_2 \dots w_n],\ w_i \in \Sigma^*, S \in CO$ e tem como objetivo converter tal cadeia na cadeia S', $S' \in CD$. Sendo CO uma sequência de sentenças S, $CO = [S_1, S_2, \dots, S_n]$, ao final de $t(S) = S'$, será gerado um conjunto de S' de estrutura análoga às do CD, $[S_1, S'_2, \dots, S'_n] \in CD$.

No primeiro momento do transdutor, as cadeias de entrada são percorridas apenas uma vez por iteração da recursão. Nesta varredura, a estrutura de árvore é construída. A cada marcador de começo de sub-árvore w_i ("abre parênteses", em ambas COs do trabalho), inicia uma nova iteração recursiva, baseada numa nova sub-cadeia $S_{i+1,j},\ i < j,\ 0 \leq i,\ j \leq n$. Nesta nova interação, o procedimento se repete até encontrar o simbolo de final de sub-árvore w_j ("fecha parênteses" em ambos os COs). Nesse momento, a varredura de sub-árvore se encerra, a estrutura de dados é atualizada com essa sub-árvore, e a varredura que originou esta continua a partir de w_{i+1}.

Durante a varredura por sub-árvore/nó, são identificadas as *POS tags*. Em ambos COs, as POS, se localizam logo após "(" (particularidades serão pormenorizadas nas Seções 3.2 e 3.3). Cadeias de caracteres anteriores ao ")" são considerados palavras, ou seja $S_{i,j} = [(, w^*, \ , w^*,)],\ w \in \Sigma$ indica um nó. Ao identificar uma *tag*, o nó lógico recebera essa *tag*. Palavras não são modificadas. *Tags* identificadas como "problemáticas", ou seja, *tags* da categoria 3, recebem uma marca específica.

Na segunda etapa, as operações ocorrem diretamente na estrutura de dados gerada. É aqui que as *tags* são transduzidas, caso sejam das categorias 1 e 2. *Tags* da categoria 3, que foram marcadas, serão resolvidas. Existem pormenores sobre a detecção e adaptação de sinais gráficos (ponto, por exemplo), também. Sua descrição será feita nas Seções 3.2 e 3.3. Recomenda-se a leitura do algoritmo desenvolvido.

Por fim, na terceira etapa, é feita a impressão textual da estrutura de dados de modo análogo ao formato do CD. É feita uma nova Busca em Profundidade, gerando a cadeia de caracteres final. Poucas adaptações precisam ser feitas nesse momento.

A Figura 10 demonstra as etapas realizadas pelo transdutor.

O transdutor pode ser encontrado no GitHub do autor, <https://github.com/Fernandomn/treebank-transductor>.

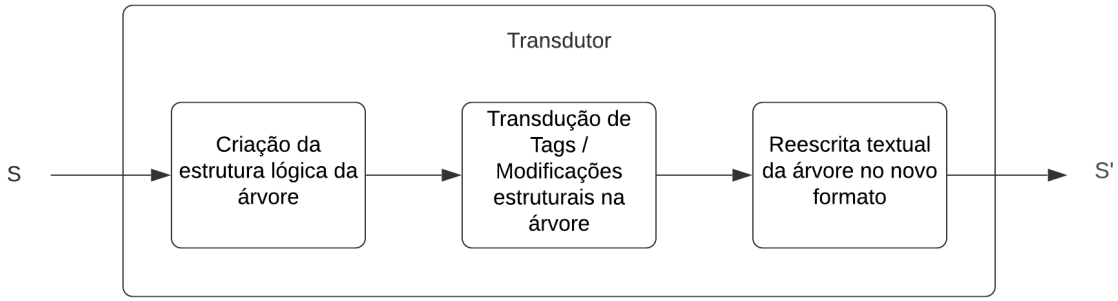


Figura 10 – Fluxograma descrevendo as etapas do transdutor

3.2 Transdução do CINTIL para o formato Penn Treebank

O pacote obtido, com o CINTIL, contém um guia de instruções (CARVALHEIRO, 2012), e o *treebank* propriamente dito em formato XML[2]. Para melhor uso, e melhor aplicação das árvores tanto para treino como para testes, fez-se necessária a separação deste arquivo em arquivos menores. Foram gerados dois tipos de arquivo: as sentenças originais ("*raw*"), e suas árvores ("*tree*").

As transduções do CINTIL para o PTB foram feitos de acordo com o procedimento explicado em 3.1. Os pormenores de categoria 3 serão explicados abaixo. A Tabela de Conversões pode ser vista na Tabela 6.

Outra dificuldade é que, como supracitado, o *tagset* informado no site oficial do CINTIL está defasado com relação ao *treebank* real. O *tagset* mais confiável, a respeito, é o "CINTIL TreeBank Handbook" (BRANCO et al., 2011). Foram usadas as *tags* listadas nele, e as que ocorrem no *treebank* concreto e que não foram previstas no *Handbook* (Por exemplo, P', C' etc).

Tabela 6 – Tabela de conversão: CINTIL para PTB

Cat.	**Tag Original**	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
1	A	Adjetivo	JJ	5527	
1	A'	Sintagma Adjetival	ADJP	114	
1	ADV	Advérbios	RB	5510	
1	ADV'	Sintagma Adverbial	ADVP	912	

Continua na próxima página

[2] <https://www.w3.org/XML/>

Tabela 6 – *Continuação da página anterior*

Cat.	Tag Original	Nome da Tag	Tag Convertida	Ocorrências	Observações
1	ADVP	Sintagma Adverbial	ADVP	428	
1	AP	Sintagma Adjetival	ADJP	1456	
1	ART	Artigo	DT	15583	
2	ART'	Artigo	NP	1	Equivaleria ao constituinte interemediário do *Determiner Phrase* (DP) Mioto, Silva e Lopes (2013). Porém, PTB não prevê esse tipo de estrutura. O sintagma mais indicado para receber determinantes (artigos) foi, portanto, NP
3	C	Complementador	CC	275	Será explicado na Seção 3.2.2
3	C'	Sintagma Complemental	_CP_	2	Será explicado na Seção 3.2.2
2	CARD	Cardinais	CD	2028	Números cardinais
2	CARD'	Sintagmas Cardinais	NP	504	(BIES et al., 1995) prevê que conjuntos de números são marcados como NP
2	CL	Clíticos	PRP	717	No CINTIL, ocorre apenas como pronome. De acordo com Branco et al. (2011), "Um pronome clítico tem a categoria CL. Ele é o núcleo de um NP"[3].
3	CONJ	Conjunções	CC	2460	Será explicado na Seção 3.2.1
3	CONJ'	Sintagma Conjuntivo	_CONJP_	92	Será explicado na Seção 3.2.1

Continua na próxima página

[3] "*A clitic pronoun has category CL. It is the head of an NP*"

Tabela 6 – *Continuação da página anterior*

Cat.	**Tag Original**	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
3	CONJP	Sintagma Conjuntivo	CONJP	609	Será explicado na Seção 3.2.1
3	CP	Sintagma Complemental	SBAR	1434	Será explicado na Seção 3.2.2
1	D	Artigo	DT	29	
2	D1	Quantificadores	DT	1	Sua descrição não aparece no Handbook, só no site. Único caso em que essa *tag* aparece no *treebank*, D1 se comporta como Artigo
2	D2	Quantificadores	JJ	1	Sua descrição não aparece no Handbook, só no site. Único caso em que essa *tag* aparece no *treebank*, D2 se comporta como Adjetivo
2	DEM	Demonstrativos	DT	1013	Para o PTB, *this*, *that*, *these*, *those* são, também, determinantes. Logo, DT
1	ITJ	Interjeições	UH	4	
2	ITJ'	Sintagma de Interjeição	INTJ	4	Por (BIES et al., 1995), "Interjeição. Corresponde aprocimadamente à etiqueta morfossintática UH (veja as diretrizes [Santorini (1990a)]"[4]
1	N	Substantivo	NNS	32989	
1	N'	Sintagmas Nominais	NP	18043	
1	NP	Sintagmas Nominais	NP	32258	

Continua na próxima página

[4] "*INTJ | Interjection. Corresponds approximately to the part-of-speech tag UH (see the POS guidelines [Santorini 1990]).*". Tradução própria.

Tabela 6 – *Continuação da página anterior*

Cat.	Tag Original	Nome da Tag	Tag Convertida	Ocorrências	Observações
2	ORD	Ordinais	CD	378	PTB não prevê o uso de ordinais. Ou melhor: eles costumam ser postos em locuções nominais
1	P	Preposição	IN	13920	
2	P’	Sintagmas Preposicionais	PP	337	Sua descrição não aparece no Handbook
2	PERCENT	simbolo percentual	NN	164	Nota 1: pode ser pronome + substantivo também (“por cento”). Nota 2: PTB considera o % como NN (*single noum*)
2	PERCENT’	Sintagma percentual	NP	80	PTB considera como NP
2	PERCENTP	Sintagma percentual	NP	36	
3	PNT	Pontuação	?	14748	Explicado na Seção 3.2.3
2	POSS	Possessivos	PP$	620	
2	POSS’	Possessivos	NP	10	Não existe um sintagma pronominal no PTB. Mantivemos como NP
1	PP	Sintagmas Preposicionais	PP	15382	
2	PRS	Pronomes Pessoais	PRP	395	
2	QNT	Quantificadores	PRP	889	De acordo com (Castilho (2010, p 55)), “Os pronomes abrigam as seguintes subclasses [...]: pessoais, demonstrativos, possessivos e quantificadores [...]”

Continua na próxima página

Tabela 6 – *Continuação da página anterior*

Cat.	Tag Original	Nome da Tag	Tag Convertida	Ocorrências	Observações
2	QNT'	Sintagma de Quantificadores	NP	19	Como supracitado, se refere ao sintagma que abriga quantificadores (pronomes)
2	REL	Relativos	PRP	861	Pronomes relativos
1	S	Sentença	S	24393	
1	V	Verbos	VB	13281	
1	V'	Sintagma Verbal	VP	2745	
1	VP	Sintagma Verbal	VP	15284	

As *tags* definidas como sendo da Categoria 3 serão analisadas abaixo, com as respectivas conversões necessárias.

3.2.1 Problemas com CONJ (Conjunção)

Conjunções são estruturas (palavras, no geral) que fazem parte da categoria semântica da Conectividade. Descrita por (CASTILHO, 2010, p 133):

> "Outra categoria semântica é a conectividade, gramaticalizada como preposições e conjunções. Essas classes ligam palavras e sentenças, com a diferença de que as preposições, como classe igualmente predicadora, atribui ao seu escopo traços de lugar, tempo, entre outros, propriedade não exercida pelas conjunções."

O mesmo autor descreve que sentenças podem ser ligadas por conjunções e que, ao fazê-lo, estamos criando uma relação conjuncional entre ambas. (CASTILHO, 2010, p 338):

> "Essa relação compreende a [...] coordenação, [...] formada por sentenças independentes umas de outras, ou de [...] subordinação, [...] formada por sentenças encaixadas umas em outras, tanto quanto [...] formada por uma sentença adjunta à outra."

A Tabela 7 mostra palavras e expressões utilizadas pelo CINTIL como conjunções.

ainda que	mas
até que	mesmo que
dado que	ou
de_ o que	para que
desde que	sem que
e	tanto mais que
em_ a medida em que	uma vez que
já que	

Tabela 7 – Expressões (conjuntos de palavras, ou palavras únicas) usadas como conjunções pelo CINTIL. Note que as preposições com *underline* se concatenam ao artigo posterior (de_ + o = do)

O *Penn Treebank* e o CINTIL lidam com conjunções de maneiras distintas. O PTB, em seu manual de anotação (BIES et al., 1995, p 117), dedica a Seção 7.5 para descrever tal fenômeno. Lá podemos ver que, para conjunções coordenadas, temos três casos: palavra simples (*and*, *but*, *or*, ...), multi palavra (*as well as*, *not to mention*, *rather than*, ...), e conjunções descontínuas (*not only... but*, *not... but instead*, ...). Palavras simples não precisam de marcação, a conjunção fica sem rótulo, como na Figura 11. Conjunções com várias palavras tem o sintagma de conjunção marcado como CONJP, e as conjunções são postas em estrutura plana, mostrado na Figura 12. Por fim, conjunções descontínuas tem apenas a parte com múltiplas palavras marcada por CONJP. A palavra isolada permanece isolada e sem marcação, como na Figura 13. O manual possui a descrição de mais casos, envolvendo Conjunções Coordenadas e *times*[5], porém não serão abordadas neste trabalho.

(NP (NP a hammer)
and
(NP a nail))

Figura 11 – Exemplo de conjunção coordenada com uma palavra (*single-word*). Adaptado de Bies et al. (1995, p 130)

O CINTIL, por outro lado, não é tão descritivo. É dito em (BRANCO et al., 2011, p 20): "Coordenação de dois constituintes A e B no sentido de uma conjunção coordenada Conj (tanto um item lexical, como *e*, ou uma vírgula) são uma cascata de adjunções [A [Conj [B]]]." [6] Pela observação do *treebank*, vemos CONJP se refere a toda a nova sentença em conjunção com a sentença inicial. CONJ' se refere ao núcleo da conjunção (aos moldes

[5] *Vezes*, no sentido de multiplicação. Exemplo, *three times*, ou *three times five*

[6] "*Coordination of two constituents A and B by means of a coordinative conjunction Conj (either a lexical item, such as e, or a comma) are a cascade of adjunctions [A [Conj [B]]].*" Tradução própria.

```
(S (NP-SBJ That)
    (VP builds
        (NP (NP confidence)
        ,
        (NP self sufficiency)
        ,
        (CONJP not to mention)
        (NP critical regulatory net worth)))
    .)
```

Figura 12 – Exemplo de conjunção coordenada com muitas palavras (*multi-word*). Adaptado de Bies et al. (1995, p 131)

```
(S (NP-SBJ The proposal)
    (VP represents
        (NP (CONJP not alone)
            (NP his own district)
            but
            (NP (NP all the people)
            (PP of
            (NP our country))))))
```

Figura 13 – Exemplo de conjunção discontínua (*discontinuous conjunction*). Adaptado de Bies et al. (1995, p 131)

da estrutura CP, que pode ser vista em (MIOTO; SILVA; LOPES, 2013, p 63)). Por fim, CONJ é a *POS tag* referente a conjunções.

CINTIL abarca toda a nova sentença da conjunção, como demonstrado na Figura 14. Como visto anteriormente, o mesmo não ocorre no PTB. Conjunções normalmente não são marcadas e, se forem, receberam a marca CONJP para o núcleo.

Com isto em mente, fez-se necessário reescrever a disposição das *tags*, para que: CONJP se referisse apenas aos núcleos conjuntivos, CONJ' fosse removida e, no contexto em que CONJ aparece dentro de *tags* CONJP, remover suas marcações, sem removê-las noutros momentos. Um exemplo pode ser visto na Figura 15.

3.2.2 Problemas com C (Complementizador)

Semelhante à CONJ em diversos aspectos, as *tags* C, C' e CP permitem a conjunção entre sentenças, tornando uma segunda sentença objeto de uma primeira. O tratamento feito com elas foi muito semelhante ao dado para a *tags* CONJ, com um diferencial: CONJ' ocorre sem necessariamente ter a *tags* CONJP como pai (12 casos), o que nunca ocorre com a família CP.

```
(S
   (S
      (NP
            (ART o) (N Manuel)
      )
      (VP
            (V é)
            (AP
                  (A maior)
                  (CONJP
                        (CONJ'
                              (CONJ de_) (CONJ o) (CONJ que)
                        )
                        (NP
                              (ART A) (N Maria)
                        )))))
   (PNT .)
)
```

Figura 14 – Sentença aTSTS-001/36, "o Manuel é maior do que A Maria". Exemplo de conjunção no CINTIL (Adaptado)

```
(S
  (S
    (NP
          (DT o)
          (NNS Manuel)
    )
    (VP
          (VB é)
          (ADJP
                (JJ maior)
                (CONJP de_ o que)
                (NP
                      (DT A)
                      (NNS Maria)
                )
          )
    )
  )
  .)
```

Figura 15 – Sentença aTSTS-001/36, "o Manuel é maior do que A Maria.", modificada pelo algoritmo desenvolvido para se adaptar ao PTB

3.2.3 Problemas com PNT (Pontuação)

Foi observado, também, que CINTIL e PTB lidam com pontuações de formas distintas. CINTIL usa a *tags* PNT para classificar estes símbolos. PTB prevê uma *tags* SYM, para símbolos. Além disso, vemos em (MARCUS; MARCINKIEWICZ; SANTORINI, 1993, p 52) que os símbolos costumam ser representados sem etiquetas, como exemplificado na Figura 16. Em (BIES et al., 1995, p 52), fica bastante claro: fora *bracketing* (parênteses, colchetes, chaves), os símbolos não recebem nenhuma *POS*. Quando recebe, como no caso de símbolos funcionando como palavras, ou símbolos matemáticos, são *tags* referentes ao sintagma. Um detalhe importante é como PTB lida com aspas e apóstrofos (*quote*, e *single-quote*). As aspas são removidas, e substituídas por dois apóstrofos ou duas crases (melhor visualizado na Figura 17).

```
( (S-1
        (PP-TMP For
                  (NP (NP the rest)
                      (PP of
                          (NP 1989))))
(PRN ,
        (S (NP-SBJ Mr. Hagen)
           (VP said
               (SBAR 0
                     (S *T*-1))))
        ,)
(NP-SBJ (NP Conrail 's)
           traffic and revenue)
"
(VP will
      (VP reflect
           (NP the sluggish economy)))
      .))
```

Figura 16 – Vírgulas marcando S entre parênteses (adaptado de Bies et al. (1995, p 52))

Fez-se necessário criar um *script* que removesse as *tags* PNT dos símbolos, do CINTIL, para reposicioná-los corretamente nas árvores, além de tratar *quotes*.

Porém, CINTIL não só identifica os PNT de forma diferentes, como também os POSICIONA de forma distinta, como podemos ver no comparativo da Figura 20. No exemplo, abordando ponto final, é definido em (BIES et al., 1995, p 52):

> "Neste corpus, cada unidade de texto é fechada no nível superior de parênteses não marcados [...]. Anteriormente, pontuações de nível superior [...] podiam ser anexadas àqueles parênteses de nível superior. Porém, nesta versão, tal

```
((SINV “
        (S-TPC-1 (NP-SBJ We)
        (VP have
            (NP (NP no useful information)
                (PP on
                    (SBAR whether
                            (S (NP-SBJ users)
                            (VP are
                                (PP-PRD at
                                    (NP risk)))))))))
                                    ,
                                    “
                                    (VP said
                                        (S *T*-1))
                                    (NP-SBJ (NP James A. Talcott)
                                        (PP of
                                            (NP (NP Boston ‘s)
                                                Dana-Farber Cancer Institute)))
                                    .))
```

Figura 17 – Exemplo de uso de aspas no PTB (fragmento adaptado da sentença wsj_0003)

> pontuação deve toda ser anexada um nível abaixo (para o nível mais alto dos parênteses rotulados), então existe apenas um nó no topo dentro dos parênteses não rotulados.” [7]

E resume em (BIES et al., 1995, p 57): “Ponto final é, por regra, um filho da estrutura de nível mais elevado” [8]. Já (BRANCO et al., 2011, p 29) define como “*Marcadores de fim de sentença estão na adjunção mais ao topo*” [9].

Vírgulas também tem uma problemática própria. Em (BIES et al., 1995, p 52):

> “Marcadores de pontuação pareados são irmãos do constituinte que eles rodeiam. Isto é verdade mesmo quando o membro inicial ou final do par pode ser visto como apagado. Por exemplo, as virgulas que definem uma cláusula subordinada

[7] “*In this corpus, each unit of text is enclosed in a top level of unlabeled brackets [...]. Formerly, top-level punctuation [...] could be attached to these top-level brackets. However, in this release, such punctuation should all be attached one level down (to the highest level of labeled brackets), so that there is only one top-level node within the unlabeled brackets.*” Tradução própria.

[8] “*Final punctuation as a rule is a child of the highest level of structure.*” Tradução própria.

[9] “*End of sentence markers are in the top most adjunction*”. Tradução própria.

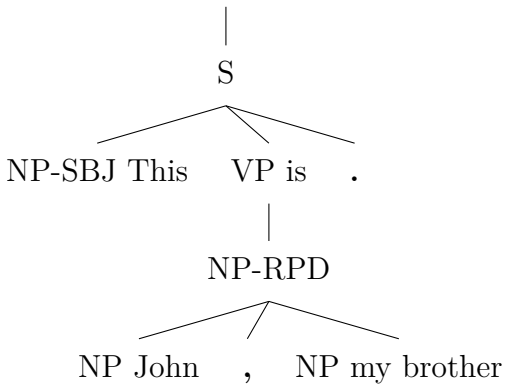


Figura 18 – "This is John, my brother."

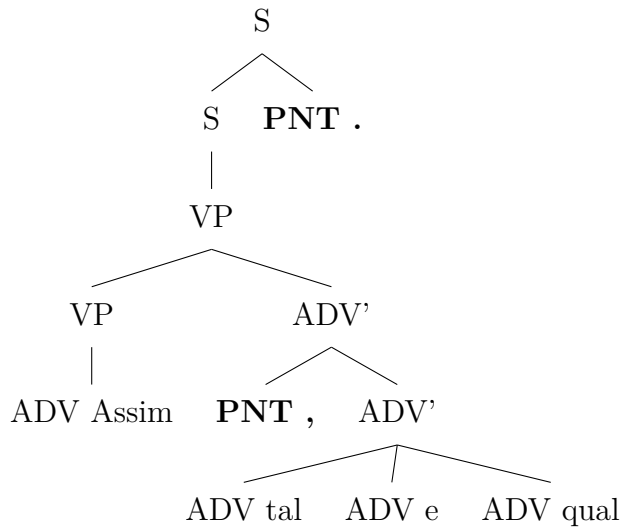


Figura 19 – "Assim, tal e qual."

Figura 20 – Comparativo entre posicionamento de sinais de pontuação entre o Penn Treebank e o CINTIL.

> ou clausula relativa de uma cláusula principal são irmãos do SBAR dominando a cláusula subordinada. [...]" [10]

Podemos ver esse fenômeno na Figura 16. Já em (BRANCO et al., 2011, p 30), "Vírgulas separando constituintes periféricos à esquerda são adjungidos à direita destes constituintes" [11], exemplo na Figura 21.

Existem casos no *treebank* em que pontuações não são associadas à etiqueta PNT. Esses casos costumam ser pontos de abreviação, reticências, ou apóstrofos (*single-quotes*) que antecedem nomes, e estão associados à etiqueta N. Para resolvê-los, primeiro mudamos suas etiquetas para PNT. Depois, continuamos com as operações previstas.

Fica claro, então, que é necessário *reposicionar* os sinais antes de passar as árvores processadas para o SP. Tal erro inviabiliza completamente qualquer tipo de teste, como podemos ver na mensagem de retorno exibida na Figura 22. Para o andamento deste trabalho, a solução foi apenas remover tais elementos, o que viabilizou o treinamento e avaliação.

[10] "*Paired punctuation marks are siblings of the constituents they surround. This is true even when the opening or closing member of the pair can be viewed as deleted. For instance, the commas that set off a subordinate clause or a relative clause from a main clause are siblings of the SBAR dominating the subordinate clause. [...].*" Tradução própria.

[11] "*Commas separating left periphery constituents are right adjoined to theses constituents*". Tradução própria.

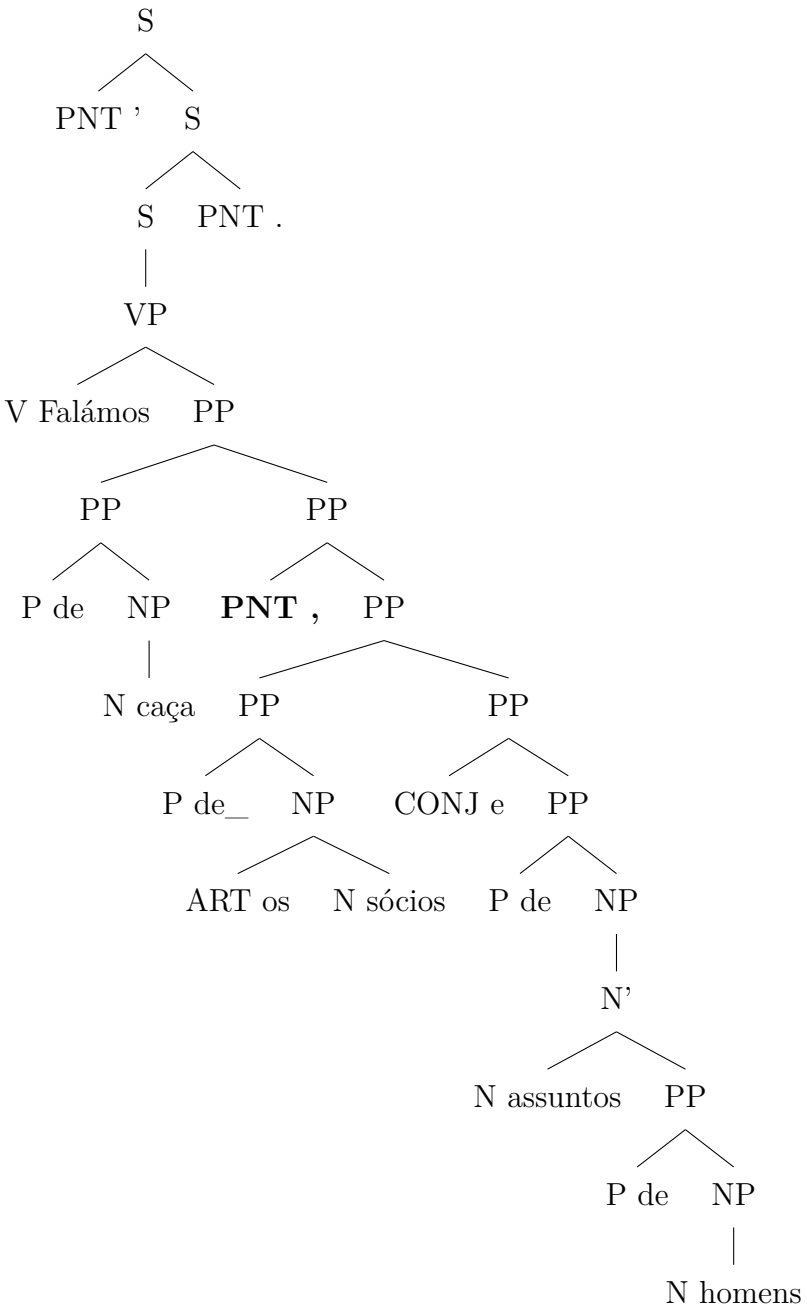


Figura 21 – Exemplo de comportamento da vírgula no CINTIL. Vírgulas separando componentes periféricos à esquerda são anexados à direita do constituinte. Adaptado de Branco et al. (2011, p 30)

3.3 Transdução do BOSQUE para o formato Penn Treebank

Como já visto, o Bosque é disponibilizado em duas variantes, português brasileiro (o CETEMFolha) e português europeu (o CETEMPublico). Este trabalho foi feito considerando ambas, porém focando na variante brasileira.

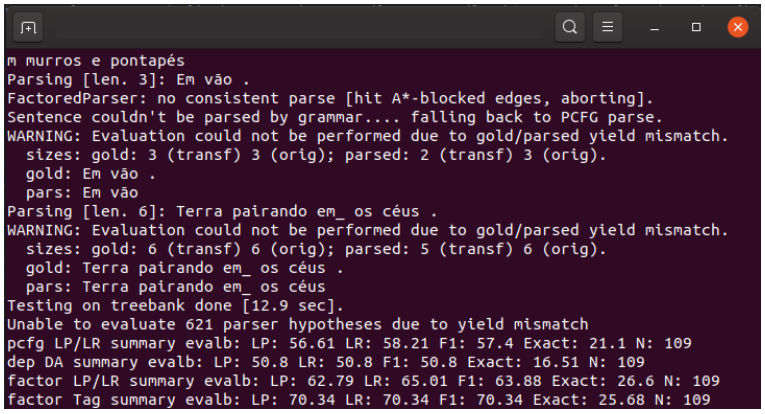


Figura 22 – Captura de tela de erro decorrente do mal posicionamento de pontuações na árvore do CINTIL, com relação ao formato PTB

Como descrito em 2.2.2.1, cada nó das árvores do Bosque são muito ricos em informação sintática. Como descrito em 2.2.1, o PTB é "pobre", se posto em comparação. Coube, então, fazer a remoção das outras características, mantendo apenas as *tags* de "forma", que equivalem às *POS tags*.

Uma dificuldade é que o arquivo de entrada está no formato ISO-8859. Ao ser convertido para o formato de texto aceito pelo SP (UTF-8), vários caracteres especiais, como letras acentuadas, se perdem. Foi necessário fazer a conversão do arquivo fonte, antes de realizar as operações de transdução. O comando utilizado pode ser visto em B.1.

Assim como o CINTIL, parte importante deste trabalho foi transduzir as *tags* do BOSQUE para o padrão PTB. Isso implicou em decisões que serão apresentadas na Tabela 8:

Tabela 8 – Tabela de conversão: BOSQUE para PTB

Cat.	**Tag Original**	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
3	acl	Forma Oracional averbais	<indefinido>	277	Não possui conversão direta. Melhor explicado em 3.3.8
1	adj	adjectivos	JJ	3484	
1	adjp	Sintagma adjectivais	ADJP	3367	
1	adv	advérbios	RB	3052	
1	advp	Sintagma adverbiais	ADVP	2288	
2	art	artigos	DT	10742	

Continua na próxima página

Tabela 8 – *Continuação da página anterior*

Cat.	Tag Original	Nome da Tag	Tag Convertida	Ocorrências	Observações
1	conj-c	conjunções coordenativa	CC	1723	Explicado na Seção 3.3.4
1	conj-s	conjunções subordinativa	IN	798	
3	cu	sintagma evidenciador de relação de coordenação	_CU_	1744	Será explicado na Seção 3.3.4
3	ec	prefixos	_EC_	80	Será explicado na Seção 3.3.1
2	fcl	Forma Oracional Finita	VP	6040	Sintagmas onde os verbos não estão no infinitivo. Sintagma verbal
2	icl	Forma Oracional não finita	VP	1827	Sintagmas onde os verbos estão conjugados. Sintagma Verbal
1	intj	interjeições	UH	22	
1	n	substantivos	NN	15724	
2	n-adj	substantivos / adjectivos	NN	174	Pesquisa mostrou que são todas as ocorrências são substantivos no CETEMFolha
2	np	Sintagma nominais	NP	22981	
1	num	numeral	CD	1625	
1	pp	Sintagma preposicionais	PP	11576	

Continua na próxima página

Tabela 8 – *Continuação da página anterior*

Cat.	Tag Original	Nome da Tag	Tag Convertida	Ocorrências	Observações
2	pron-det	pronomes determinativos	DT	1580	Pelo Santorini (1990a), "Esta categoria inclui [...] os determinantes indefinidos *another*, *any*, *some*, *each*, *either* [...], *neither* [...], *that*, *these*, *this* and *those* [...]" [12]. No português, também, por Mioto, Silva e Lopes (2013, p88) "[...] DP [*Determiner Phrase*] pode ter seu núcleo D [*Determiner*] preenchido por um item que tenha valor de determinante como artigos, demonstrativos e interrogativos[...]"
2	pron-indp	pronomes independentes	PRP	1001	PTB considera 4 tipos de pronomes: os pessoais, possessivos, wh-possessivos e wh-pessoais. Decidiu-se manter a marcação de pronomes pessoais. Isso é reforçado pela própria descrição de Freitas e Afonso (2007), "pronome independente (com comportamento semelhante ao nome)"
2	pron-pers	pronomes pessoais	PRP	891	
2	prop	nomes próprios	NNP	4575	
1	prp	preposições	IN	11694	

Continua na próxima página

[12] "*This category includes [...] the indefinite determiners another, any, some, each, either [...], neither [...], that, these, this and those [...]*"

Tabela 8 – *Continuação da página anterior*

Cat.	Tag Original	Nome da Tag	Tag Convertida	Ocorrências	Observações
2	sq	Sintagma sequências discursivas	S	56	Marcador de Sentença
1	v-fin	verbos finitos	VBP	6167	Verbos conjugados
1	v-ger	verbos gerúndios	VBG	328	
1	v-inf	verbos infinitivos	VB	1684	
1	v-pcp	verbos particípios	VBN	1577	
1	vp	Sintagma verbais	VP	8103	
3	x	<desconhecido>	VB	552	Explicado na Seção 3.3.5

Primeiramente, transduzimos todas as *tags* em sequência, mantendo a estrutura das árvores originais. Notamos, então, que para além da tradução de *tags*, algumas particularidades entre *treebanks* precisou de uma análise distinta, para maior consistência. Tais procedimentos são descritos a seguir.

3.3.1 Problemas com EC (Prefixos)

Prefixos são marcados com a *tag* "ec". No caso em específico, o prefixo (ou morfema prefixal) que recebem tais marcações são aqueles ligados à palavra por hifens, como na Figura 23:

```
...
    (>N:ec:ex-:: ex-)
    (H:n:jogador:M_P::: jogadores)
...
```

Figura 23 – Trecho da sentença CF515-1, do Bosque: "Ex-jogadores elogiam os colunistas Telé e Cruyff". Demonstração da aplicação da *tag* "ec".

O PTB não prevê situações como estas. Pelo contrário, como podemos ver no guia de marcações do PTB (BIES et al., 1995, p 315), tal estrutura é ignorada. Podemos

ler a respeito do uso de hifens em (*Ibid., p 58*), mas tal trecho não nos informa sobre o tratamento de prefixos. Recolhendo um exemplo do próprio corpus, analisemos a Figura 24.

```
...
( (S (S
        (PP-TMP In
                    (NP mid-October)
        ),
        (NP-SBJ-1 Time magazine)
        (VP lowered

...
```

Figura 24 – Fragmento da sentença wsj_0012, do PTB: "*In mid-October, Time magazine lowered its guaranteed circulation rate base for 1990 while not increasing ad page rates; with a lower circulation base, Time's ad rate will be effectively 7.5% higher per subscriber; a full page in Time costs about $120,000.*"

A flexão "*mid-October*" é representada, sozinha, como um sintagma nominal (*noun phrase*). Este padrão se repete em outros exemplos coletados. Portanto, fez-se necessário a criação de uma sub-tarefa que removesse tais *tags*, e que a estrutura fosse refeita, de modo a seguir os padrões do PTB.

3.3.2 Problemas com % (Porcentagem)

Para o Bosque, o sinal de percentagem (%) participa de um nó isolado marcado como substantivo, como na Figura 25. De acordo com a (FREITAS, 2006, p 113-114), este é um caso tratado como partitivo. "Por expressões partitivas entende-se tipos de expressões de quantificação que designam partes de um todo", e "Geralmente a unidade dividida em partes (expressa em expressões partitivas) é de natureza nominal ou são quantificadores". Para o PTB, "Porcentagem é simplesmente um NP plano, sendo ou não escrito com um espaço" [13], como pode ser observado na Figura 26.

A estrutura foi reproduzida pelo *script* desenvolvido, mas o SP não foi capaz de processá-lo. Num momento futuro, será investigado o motivo de tal comportamento. A priori, fez-se necessário descartar tais sentenças. Isso dá um total de 149 sentenças, aprox. 3,5% do *dataset* completo.

[13] *Percent is simply a flat NP, whether or not it is written with a space.* Tradução própria.

CF77-4 “Cerca de 72% dos empresários da construção querem que o próprio setor negocie a conversão dos contratos para a URV, enquanto 28% desejam que o governo estabeleça as regras.”

A1
STA:fcl
=SUBJ:np
==>N:ap
===>A:adv(’cerca_de’) Cerca_de
===H:num(’72’ <card> M P) 72
==**H:n(’%’ M P)** %
==N<:pp
===H:prp(’de’ <sam->) de
===P<:np
====>N:art(’o’ <-sam> <artd> M P) os
====H:n(’empresário’ M P) empresários
...

Figura 25 – Exemplo de marcação de porcentagem pelo Bosque, formato Árvores Deitadas. Adaptado de (FREITAS, 2006, p 115)

(NP 15 **percent**)
(NP 15 **per cent**)
(NP 8.45 %)

Figura 26 – Exemplo de representação de porcentagem para o PTB. Adaptado de Bies et al. (1995, p 308)

3.3.3 Problemas com pontuação

O Bosque tem uma política de tratamento de símbolos mais parecida com a do PTB. Porém, envolve uma pluralidade maior de possíveis sinais, além de usar sinais não convencionais, como «, por exemplo. A Tabela 9 mostra os símbolos utilizados, além de suas respectivas frequências de ocorrência no *treebank*.

Simbolo	Frequência	Simbolo	Frequência
!	59	;	208
’	45	?	78
,	4432	[	1
-	1	]	1
- -	103	{	453
.	3396	}	456
...	17	«	707
\	3	»	703

Tabela 9 – Tabela de símbolos presentes no CETEMFolha, e suas respectivas frequências de aparecimento.

O Bosque tem uma descrição extensa sobre o posicionamento de pontuações. Por (FREITAS, 2006, p 27), "A pontuação não tem qualquer informação morfossintática associada, embora possa ser um indicador de estatuto sintáctico, estando tão só indentada". A priori é uma política muito semelhante à do PTB, como vimos em 3.2.3.

Existem três casos distintos a serem considerados na hora de posicionar pontuações: inicio, final, e dentro de frases. Para pontuações em inicio e fim de frase, (FREITAS, 2006, p 28) "indentação ao mais alto nível de constituinte imediatamente abaixo da raiz". Para pontuações dentro de frase, temos dois casos, como delimitadores, e como separadores. Para delimitadores,

> "[É a] pontuação que delimita trechos de texto colocada ao mesmo nível dos nós mais altos correspondentes a esse trecho, quer seja um nó não terminal ou o um nó terminal. Estes casos incluem os seguintes tipos de pontuação: aspas, parênteses, vírgulas, travessões."

Para separadores, (FREITAS, 2006, p 30) "[É a] pontuação que separa trechos, colocada ao mesmo nível do trecho que se inicia a partir da pontuação que o separa do trecho anterior. Incluem-se neste caso a vírgula, ponto e vírgula, dois pontos, travessão". Por fim, (FREITAS, 2006, p 33):

> "A pontuação final permite identificar a frase como constituindo um elemento comunicativo [...], e por isso é considerada como fazendo parte integrante da frase. Por vezes, uma 'frase analisada' corresponde a uma sequência de 'frases' como funções principais (STA, QUE, EXC, e UTT)."

As sentenças geradas pelo *script* desenvolvido não eram processadas corretamente pelo SP, que acusa erro semelhante ao citado em 3.2.3, como pode ser visto na Figura 27. Assim como no CINTIL, removemos os símbolos das sentenças, para permitir a continuidade dos testes.

3.3.4 Problemas com CU (Coordenação)

Coordenação é uma das *tags* mais problemáticas do BOSQUE. Como destacam Wing e Baldridge (2006, p 4):

> "Cláusulas conjuntivas no Floresta nativo são do tipo CU, independentemente do tipo de constituinte conjunto. Isso faz com que a gramática aprendida do

```
  pars: Não explica como mas garante que não vai perder a próxima eleição em Ita
baiana
Parsing [len. 21]: Entregar- se , com a garantia de redução de+ a pena , seria u
m bom negócio '' , disse Maciel .
WARNING: Evaluation could not be performed due to gold/parsed yield mismatch.
  sizes: gold: 21 (transf) 21 (orig); parsed: 16 (transf) 21 (orig).
  gold: Entregar- se , com a garantia de redução de+ a pena , seria um bom negóc
io '' , disse Maciel .
  pars: Entregar- se com a garantia de redução de+ a pena seria um bom negócio d
isse Maciel
Parsing [len. 17]: Segundo eles , os laços com a aliança não vão afetar as relaç
ões com a Rússia .
WARNING: Evaluation could not be performed due to gold/parsed yield mismatch.
  sizes: gold: 17 (transf) 17 (orig); parsed: 15 (transf) 17 (orig).
  gold: Segundo eles , os laços com a aliança não vão afetar as relações com a R
ússia .
  pars: Segundo eles os laços com a aliança não vão afetar as relações com a Rús
sia
Parsing [len. 22]: Em+ a locadora Clean_Car a diária de+ o mesmo carro sai por R
$ 72 com limite de 200 quilômetros por dia .
WARNING: Evaluation could not be performed due to gold/parsed yield mismatch.
  sizes: gold: 22 (transf) 22 (orig); parsed: 21 (transf) 22 (orig).
  gold: Em+ a locadora Clean_Car a diária de+ o mesmo carro sai por R$ 72 com li
mite de 200 quilômetros por dia .
```

Figura 27 – Erro de não casamento (*mismatch*) entre árvores ao executar o SP sobre o Bosque pré-treinado.

> *treebank* cometa erros como confundir conjunções de sintagmas nominais e conjunções sentenciais" [14].

O PTB tem suas próprias regras para determinar a estrutura de coordenações, reservando o Capítulo 7 de seu *Bracketing Guidelines*[15] (BIES et al., 1995, p 117). Seguindo tais regras, num primeiro momento de tradução, apenas marcamos as coordenações com a *tag* auxiliar _CU_. Depois, fizemos uma nova verificação, reconhecendo as sentenças onde essa marca aparece.

```
(H:cu
    (CJT:np
        (H:n:saúde:F_S::anr_np-def: saúde))
    (CO:conj-c:e:::: e)
    (CJT:np
        (H:n:educação:F_S::np-idf: educação)))))))
```

Figura 28 – Exemplo de uso do sintagma evidenciador de coordenação no Bosque. Fragmento da sentença CF5-2

Como dito em 3.2.1, que o PTB lida com a coordenação de basicamente três formas: palavras simples, palavras múltiplas e conjunções descontínuas. A *tag cu* define o sintagma que encabeça a coordenação. Isto pode ser melhor visto na Figura 28. Isto facilita o

[14] "*Conjoined clauses in the native Floresta are of type CU, regardless of the type of constituents being conjoined. This causes grammars learned from the treebank to make errors such as conflating noun phrase conjunctions and sentential conjunctions*". Tradução própria.

[15] Manual de Agrupamento

procedimento pois, deve-se então verificar se os sintagmas filhos possuem a mesma *tag*, e se são nós terminais ou não-terminais. Sendo terminais, devem ser impressos numa estrutura achatada (*flat structure*). Sendo não-terminais, de mesma categoria, a *tag* cu deve ser convertida para uma *tag* equivalente. Sendo sintagmas de categorias distintas, cu é convertido para UCP (*Unlike Coordinated Phrase*[16]).

CF766-6 E o Brasil?
(FRASECF766-6 (QUE:np (**CO:conj-c:e:: E**)
(>N:art:o:M_S::artd: o)
(H:prop:Brasil:M_S::: Brasil)
(?)))

Figura 29 – Exemplo árvore onde palavra marcada por conj-c (conjunção coordenativa) não implica em conjunção entre sentenças

Tem-se, também, que tomar cuidado com os casos de palavras simples, em *flat structure*. Por dois motivos: primeiro, não necessariamente uma palavra que, no Bosque, é marcada por conj-c (conjunção coordenativa), será um indicador de conjunção, efetivamente. Exemplo na sentença CF766-6, Figura 29. Nesses casos, a palavra deve ser marcada com a *tag* CC para o PTB. O segundo problema é a correta escrita da *flat structure*. O (BIES et al., 1995, p 117) mostra alguns exemplos possíveis. Empiricamente, notamos que a forma correta deve ser como na Figura 30, ou seja, o valor pós-conjunção deve ser destacado num novo sintagma.

(VP
(IN porque)
(VP
(VBP tem)
)
(NP
(NN gente)
)
(VP **comprando e**
(**VBG vendendo**)))

Figura 30 – Exemplo de como coordenações *single word* devem se comportar. Fragmento da conversão da sentença *CF400-2 Diretor do Banco Central não acredita que o real esteja valorizado porque «tem gente comprando e vendendo»*

[16] "Sintagma Coordenado Diferente". Tradução própria.

3.3.5 O par x e X

Ainda há, no *corpus*, o par de *tags*, X e x. X é uma *tag* do tipo função, e x uma *POS tag*. Na Bíblia Florestal (FREITAS; AFONSO, 2007), não há uma descrição formal para ambas. Apenas notou-se que nunca aparecem em par, mas quase sempre próximas. Num primeiro momento, a solução foi ignorar este par, descartando toda sentença/árvore em que aparecem. Mas uma revisão mostrou que seria uma grande perda desprezá-los, e isso nos levou a uma série de desdobramentos interessantes.

X:adv	1	CJT:x	480
X:cu	22	EXC:x	1
X:np	3	N<:x	18
X:pp	130	P<:x	1
		PIV:x	3
		UTT:x	2

Tabela 10 – Pares possíveis para as tags X, e frequência de aparecimento no CETEMFolha

Na Tabela 10, mostramos quais as ocorrências das *tags* X e x, em conjunto com seus possíveis pares (ou seja, "$(F : x)$ e $(X : f)$", sendo f e F *tags* quaisquer). Pode-se notar que, por mais que ambas sejam casadas com diversas etiquetas, no geral se associam à *tags* de conjunção. Uma hipótese, então, é:

> X e x são *tags* coringas. Elas ocupam o espaço da Função ou da Forma (respectivamente) quando a informação realmente relevante é apenas o seu par.

Isto gera um problema. Como descrito em 2.2.2.1, todo nó tem pelo menos o par F:f, e como foi dito no começo da seção, deu-se preferência às *tags forma* (por serem equivalentes às *POS tags*) em detrimento às outras. As *tags* x nos forçam a voltar a olhar para as *tags* irmãs. Ou seja, nos casos em que o nó possui uma *tag* x, F informará o valor a ser convertido. As *tags* F, então, necessitam de uma nova tabela, que pode ser vista na Tabela 11.

Tabela 11 – Tabela de conversão: BOSQUE para PTB (Funções relevantes)

Tag Original (Português)	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
>A	dependente em adjp ou advp (antecede o núcleo)	>A	371	Explicado em 3.3.6
A<	dependente em adjp ou advp (segue o núcleo)	A<	272	Explicado em 3.3.6
ACC	objecto directo (incluindo alguns tipos de se)	depende da *form_tag*	4315	Explicado em 3.3.8
ADVL	adjunto adverbial	depende da form	6032	Explicado em 3.3.8
CJT	elemento conjunto	_CJT_	3945	Explicado em 3.3.7
EXC	enunciado exclamativo	S	36	
H	núcleo	depende da form	40148	
KOMP<	complemento comparativo	_KOMP_	40	Explicado em 3.3.9
>N	adjunto adnominal (antecede o núcleo)	NP	14009	Dobra do NP por adjunto, como visto em Mioto, Silva e Lopes (2013, p 67) [17]
N<	adjunto adnominal (segue o núcleo)	NP	9208	Dobra do NP por adjunto, como visto em Mioto, Silva e Lopes (2013, p 67)

Continua na próxima página

[17] Mioto, Silva e Lopes (2013, p 67) descreve a dobra de sintagmas para adjuntos.
"[...] existem ainda sintagmas que são licensiados numa sentençça sem serem complemento ou especificador de um núcleo. São os chamados **adjuntos**. [...] Um adjunto [...] é um sintagma que está apenas contido na projeção máxima de um núcleo. [...] A representação do adjunto sempre implica a duplicação da categoria com a qual ele está relacionado. Desta forma, o adjunto vai ser dominado apenas pelo segmento de cima da categoria duplicada."

Tabela 11 – *Continuação da página anterior*

Tag Original (Português)	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
OC	predicativo do objeto	depende da form	102	Explicado em 3.3.8
>P	dependente da preposição	PP	71	Por observação, e por Mioto, Silva e Lopes (2013, p 67)
P	predicador	VP	8053	Pela Freitas (2006, p 60), O predicador é sempre de natureza verbal e, por isso, pode exibir apenas formas verbais
P<	argumento de preposição	PP	11574	Por observação, e por Mioto, Silva e Lopes (2013, p 67)
PIV	objecto preposicional	PP	1097	
QUE	enunciado interrogativo	S	64	
SC	predicativo do sujeito	VP	1254	
STA	enunciado declarativo	S	3683	
UTT	enunciado	S	468	

A Tabela 11 é um fragmento da Tabela 23, que pode ser encontrada nos apêndices. Esta última possui, efetivamente, todas as *tags* de função, seus nomes, possíveis conversões etc. Diversas delas não serão utilizadas neste trabalho, portanto não necessitam ser abordadas nesta Seção.

Nem todas as *tags* de função são explícitas quanto ao sintagma que elas definem. Ou seja, uma informação de sintagma precisa seria obtida pela *tag* f, que no caso é x, que não tem valor sintagmático, e o PTB não prevê sintagmas não marcados, ou etiquetas coringa. Essas funções, e as estratégias utilizadas para resolver tais problemas, serão abordadas nas próximas seções.

3.3.6 Problemas com >A e A<

Vemos em (FREITAS, 2006, p 13) "A indicação de '>' e '<' presentes nas funções acima, reflectem a posição do dependente face ao núcleo, como setas direccionadas para o núcleo cuja natureza está indicada por letras maiúsculas". Vemos também que "Por outro lado, >A significa que o dependente aponta para o núcleo de natureza adjectival ou adverbial ('A') que se encontra à direita do dependente ('>')".

A possível ambiguidade da sentença anterior é resolvida em (FREITAS, 2006, p 58), no qual se lê "As mesmas etiquetas, A< e >A, como se pode constatar, estão presentes tanto nos sintagmas adjectivais como nos sintagmas adverbiais enquanto modificadoras de adjectivos ou advérbios". Ou seja, para a correta conversão de >A (por exemplo) para ADVP, ou ADJP, é necessário olhar o nó irmão, e verificar o seu núcleo. Este sendo adjetivo ou sintagma adjetival, >A se torna ADVP, e análogo para advérbios. O que fazer em casos como da sentença CF761-3 (Figura 31), em que o nó irmão de >A é um advérbio, mas o núcleo do sintagma é adjetivo, se torna um novo problema.

```
...
(SC:adjp
    (>A:x (X:pp
            (H:prp:de:::: de)
            (P<:np
                (>N:num:30:M_P::card: 30)
                (H:n:%:M_P::anr_np-count:anr_np-def: %)))
            (X:pp
                (H:prp:a:::: a)
                (P<:np
                    (>N:num:40:M_P::card: 40)
                    (H:n:%:M_P::anr_np-count:anr_np-def: %))))
    (>A:adv:mais:_::KOMP:quant: mais)
    (H:adj:rápido:F_P::: rápidas)
    (KOMP<:acl

...
```

Figura 31 – Exemplo de estrutura ambígua de >A. Fragmento da árvore da sentença CF761-3 "As novas impressoras a laser da HP vêm com um novo padrão de velocidade 12 páginas por minuto (ppm) e são de 30% a 40% mais rápidas que as da geração anterior".

Para este trabalho, os nós não conservam as informações de função, apenas de forma. Assim sendo, >A será transduzido de acordo com o primeiro nó válido na direção indicada por > ou <.

3.3.7 Problemas com CJT (Conjunção)

Ainda sobre sintagmas evidenciadores da relação de coordenação (visto em 3.3.4), (FREITAS, 2006, p 20) descreve:

> "A sua estrutura interna evidencia também a relação de coordenação: duas ou mais partes coordenadas como função e uma ou mais conjunções coordenativas, que podem ou não estar presentes. A forma de cada uma das partes coordenadas segue os princípios gerais dos sintagmas não verbais, isto é, dependendo da sua estrutura interna e em particular da função dos dependentes do núcleo do sintagma, ou dos sintagmas verbais".

As "partes coordenadas" citadas são nós cuja função é CJT. Como é melhor explorado em (FREITAS, 2006, p 96) "nós dependentes daquele, terminais ou não terminais, ao mesmo nível que correspondem às partes coordenadas $(CJT : forma)$".

Quando tal tipo de função ocorre para um nó, a solução de transdução costuma ser simples: basta utilizar o valor de forma. O problema, claro, é quando a tag *forma* é "x", a etiqueta "coringa". Nestes casos, empiricamente, notou-se uma frequência alta de sentenças internas à $(CJT : x)$ que possuem valor sintático de sintagma verbal (VP). Portanto, decidimos que os casos $(CJT : x)$ seriam convertidos para VP.

CJT&ACC:np	1	CJT:fcl	596
CJT&ADVL:pp	3	CJT:icl	123
CJT&PASS:pp	1	CJT:n-adj	12
CJT&PRED:adjp	2	CJT:np	1990
CJT:acl	12	CJT:pp	392
CJT:adjp	288	CJT:v-ger	2
CJT:advp	24	CJT:v-pcp	22
CJT:cu	6	CJT:x	480

Tabela 12 – Possíveis combinações de CJT.

3.3.8 Problemas com ACL (Orações Averbais)

(FREITAS; AFONSO, 2007), "Finalmente, na oração averbal [18] [19] , o verbo não está presente, mas normalmente estas orações são encabeçadas por uma conjunção subordinativa que indica a natureza oracional do período". Ao passo que as *tags* fcl e icl (orações finitas e não-finitas) podemos afirmar serem conversíveis para VP, a *tag* acl nos obriga a fazer conversões para a forma de oração subordinada (BIES et al., 1995, p 172), porém também

[18] Note-se que, em (FREITAS, 2006, p 12), acl é grifada como "oração deverbal"
[19] "the crawling chaos...". Lovecraft (1920)

não de forma assertiva. Pois, como vemos na Tabela 13, acl pode se ligar a qualquer forma sintagmática, inclusive início de orações. Porém, observar a função costuma dar uma informação bastante definitiva, conversão que adotamos nos casos de ADVL:acl, EXC:acl, N<:acl, N<PRED:acl, QUE:acl, STA:acl, UTT:acl. Para casos como ACC:acl e OC:acl, foram tomadas opções arbitrárias: foram convertidas para NP. Faltam dois casos então, KOMP< e CJT (novamente).

ACC:acl	1	N<PRED:acl	6
ADVL:acl	78	OC:acl	31
CJT:acl	12	QUE:acl	1
EXC:acl	4	SC:acl	15
KOMP<:acl	44	STA:acl	3
N<:acl	40	UTT:acl	63

Tabela 13 – Possíveis combinações da *tag acl*, e frequência de ocorrência no CETEMFolha

3.3.9 Problemas com KOMP< (Complementos)

Esta etiqueta é definida, por (FREITAS, 2006, p 57) como "KOMP<, segundo termo de uma oração comparativa ou oração consecutiva". E em (FREITAS, 2006, p 116):

> "Em termos de representação em árvores, a particularidade é o segundo termo de comparação ser etiquetado com KOMP< (etiqueta de sintagma) ao mesmo nível de constituintes que o adjectivo ou advérbio e o seu dependente. [...] KOMP< terá como forma 'fcl', no caso de o verbo principal estar expresso ou 'acl', no caso de o verbo principal estar omisso".

É reservado o Capítulo 22 do (BIES et al., 1995, p 284) para o estudo de comparativos. Nele, vemos:

> "*than*, *that*, or *as* é categorizado tanto um PP ou um SBAR, e uma certa quantidade de variações existem na escolha de PP, ou SBAR. SBAR é usado quando o resto do sintagma que contem *than/that/as* é uma sentença flexionada, ou quando contém um sujeito. PP é, no geral, quando o resto do sintagma que contem *than/that/as* é um constituinte simples [...]" [20].

Dois exemplos simples podem ser vistos na Figura 32. Foi necessário, então, repetir tal comportamento na transdução.

[20] "*The than, that, or as is bracketed as either a PP or an SBAR, and a certain amount of variations exists in the choice of PP, or SBAR. SBAR is used when the rest of the than/that/as-phrase is a tensed sentence, or when it contains a subject. PP is in general when the rest of the than/that/as-phrase is a single constituent [...]*". Tradução própria.

```
...
(PP than/as
    (xP rest of phrase))


(SBAR than/that/as
        (S rest of phrase))

...
```

Figura 32 – Exemplos de configurações das sentenças comparativas no PTB. Adaptado de (BIES et al., 1995, p 284)

3.3.10 Problemas com CJT:acl

Definitivamente, a mais desafiadora das conversões. Sentenças averbais com valor de conjunção. Não existe uma forma precisa de converter este par.

Notou-se também que observar os nós filhos, ou mesmo os pais, não nos dariam informações precisas que nos permitisse afirmar qual seria uma transdução interessante, que fosse reconhecível tanto por PTB como por SP. Ser arbitrário, e defini-lo como VP ou NP, por exemplo, causaria muitos problemas. Como trabalho futuro, caberá o estudo deste par. Por ora, decidiu-se transduzi-los para "S" (marcador de Sentença).

3.3.11 Problemas com o Bosque

Dentre as dificuldades na transdução Bosque/PTB, alguns erros no *dataset* merecem ser frisados. Como já dito, o Bosque originalmente é distribuído no formato Árvores Deitadas, mas possui uma distribuição em formato *Penn Treebank*. A observação nos mostra que, na prática, apenas foram substituídos os símbolos de endentação pelos parênteses do PTB. Destacam-se alguns casos.

P.vp - Na sentença CF624-3, a separação de etiquetas está incorreta. Ao invés de estar escrito "P:vp", está grifado "P.vp", como visto na Figura 33.

Nó terminal fechado de forma incorreta - Alguns nós que necessariamente são terminais (como determinantes, ou pronomes) ocorrem fechados de forma inconsistente. A correção foi necessária. A sentença CF624-3 contém um exemplo, que pode ser visto na Figura 34.

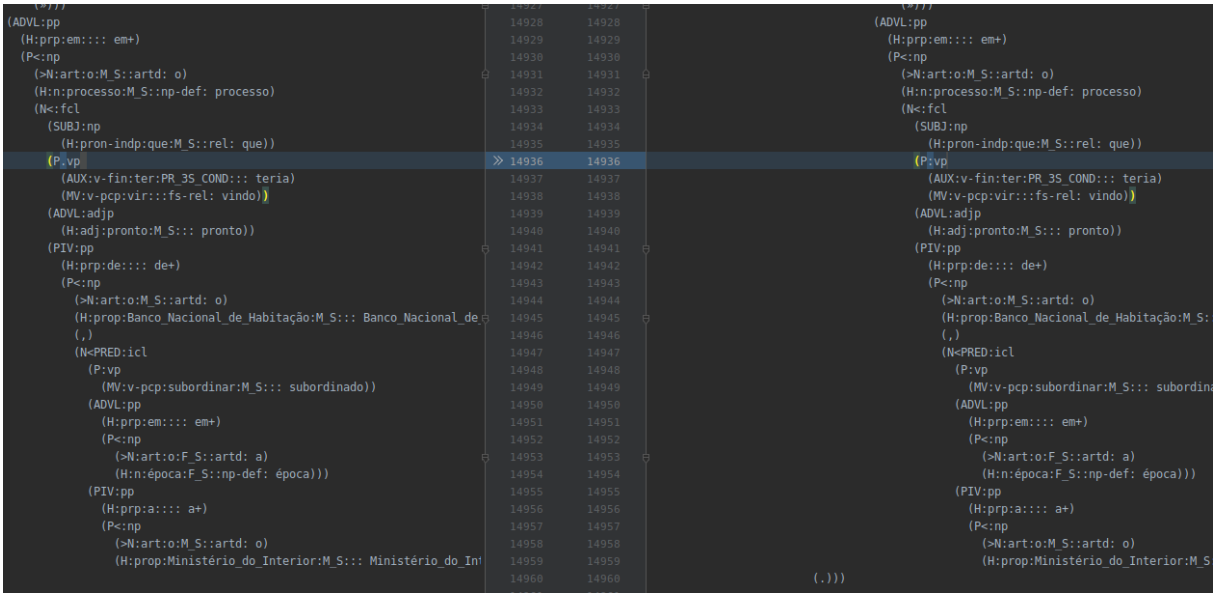


Figura 33 – Erro na marcação do par P:vp

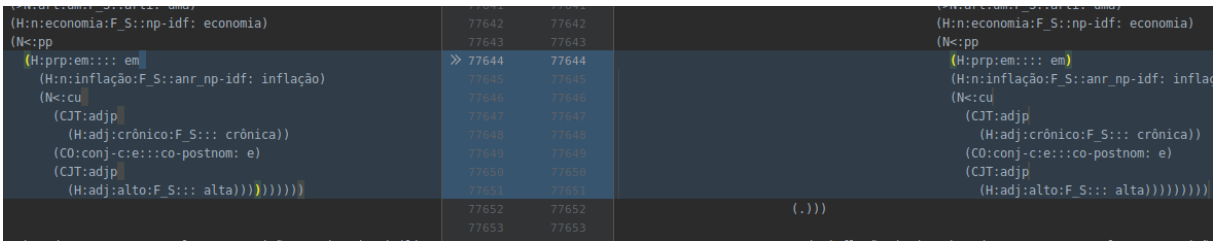


Figura 34 – Erro no fecho do nó H:prp

Símbolos marcados incorretamente - Bosque apresenta símbolos entre parentes, sem etiquetas. Isto ocorre em duas sentenças. Um exemplo é na sentença CF322-3, como pode ser visto na Figura 35, com o parênteses abrindo para o símbolo, mas sem o fecho na sequência.

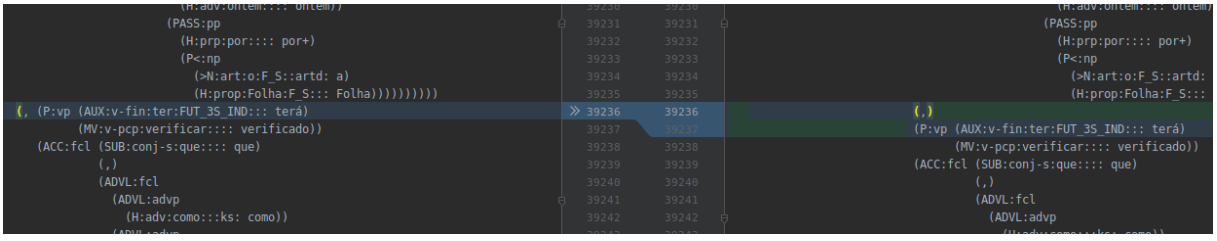


Figura 35 – Erro na marcação de símbolos

3.4 Treinamentos

Com as *tags* convertidas, foram feitos os treinamentos. O ato do treino é relativamente simples: deve-se, através do terminal do seu sistema, navegar até o diretório onde se

encontra o *stanford-parser.jar*, o arquivo que contém os *parsers* do SP a serem utilizados. Os comandos utilizados podem ser vistos em A.3.

Note que usamos apenas 1014 arquivos nesta demonstração, que é um fração de 10% dos arquivos/árvores do CINTIL disponíveis.

Para melhor verificação dos resultados, foi utilizado o método de *10-fold cross-validation*. Este método, como explicado por James et al. (2013),

> "[Esta abordagem] envolve dividir aleatoriamente o conjunto de observações em k grupos, ou dobras, de tamanhos aproximadamente iguais. O primeiro grupo é tratado como conjunto de validação, e o método se encaixa nos k – 1 grupos restantes." [21]

O *corpora* foi dividido em 10 partes de 1014 sentenças. Foi feito o treinamento com nove partes, deixando a décima parte restante para o treino.

A Figura 36 demonstra o uso do *10-fold* neste projeto.

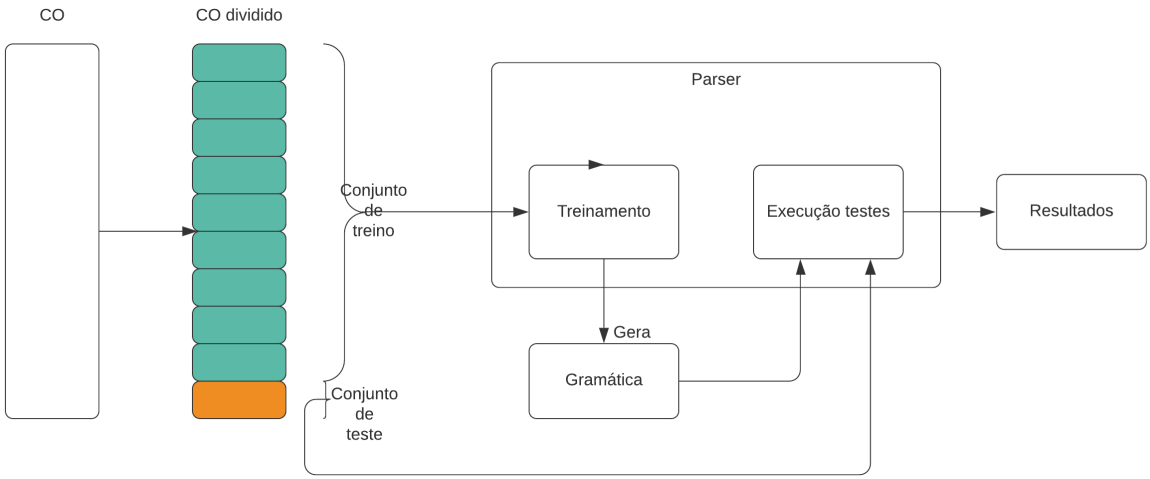


Figura 36 – Fluxograma *10-fold validation*

Na sequência, foi executado o treinamento com as partes restantes. De maneira análoga, no mesmo diretório (ou seja, fazendo testes sobre o CINTIL transduzido), foi utilizado o comando A.3.

Os treinamentos foram feitos alternando a parte de teste e as partes de treino, totalizando a criação de dez gramáticas, e dez resultados de treinos distintos.

[21] "*[. . .] involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k – 1 folds*". Tradução própria.

O resultado total dos testes resultou na extensa Tabela 19, que pode ser vista nos apêndices. E os comentários dos resultados pode ser encontrado em 4.1.

De modo análogo ao CINTIL, fizemos o treinamento baseado num comando simples, como visto no código B.2.

Também foi utilizado o *10-fold validation* para o Bosque. Porém, os testes foram feitos majoritariamente com o CETEMFolha.

O resultado completo dos testes pode ser visto na Tabela 22, nos apêndices. Os comentários dos resultados pode ser visto em 4.2.

Os resultados serão apresentados no Capítulo 4.

4 Avaliações

Daqui em diante, mostraremos os resultados dos experimentos, separados por cada um dos *corpus* base. Ao final desta Seção, será feito um debate acerca dos resultados alcançados.

Os testes foram feitos utilizando um computador Intel Core i7-7700HQ de 64bits, com 16GB de memória RAM.

4.1 Avaliação do CINTIL

Nesta seção, demonstraremos os resultados obtidos com a transdução do CINTIL.

4.1.1 Treinamento

Na Tabela 14, podemos ver o relatório de cada treinamento do SP com cada *fold* do CINTIL transduzido.

Grammar	States	Tags	Words	UnaryR	BinaryR	Taggings
1	101	12	2775	52	359	2883
2	118	13	3174	61	423	3292
3	113	13	3142	62	414	3268
4	119	13	3188	61	419	3316
5	121	13	3198	60	424	3332
6	120	13	3278	62	425	3413
7	119	13	3259	62	427	3387
8	119	13	3246	61	422	3362
9	118	13	3188	60	415	3314
10	120	13	3246	62	424	3372

Tabela 14 – Resultados dos treinamentos do CINTIL, para os 10 *folds*

Pode-se notar que, além do primeiro *fold* (que abrange os últimos nove décimos do *treeebank* para treinamento, e reserva o primeiro para testes), os resultados são bastante semelhantes. Deduz-se, então, que o primeiro décimo do *dataset* temuma expressividade maior no processo de treino. O sexto *fold* (reserva a 6ª parte para testes) parece ser o com melhores resultados gerais, como pode ser visto na Figura 37.

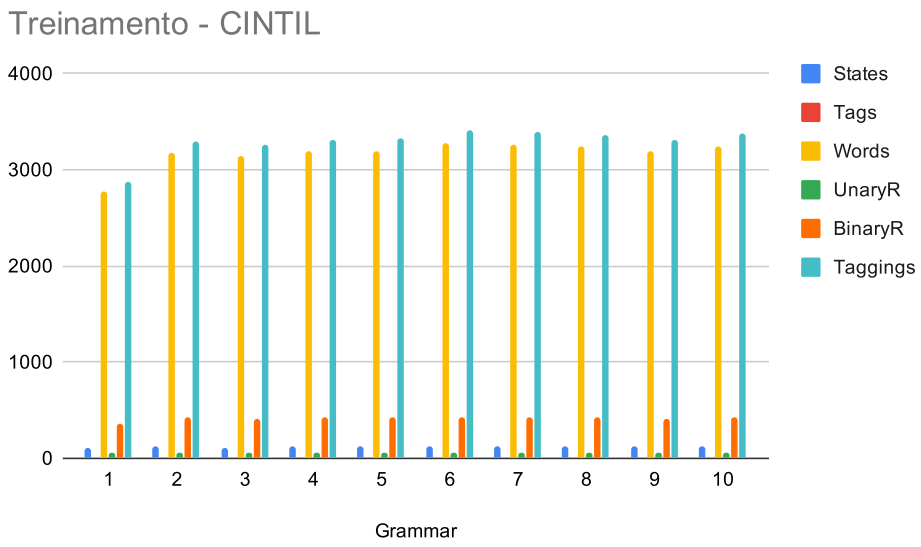


Figura 37 – Gráfico de resultados do treinamento do LexicalizedParser, usando o CINTIL transduzido

As colunas necessitam de explicações próprias. O FAQ do SP é pouco informativo, e o do CoreNLP possui a mesma dificuldade.

Pelo código-fonte do SP[1], *States* representa a quantidade de Índices de Estado. Inferiu-se que é a quantidade de estados de transição da gramática gerada pelo treino.

Tags, por sua vez, representa a quantidade de Índices de *Tags*. Seria a quantidade de *Tags* registradas durante o treino (note que o número varia pouco).

Words, de forma análoga, representa a quantidade de Índices de Palavras. Deduz-se que, a quantidade de palavras distintas verificadas pelo treinamento.

UnaryR e *BinaryR* correspondem, respectivamente, às quantidades de regras das gramáticas Unária e Binária. A descrição de suas classes é, na sequencia, "Mantém indexação eficiente das regras unárias da gramática" [2] e "Mantém indexação eficiente das regras binárias da gramática" [3] . Pelo Javadoc do SP[4],

- Gramática Unária - consiste em regras de reescrita unárias, uma por linha, cada qual na forma A -> B, seguida pela probabilidade log normalizada; [5]
- Gramática Binária - consiste em regras de reescrita binárias, uma por

[1] A referência foi a classe LexicalizedParser.java, que está disponível em <https://github.com/chbrown/stanford-parser/blob/master/edu/stanford/nlp/parser/lexparser/LexicalizedParser.java>

[2] "*Maintains efficient indexing of unary grammar rules*"

[3] "*Maintains efficient indexing of binary grammar rules*"

[4] <https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/parser/lexparser/package-summary.html>

[5] *Unary Grammar - consists of unary rewrite rules, one per line, each of which is of the form A -> B, followed by the normalized log probability.*

> linha, cada qual na forma A -> B C, seguida pela probabilidade log normalizada. [6]

Cabe frisar que tal modelo, utilizando regras unárias e binárias, segue o padrão da Forma Normal de Chomsky. Por (MANNING; SCHÜTZE, 1999, p 389), Gramáticas na Forma Normal de Chomsky possuem apenas regras binárias e unárias, na forma:

$$\begin{aligned} N^i &\rightarrow N^j N^K \\ N^i &\rightarrow w^j \end{aligned} \tag{4.1}$$

Onde N^i se refere à nós não-terminais (para este trabalho, nós internos das árvores e raiz), e w^i se refere aos nós terminais (para este trabalho, nós contendo palavras).

Por fim, *Taggings* se refere a "o número de regras (etiquetas reescritas como palavras) no *Lexicon*" [7]. *Lexicon* é uma interface do software, descrita como:

> "Uma interface entre *lexicons* e o *lexparser*. Sua responsabilidade primária é prover uma probabilidade condicional P(palavra|etiqueta), que é preenchida pelo método *#score*. Dentro do *lexparser*, *Strings* são representadas canonicamente, e etiquetas e palavras são geralmente representadas por inteiros." [8]

4.1.2 Coleta de Resultados

Nos apêndices, a Tabela 19 traz os resultados completos dos testes com o CINTIL. Nesta seção serão utilizados um recorte dos dados.

Começando pelos dados da PCFG interna ao LexicalizedParser, podemos ver o seu resultado na Tabela 15, e na Figura 38.

A média da *F1-Score* é 59.304%, e seu desvio padrão é de aprox. 4.208%.

Fica claro que a simples transdução de árvores da língua portuguesa para os padrões da língua inglesa, conservando as palavras, é insuficiente para o bom resultado do *parser*. Que se pese que as características que precisaram ser removidas, como pontuações, também tiveram elevada influencia nestes índices.

---

[6] *Binary Grammar - consists of binary rewrite rules, one per line, each of which is of the form A -> B C, followed by the normalized log probability.*

[7] *[...]the number of rules (tag rewrites as word) in the Lexicon.* Tradução própria.

[8] "*An interface for lexicons interfacing to lexparser. Its primary responsibility is to provide a conditional probability P(word|tag), which is fulfilled by the #score method. Inside the lexparser, Strings are interned and tags and words are usually represented as integers.*" Tradução própria.

pcfg LP/LR	LP	LR	F1
fold 1	58.09	65.99	61.79
fold 2	62.52	62.32	62.42
fold 3	55.69	54.58	55.13
fold 4	54.03	51.31	52.63
fold 5	61.48	64.64	63.02
fold 6	58.81	57.61	58.21
fold 7	62.51	63.67	63.09
fold 8	53.03	54.05	53.54
fold 9	61.76	57.5	59.56
fold 10	63.32	63.98	63.65

Tabela 15 – Resultados do treinamento da PCFG do SP, usando dados do CINTIL

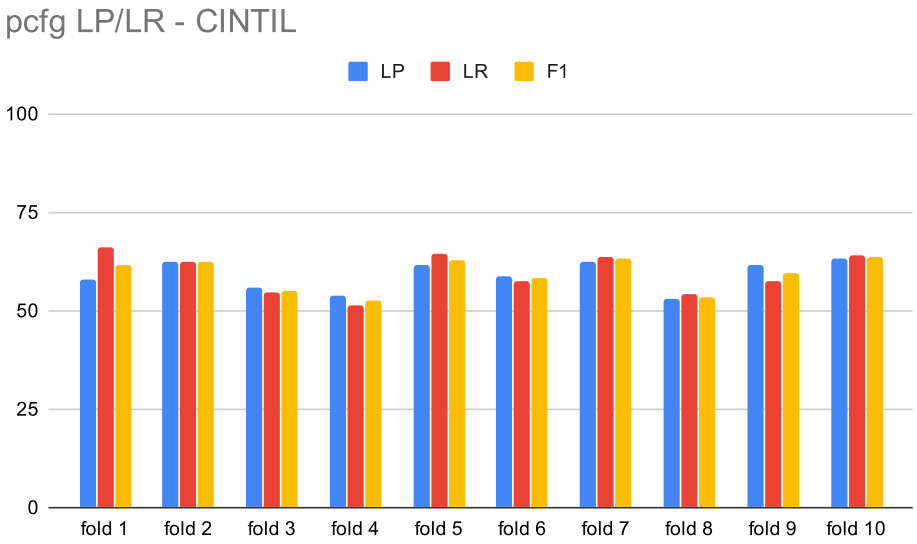


Figura 38 – Gráfico de resultados dos testes do PCFG do LexicalizedParser, usando o CINTIL transduzido

4.1.3 Análise de Erro dos treinamentos do SP com dados transduzidos do CINTIL

Foram separadas alguns exemplos de sentenças transduzidas, classificadas pelo *Stanford Parser*. Serão exibidas as sentenças já em formato de árvore. Serão feitos comparativos entre as árvores das sentenças originais, em relação às árvores resultantes dos processo de transdução. Para incrementar o debate, também será demonstrada a árvore gerada pelo *Stanford Parser* treinado com as gramáticas geradas por este trabalho. Para as análises a seguir, utilizou-se a gramática gerada a partir do sexto *fold*. As imagens foram produzidas no *website* jsSyntaxTree[9].

[9] <http://www.ironcreek.net/syntaxtree/>

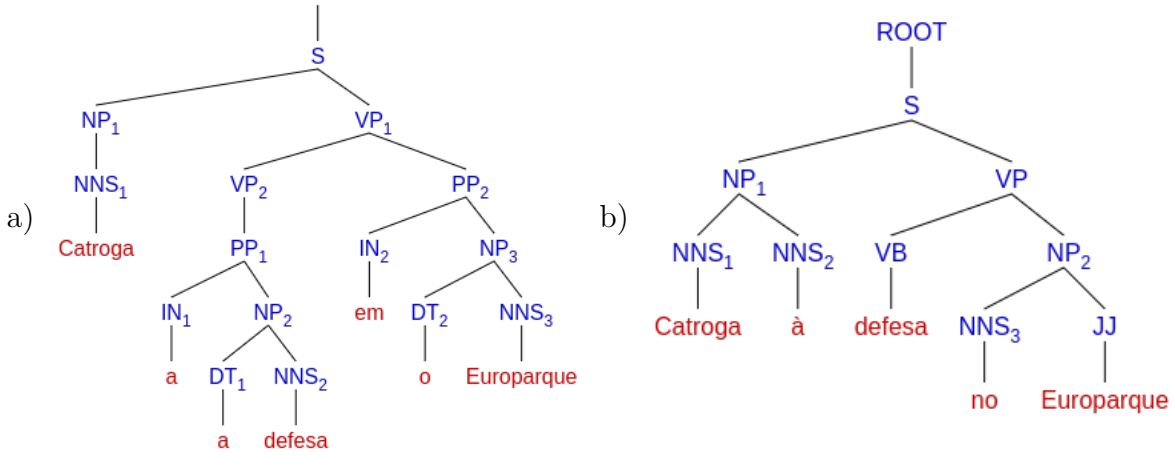


Figura 39 – Estudo da sentença eCTMP-000647/78121, “Catroga à defesa no Europarque”, que originalmente não possui nenhuma pontuação. Em a), vemos a sentença no seu formato original no CINTIL. Em b), o resultado de sua transdução

Na Figura 39, vemos o resultado da classificação de uma sentença originalmente sem pontuações. Se, por um lado, de fato a falta de pontuação ajudou no *parsing*, as contrações de preposição (“à”, “no”) não são identificados. Neste momento, é importante lembrar que nenhuma modificação foi feita no Stanford Parser. Logo, não afetamos seu modelo de linguagem padrão, responsável por este tipo de tratamento

Na Figura 40 podem ser vistas as problemáticas do tratamento de conjunções.

A confusão gerada pela falta de treinamento sobre pontuações permanece, sendo uma das maiores fontes de erros. O VP_3 “não aconteça” se torna um $ADJP$, considerando como núcleo o advérbio “não”. O esforço de manter a estrutura de conjunções, visto na Seção 3.2.1 é desfeito, convertendo o sintagma solto em *flat structure* “para que” se converte num PP. Vale notar que ele não se torna um sintagma conjuntivo (CONJP), e “que” não é marcado como CC, que é a tendência tradicional do *parser*. Também, apesar da tendência da língua inglesa, que costuma ter o núcleo do sintagma posicionado mais à direita (CHARNIAK, 1997, p 40), isso não ocorre nem no PP, nem no ADJP.

Na Figura 41, pode-se verificar as alterações realizadas sobre o sintagma CP, como visto na Seção 3.2.2. O sintagma em questão está acompanhando bem a estrutura da árvore original, e da árvore transduzida.

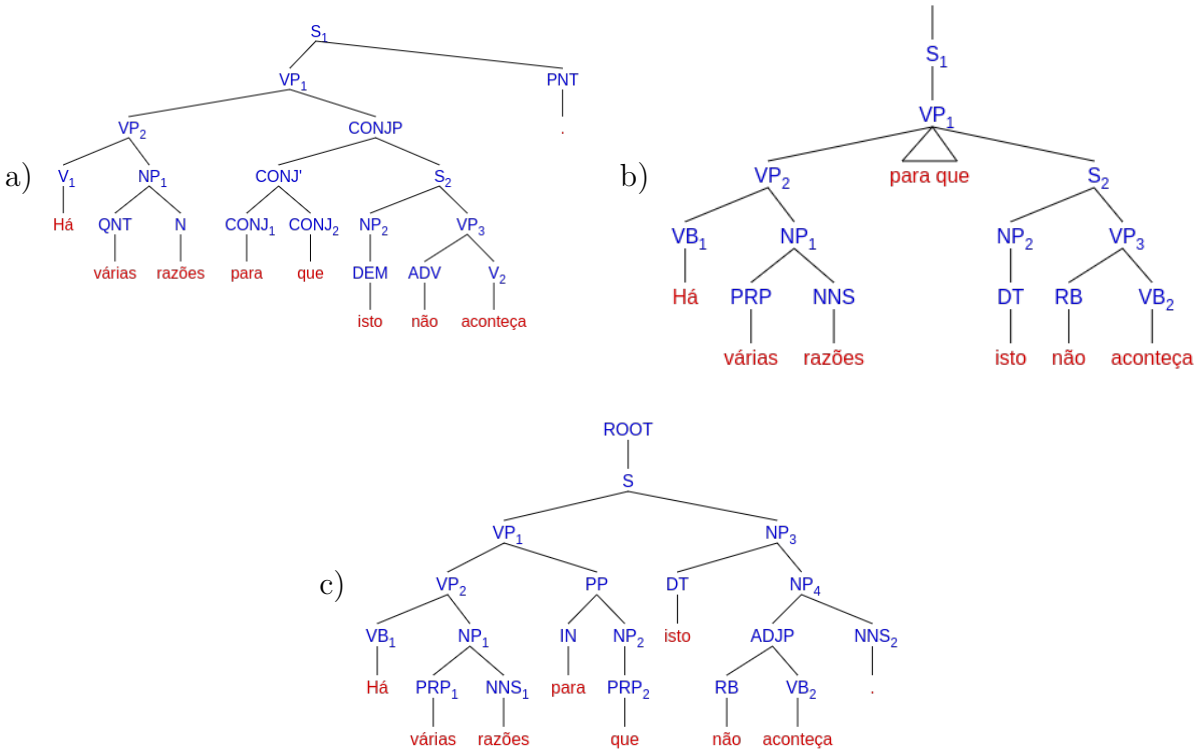


Figura 40 – Estudo da sentença eCTMP-001150/117736, "Há várias razões para que isto não aconteça.", que possui CONJP internamente. Em a), temos a árvore como se apresenta originalmente no CINTIL. Em b), temos a mesma sentença, pós transdução. Em c), temos o resultado da classificação do SP, utilizando a gramática gerada neste trabalho (após o processo de transdução)

Por fim, a Figura 42 possui um exemplo utilizando vírgulas. Pode-se notar a dificuldade léxica, onde palavras como "vistos" e "adiadas" são confundidos. Acredita-se que, se a contração da palavra "pelos" fosse detectada, a presença do Determinante resolveria tal problema. Os erros de marcação destes terminais afetam toda a marcação da árvore acima. Sem o treino utilizando pontuações, tais símbolos se tornam ou substantivos (NNS), ou adjetivos (JJ). Note-se também que, como demonstrado nas Figuras 42 e 40, o mesmo sinal pode receber classificações diferentes em sentenças diferentes.

4.2 Avaliação do BOSQUE

Nesta Seção, serão mostrados os resultados das execuções baseadas no BOSQUE.

4.2.1 Treinamento

Na Tabela 16, vemos os resultados dos treinamentos do SP sobre Bosque transduzido por este trabalho. Note que os campos são equivalentes aos explicados em 4.1.1.

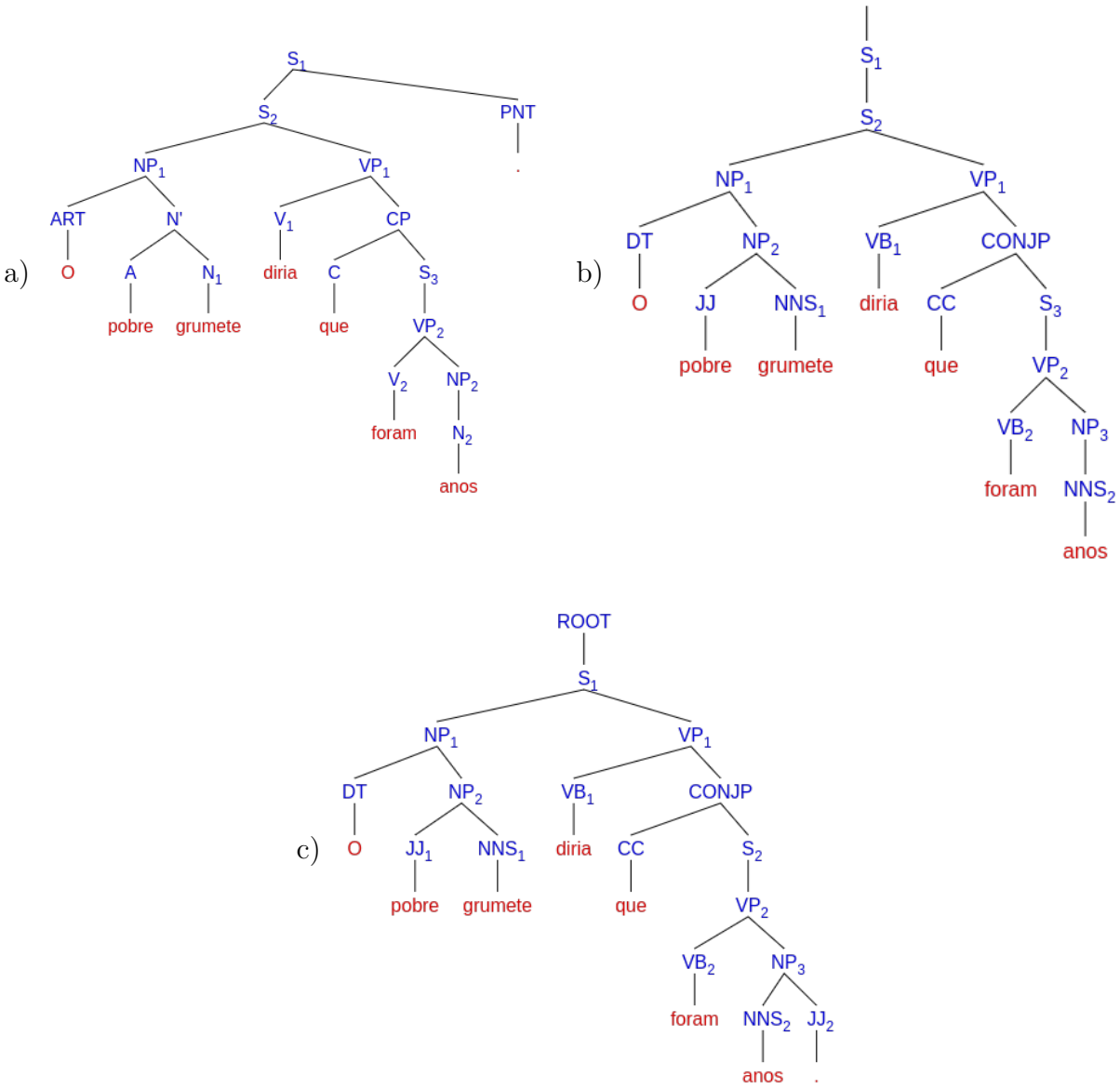


Figura 41 – Estudo da sentença eCTMP-000694/81773, "O pobre grumete diria que foram anos.", que possui CP internamente. Em a), temos a árvore como se apresenta originalmente no CINTIL. Em b), temos a mesma sentença, pós transdução. Em c), temos o resultado da classificação do SP, utilizando a gramática gerada neste trabalho (após o processo de transdução)

A Figura 43 traz a visualização desses resultados. Pode-se observar que as execuções do SP utilizando dados transduzidos do BOSQUE tornaram as gramáticas mais robustas que as vistas na Seção 4.1.1.

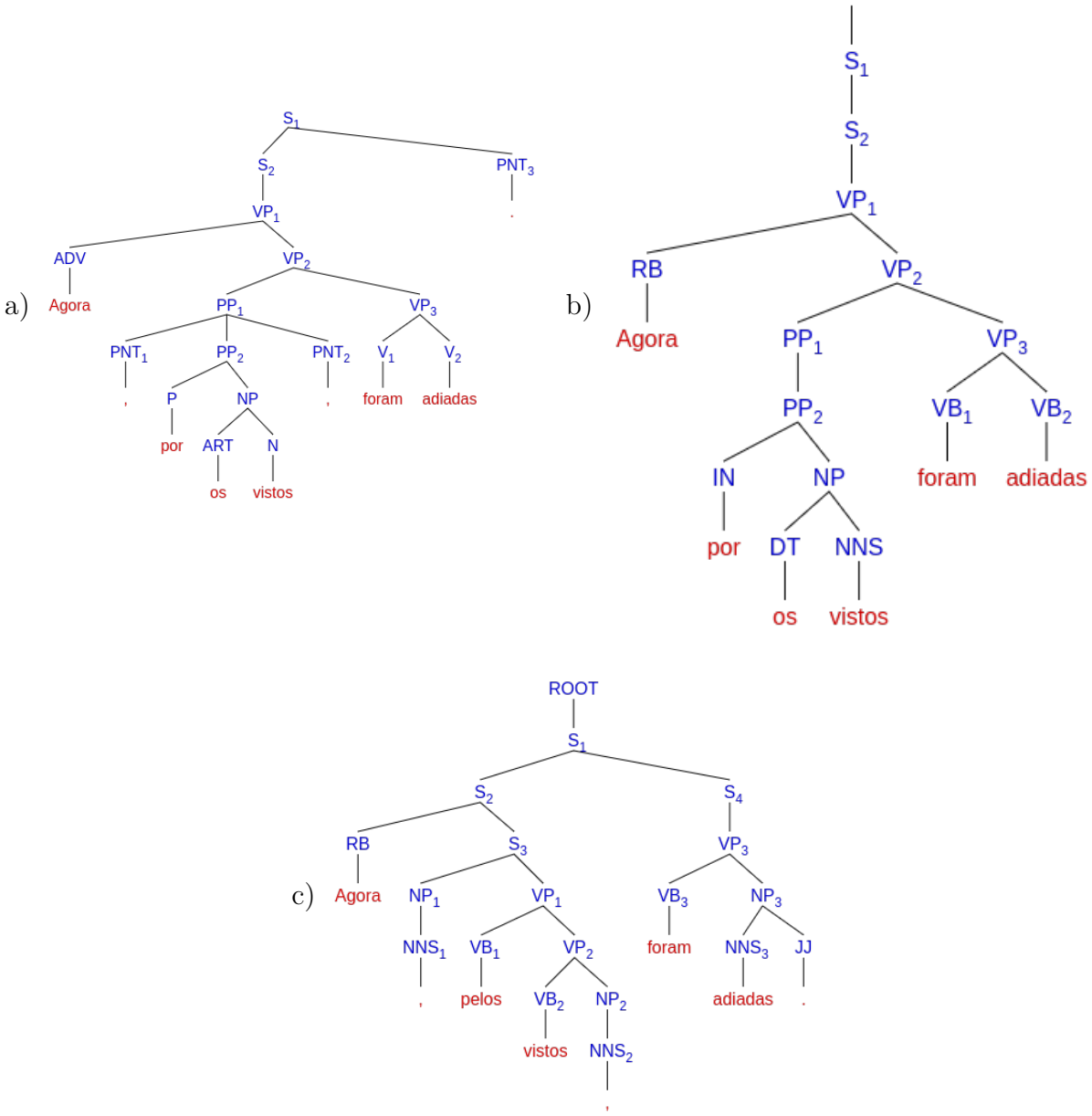


Figura 42 – Estudo da sentença eCTMP-001597/153293, "Agora, pelos vistos, foram adiadas.", que possui vírgulas. Em a), temos a árvore como se apresenta originalmente no CINTIL. Em b), temos a mesma sentença, pós transdução. Em c), temos o resultado da classificação do SP, utilizando a gramática gerada neste trabalho (após o processo de transdução)

4.2.2 Coleta de Resultados

Na Tabela 17, vemos o resultados dos testes utilizando o SP sobre o Bosque transduzido.

A média da *F1-Score* é de 49.551%, e tem o desvio padrão de aprox. 1.992%. Apenas

Grammar	States	Tags	Words	UnaryR	BinaryR	Taggings
1	1688	19	13641	136	4443	14105
2	1668	19	13786	130	4404	14254
3	1650	19	13655	138	4403	14117
4	1672	19	13700	126	4421	14152
5	1677	19	13691	137	4454	14159
6	1694	19	13717	139	4457	14173
7	1691	19	13739	138	4440	14208
8	1702	18	13685	138	4453	14151
9	1642	18	13744	132	4402	14205
10	1712	19	13767	139	4505	14244

Tabela 16 – Resultados dos treinamentos do Bosque, para os 10 *folds*

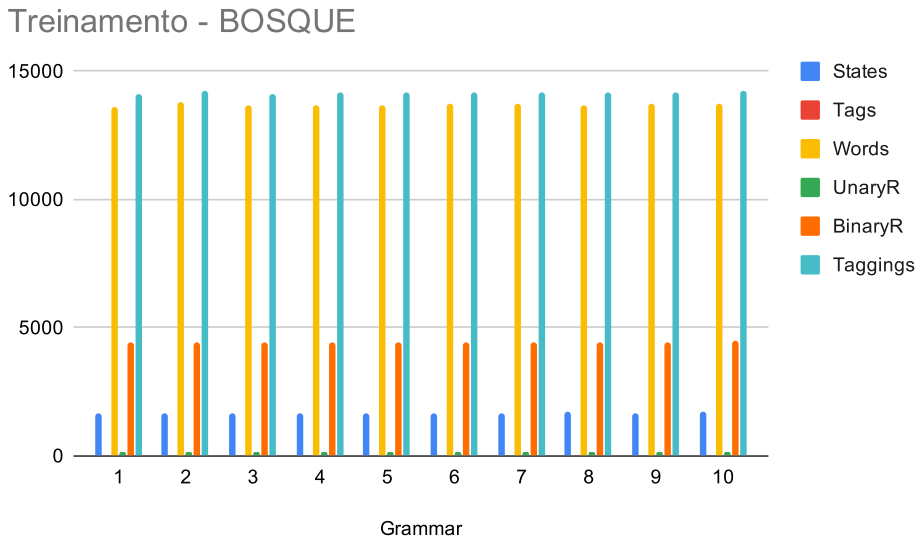


Figura 43 – Gráfico de resultados do treinamento do LexicalizedParser, usando o BOSQUE transduzido

pcfg LP/LR	LP	LR	F1
fold 1	51.61	47.9	49.69
fold 2	51.78	47.68	49.64
fold 3	49.25	45.65	47.38
fold 4	52.55	49.05	50.74
fold 5	51.03	46.05	48.41
fold 6	52.8	49.15	50.91
fold 7	48.86	45.21	46.97
fold 8	50.1	45.81	47.86
fold 9	52.28	48.49	50.32
fold 10	55.37	51.91	53.59

Tabela 17 – Resultados do treinamento da PCFG do SP, usando dados do BOSQUE

os dois últimos *folds* superam os 50%.

Nota-se, portanto, que apesar de números mais robustos ao produzir as gramáticas, isto não implica em resultados melhores, tomando como base a avaliação F1.

A Figura 44 apresenta a visualização destes resultados.

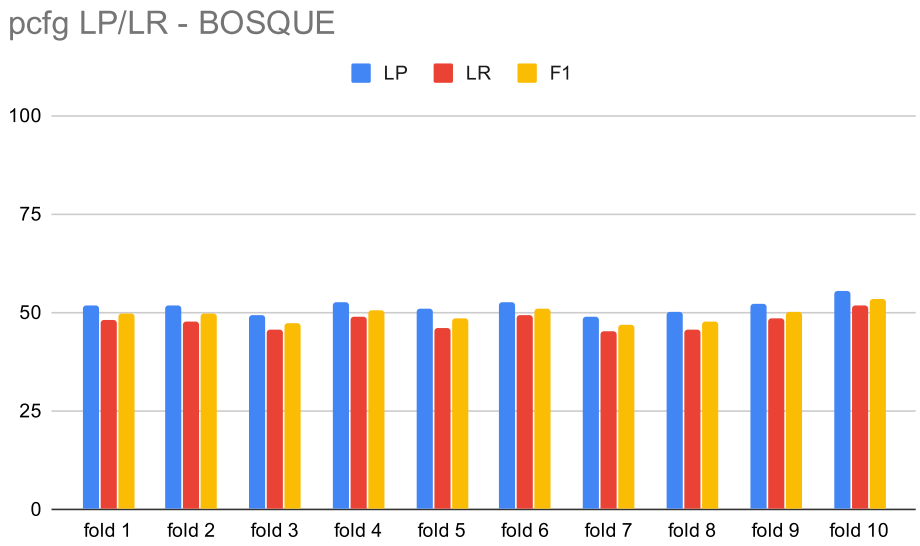


Figura 44 – Gráfico de resultados dos testes do PCFG do LexicalizedParser, usando o BOSQUE transduzido

4.2.3 Análise de Erro dos treinamentos do SP com dados transduzidos do BOSQUE

De modo análogo ao do CINTIL, serão apresentados casos pra exemplificar o resultado das transduções e da execução do CINTIL. Utilizaremos a gramática do décimo treino para as classificações e avaliações. O Bosque tem mais casos problemáticos que o CINTIL. Portanto, alguns deles serão exibidos apenas nos Apêndices.

Como visto anteriormente, a cada caso serão apresentadas, primeiro, as árvores como se apresentam no BOSQUE originalmente. Depois, as árvores transduzidas como resultado deste trabalho. Por fim, as árvores geradas pelo SP com as gramáticas geradas a partir da transdução do BOSQUE. No caso das árvores originais, foram removidas as informações adicionais dos nós, mantendo apenas $(F : f)$.

A Figura 45 mostra as diferenças entre árvores sem pontuação. Ocorreram poucos erros de *parsing*. Nenhum nó terminal foi classificado erroneamente, apenas nós internos

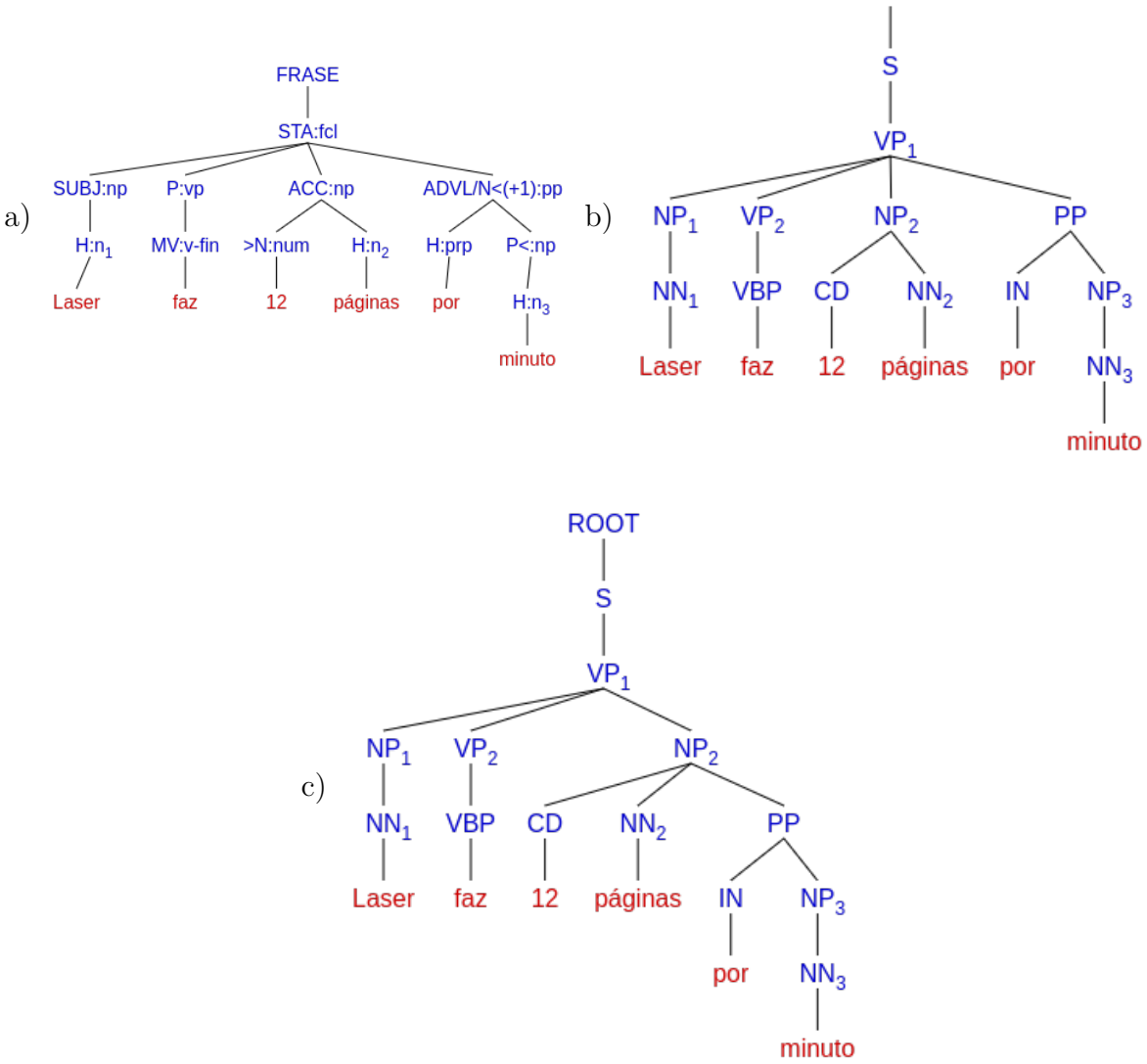


Figura 45 – Estudo da sentença CF761-1, "Laser faz 12 páginas por minuto", que originalmente não possui nenhuma pontuação. Em a), vemos a árvore relativa à sentença original no BOSQUE. Em b), a sentença pós transdução. E em c), a mesma sentença, classificada pelo SP com uma gramática gerada por este trabalho

(não-terminais). Ocorre uma ambiguidade no NP "12 páginas", que agora abarca também o PP irmão.

A Figura 46 traz uma sentença que, originalmente, possui o par $KOMP <: acl$ internamente. Pode-se notar como as regras unárias (Não-terminal → terminal) são mais consistentes que as binárias. Também, a separação de palavras originalmente conjuntas, como "por exemplo" e "do que". Ambas expressões aparecem pouco no CETEMFolha (22 e 34 vezes, respectivamente). Porém, pelas observações anteriores, supõe-se que o SP não

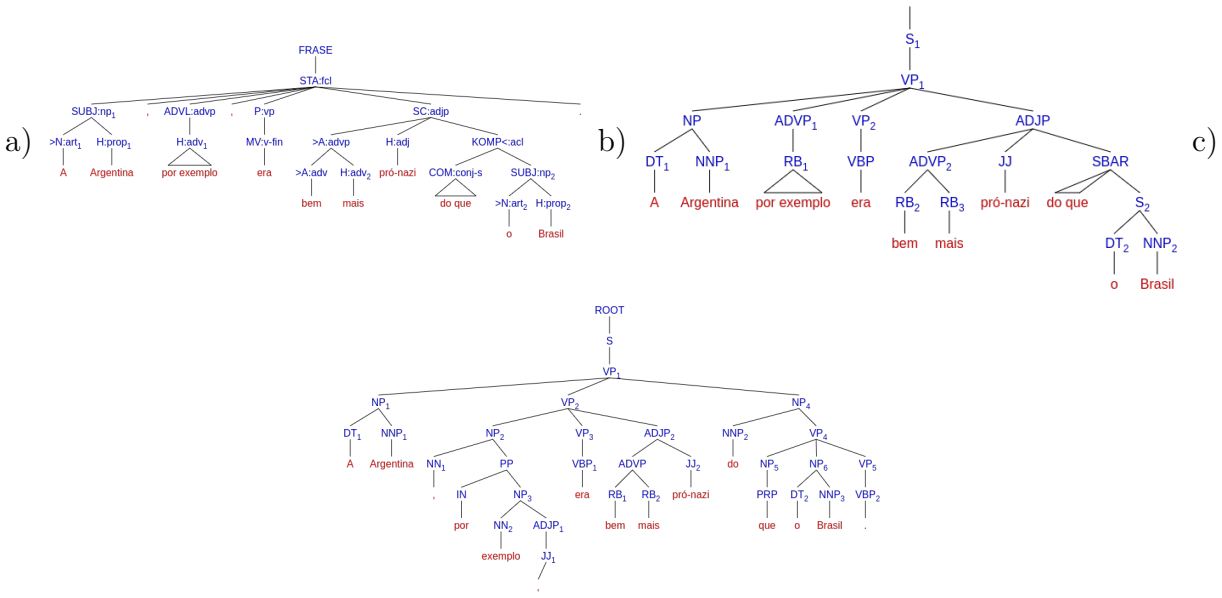


Figura 46 – Estudo da sentença CF766-10, “A Argentina, por exemplo, era bem mais pró-nazi do que o Brasil.”, que possui a estrutura KOMP<:acl internamente. Em a), vemos a árvore relativa à sentença original no BOSQUE. Em b), a sentença pós transdução. E em c), a mesma sentença, classificada pelo SP com uma gramática gerada por este trabalho

está pronto para lidar com nós terminais concatenados durante a execução. A estrutura de Comparação (estudada em 3.3.9) foi desfeita pelo SP.

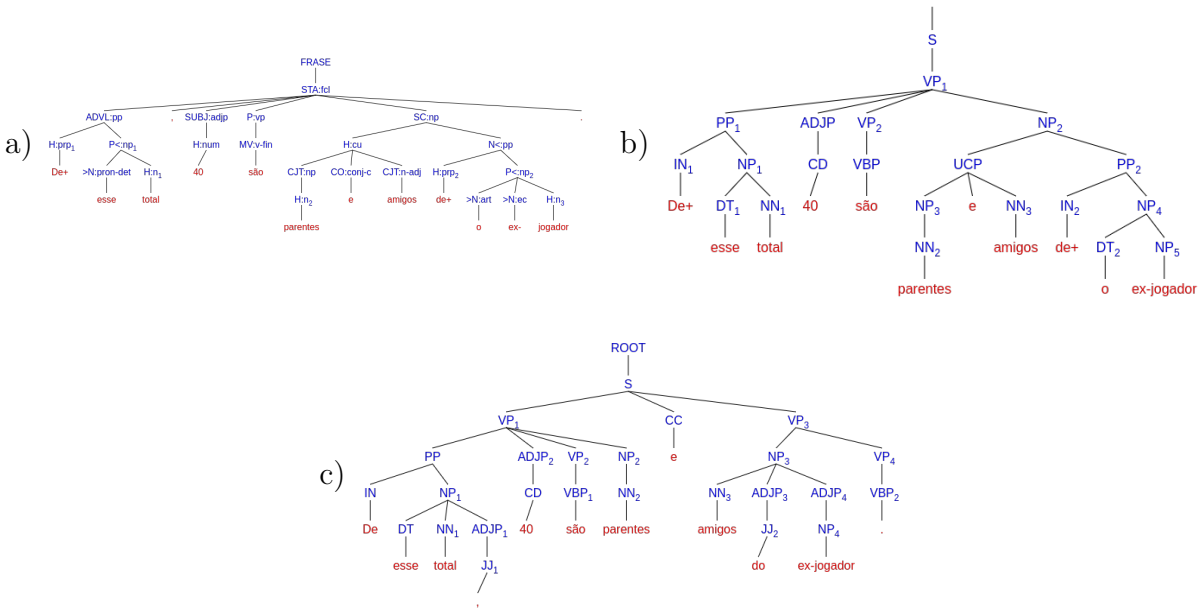


Figura 47 – Estudo da sentença CF866-2, “De esse total, 40 são parentes e amigos do ex-jogador.”, que possui a *tag ec*. Em a), vemos a árvore relativa à sentença original no BOSQUE. Em b), a sentença pós transdução. E em c), a mesma sentença, classificada pelo SP com uma gramática gerada por este trabalho

A Figura 47 foi gerada com o intuito de analisar a aplicação da *tag ec*, que não resultou em fatos inesperados. Mas a conjunção "parentes e amigos" foi desfeita pelo SP. "E" se tornou irmão dos dois sintagmas, tal qual uma vírgula, e recebeu a *tag CC*, que pouco aparece devido às conversões feitas na transdução para que o SP aceitasse conjunções. O mesmo ocorre no *parsing* da sentença CF318-C (fragmento em 49), mostrando que, apesar deste trabalho ter se guiado pelo *Bracketing Guidelines* (BIES et al., 1995), algumas estruturas poderiam se manter equivalentes às apresentadas no Bosque.

4.3 Discussão

Pela observação dos dados abordados, nota-se que a transdução de *datasets* pode não ser o melhor caminho para suprir a necessidade de *parsers* para o Português. Porém, existem algumas considerações que precisam ser feitas.

Primeiramente, deve-se notar que, apesar do treino do SP com dados transduzidos pelo CINTIL possuírem uma maior quantidade de dados, os resultados do treino deste (visto na Tabela 14) apresentam números menores, se comparados com os números da mesma operação sobre o Bosque (Tabela 16). Podemos levantar algumas hipóteses:

- A maior quantidade de dados permitiu que o SP tivesse mais acesso à árvores problemáticas, fazendo com que o resultado geral fosse inferior;
- Apesar do tamanho do Bosque ser menor, suas árvores são mais completas, permitindo maior generalização;
- A transdução dos dados do Bosque foi feita de forma mais acurada, permitindo um treinamento melhor.

Pode-se notar que o SP registrou mais *tags* sobre o Bosque do que sobre o CINTIL. Fazer uma comparação entre as Tabelas 6 e 8 poderia ser uma alternativa, mas não é suficiente: Foram usadas 20 *tags* na transdução do Bosque, e também 20 para o CINTIL. Mesmo considerando *tags* com mais de uma possibilidade do Bosque, para alguns casos (como comentado em 3.3.5 e 3.3.8), não é suficiente. Deve-se considerar, também, a quantidade de regras aprendidas em casa caso: no caso com mais regras do treinamento sobre CINTIL (*fold 7*), foram aprendidas 427 regras binárias, contra 4402 do caso do treino com o Bosque com menos regras (*fold 9*). Observar as Figuras 37 e 43 permite ver essas diferenças.

Cabe observar as *tags* utilizadas em cada *treebank*. Como pode-se ver na Tabela 18, basicamente foram usadas as mesmas *tags*. As diferenças expressivas estão, no CINTIL,

no uso repetitivo das interjeições, e no uso de CONJP. BOSQUE, por outro lado, tem uma pluralidade maior de marcações relativas à verbos. Retornar à Tabela 8 permite a constatação de que tal característica vem desde o *treebank* original.

CINTIL	BOSQUE
ADVP	ADJP
ADJP	ADVP
CC	CC
CD	CD
CONJP	DT
DT	IN
IN	JJ
INTJ	NN
JJ	NP
NN	NNP
NNS	PP
NP	PRP
PP	RB
PP$	S
PRP	UH
RB	VB
S	VBP
UH	VBG
VB	VP
VP	VBN

Tabela 18 – Tags utilizadas nas transduções do CINTIL e do Bosque

Com isto, pode-se deduzir o oposto da segunda hipótese: pelo fato das árvores do BOSQUE serem mais completas e mais robustas, deve-se criar uma quantidade de regras maior para este *dataset*. A quantidade de regras não torna as gramáticas geradas por este processo *melhores*. Pelo contrário, quando se põe em perspectiva que o CETEMFolha é muito menor, em quantidade de sentenças, que o CINTIL [10] , nota-se a desproporcionalidade dos números. Tal fato demonstra que o Stanford Parser tem dificuldade de realizar generalizações para as árvores advindas do BOSQUE.

Poderia-se supor que foi dada maior atenção para o desenvolvimento do transdutor relativo a um *treebank* em detrimento a outro. Isto não é verdade. Toda alteração feita, como já dito ao longo do trabalho, foi com o objetivo de fazer com que o *Stanford Parser* fosse capaz de receber as florestas sintáticas transduzidas. Toda alteração e desenvolvimento foram feitos pensando no mínimo necessário para que o processamento fosse possível. Com isso, quer-se dizer que: A ambos conjuntos de dados, foi dada atenção equivalente. O código

[10] O CINTIL é 2.4 vezes maior

foi escrito na medida que o SP e os *treebanks* indicavam problemas que demandavam soluções.

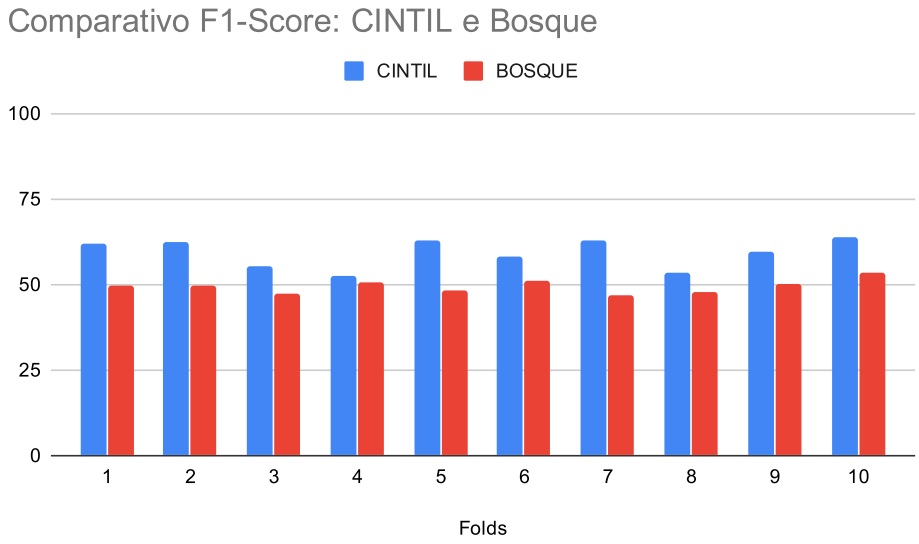


Figura 48 – Comparativo entre resultados de *F1-Score* para os 10 *folds* do CINTIL e do Bosque

Ambos *datasets* tiveram resultados baixos de *F1-score*. O CINTIL teve desempenho melhor, porém ainda assim pouco expressivo, entre 52% e 64%. O Bosque flutua entre 46% e 54%. A Figura 48 permite melhor visualização. As Seções 4.1.2 e 4.2.2 demonstram o problema. Deve-se levar em consideração os pontos abordados sobre as características da métrica de avaliação explanadas em 2.5.2, mas é inegável que os *parse* obtidos estão muito aquém do esperado.

Faz todo o sentido retomar este trabalho no futuro, aprimorar suas características, e torná-lo mais preciso, para que não seja necessário descartar nenhuma sentença que seja. A ausência de pontuações tem reflexos diretos no *score* final. Porém, notam-se ainda erros relevantes nas classificações. Erros esses não relacionados aos que receberam análises exclusivas nas Seções 3.2 e 3.3.

Pode-se considerar, também, o impacto do uso do modelo de linguagem do SP utilizado. O que é coerente, uma vez que houve uma escolha deliberada por usar o Stanford Parser sem quaisquer modificações - incluindo de configurações. Como não houve mudanças neste sentido, o modelo utilizado foi o padrão, referente à língua inglesa.

Mais: ao optar-se por não fazer grandes desenvolvimentos a partir dele, ou até mesmo não utilizar modelos de outras linguagens que estão disponíveis e distribuídos

oficialmente, era sabido que o modelo padrão que compõe o SP é o da língua inglesa. Portanto, em nada otimizado para o português, qualquer variante que seja.

Uma dos problemas motivadores deste trabalho, a ausência de *parsers* para língua portuguesa, ainda parece um pouco distante. Pelo menos no atual estágio dos transdutores desenvolvidos. Porém, transduzir os dois *datasets*, de modo que o SP fosse capaz de processá-las foi um grande avanço. Isto torna uma potencial nova etapa possível, essa sim podendo ser a resposta: adaptar o SP para o Português. Produzir um novo modelo de linguagem para a ferramenta parece ser uma alternativa plausível.

Esta ideia não é nova, o LX-Parser ((SILVA et al., 2010) e (SILVA; BRANCO; GONÇALVES, 2010)) foi desenvolvido justamente com este pensamento, mas se encontra obsoleto. Além de que, foi necessário usar um novo *dataset* no seu desenvolvimento. Criar transdutores que adaptem florestas sintáticas pré-existentes pode ser um avanço nesta parte do desenvolvimento. Isto, claro, se pensarmos num desenvolvimento já apoiado em ferramentas e bibliotecas pré-existentes. Desenvolver um *parser* novo sempre é uma possibilidade.

Com isto, a inquietação motivadora deste trabalho fica ainda sem solução: *Seria possível desenvolver um parser tentando utilizar apenas materiais já desenvolvidos (a saber, parsers já existentes, bem como corpus de treino já existentes), de modo que ele seja eficiente?* Este trabalho não é capaz de responder. Ficou claro que diversas adaptações feitas para a conclusão do mesmo não foram as ideais. Portanto, um estudo futuro seria capaz de nos aproximar da resposta. Evidente, também, que mesmo com transduções melhor projetadas e executadas, este procedimento apenas não é capaz de trazer bons resultados, uma vez que características outras do Stanford Parser, a saber, modelos de linguagem, ainda tem grande peso nos resultados finais. Sendo assim, é possível afirmar: O procedimento de transdução, apenas, não é suficiente para treinar o Stanford Parser de maneira eficiente.

A de se pesar, contudo, as contribuições que este trabalho trouxe. Primeiramente, foi feita uma catalogação inter-*corpora* robusta, com fundamentação linguística. Segundo, a própria metodologia de desenvolvimento de transdução inter-*corpora*, bem como os dois transdutores necessários para sua realização. Foram geradas gramáticas e análises que podem ser analisadas pela comunidade. E, por fim, este trabalho é um indicativo do caminho a seguir em pesquisas futuras sobre relações inter-*corpora*, e sobre *parsers* para a língua portuguesa.

5 Considerações Finais

Neste trabalho foram estudados *parsers*, suas limitações e possibilidades. Foi visto que existem *parsers* principalmente para a língua inglesa, e que há poucos *parsers* para a língua portuguesa. Mostrou-se que existem conjuntos de dados em Português que podem ser usados para treinar *parsers*. Foi desenvolvida uma metodologia de transdução de dados pré-existentes, de língua portuguesa, para a estrutura aceita pelo *parser* escolhido, sem perda de informação lexical. Essa metodologia prevê, também, a adaptação de estruturas internas de sentenças quando necessário. Foram desenvolvidos dois transdutores que realizaram tais adaptações, com pontuação média F-Score de 49.5% (SP operando com o BOSQUE transduzido), e 59.% (SP operando com o CINTIL transduzido).

Por fim, ficam algumas propostas de trabalhos futuros. A revisão deste trabalho, ampliando os casos tratados. Por exemplo, realizar transduções se preocupando não apenas com as*tags*, mas também com as estruturas internas das árvores; a correção de estruturas que, para este trabalho, não estão em padrão aceitável, como sinais de pontuação, e a porcentagem da transdução do Bosque (3.3.2); o desenvolvimento de transdutores como alternativa ao desenvolvimento de *parsers*; a implementação de *parser*, com desenvolvimento *out-of-the-box* para o português; dentre outras possibilidades.

Apêndices

APÊNDICE A – cintil

A.1 Tabelas

Tabela 19 – Tabela com resultados completos do CINTIL

	LP	**LR**	**F1**	**Exact**	**N**
1					
pcfg LP/LR summary evalb:	58.09	65.99	61.79	27.46	863
dep DA summary evalb:	69.47	69.47	69.47	34.26	858
factor LP/LR summary evalb:	59.69	68.88	63.96	28.38	863
factor Tag summary evalb:	81.94	82.78	82.36	54.92	863
2					
pcfg LP/LR summary evalb:	62.52	62.32	62.42	18.27	93
dep DA summary evalb:	70.69	70.69	70.69	24.73	93
factor LP/LR summary evalb:	67.82	69.88	68.83	20.43	93
factor Tag summary evalb:	85.89	86.35	86.12	38.7	93
3					
pcfg LP/LR summary evalb:	55.69	54.58	55.13	15.0	100
dep DA summary evalb:	67.4	67.4	67.4	24.0	100
factor LP/LR summary evalb:	64.95	65.75	65.35	25.0	100
factor Tag summary evalb:	83.63	83.96	83.79	36.0	100
4					
pcfg LP/LR summary evalb:	54.03	51.31	52.63	13.79	87
dep DA summary evalb:	70.44	70.44	70.44	29.88	87
factor LP/LR summary evalb:	61.23	59.49	60.35	12.64	87
factor Tag summary evalb:	85.37	85.37	85.37	35.63	87
5					
pcfg LP/LR summary evalb:	61.48	64.64	63.02	23.17	82
dep DA summary evalb:	63.81	63.81	63.81	24.39	82
factor LP/LR summary evalb:	64.61	69.12	66.79	30.48	82
factor Tag summary evalb:	84.94	84.94	84.94	40.24	82
6					
pcfg LP/LR summary evalb:	58.81	57.61	58.21	14.49	69
dep DA summary evalb:	69.88	69.88	69.88	27.53	69
factor LP/LR summary evalb:	68.56	68.06	68.31	20.28	69
factor Tag summary evalb:	85.97	86.35	86.16	36.23	69

Continua na próxima página

Tabela 19 – *Continuação da página anterior*

	LP	**LR**	**F1**	**Exact**	**N**
7					
pcfg LP/LR summary evalb:	62.51	63.67	63.09	25.33	75
dep DA summary evalb:	74.13	74.13	74.13	36.0	75
factor LP/LR summary evalb:	67.72	69.08	68.4	28.0	75
factor Tag summary evalb:	88.09	88.09	88.09	44.0	75
8					
pcfg LP/LR summary evalb:	53.03	54.05	53.54	16.41	67
dep DA summary evalb:	68.04	68.04	68.04	26.86	67
factor LP/LR summary evalb:	59.23	61.76	60.47	17.91	67
factor Tag summary evalb:	85.65	85.65	85.65	35.82	67
9					
pcfg LP/LR summary evalb:	61.76	57.5	59.56	21.11	90
dep DA summary evalb:	67.12	67.12	67.12	26.96	89
factor LP/LR summary evalb:	65.31	62.84	64.05	24.44	90
factor Tag summary evalb:	86.35	86.35	86.35	45.55	90
10					
pcfg LP/LR summary evalb:	63.32	63.98	63.65	20.89	67
dep DA summary evalb:	65.9	65.9	65.9	28.35	67
factor LP/LR summary evalb:	65.13	67.55	66.32	19.4	67
factor Tag summary evalb:	84.63	85.01	84.82	40.29	67

-cp	*ClassPath*. Indica o diretório onde se encontra a classe principal a ser executada
-mx4g	Quantidade de memória usada. No caso, 4 GB.
LexicalizedParser	*Parser* utilizado, dentre os disponibilizados
-loadFromSerializedFile	Carrega a gramática serializada, gerada na execução de treinamento anterior
arquivo.txt	Arquivo que contém sentenças a serem classificadas pelo SP

Tabela 20 – Comandos para uma execução simples do *Stanford Parser*, utilizando o terminal.

A.2 Imagens

A.3 Códigos

O *parsing* do PS é feito executando o comando A.1.

Listing A.1 – Execução de *parsing* em sentenças transduzidas a partir do CINTIL

```
java -cp stanford-parser.jar -mx4g edu.stanford.nlp.parser.
    lexparser.LexicalizedParser -loadFromSerializedFile ~/<
    diretorio de armazenamento>/serialGrammarBOSQUE6
    sentencas_teste_cintil.txt
```

Para realizar o treinamento do SP a partir do CINTIL, deve-se executar o comando A.3:

Listing A.2 – Execução de treinos do Stanford Parser para o CINTIL

```
java -cp ~/<diretorio de trabalho>/stanford-parser.jar -mx4g
     edu.stanford.nlp.parser.lexparser.LexicalizedParser -
    train ~/<diretorio do treebank>/tree-trad 1-1014 -
    saveToSerializedFile ~/<diretorio de armazenamento>/
    serialGrammarCINTIL -saveToTextFile ~/<diretorio de
    armazenamento>/textGrammarCINTIL
```

O código acima merece algumas explicações a parte, para quem não está familiarizado ao uso do SP pelo terminal.

A Tabela 24 mostra um fragmento das possibilidades de comandos a serem usados pela interface do terminal do SP.

Explicado rapidamente na Tabela 20.

Listing A.3 – Execução de testes do Stanford Parser para o CINTIL

```
java -cp stanford-parser.jar -mx4g edu.stanford.nlp.parser.
    lexparser.LexicalizedParser -loadFromSerializedFile /home
    /fernando/projeto-final-parsers/serialized-files/
    serialGrammarCINTIL1 input/sentencas_teste_cintil.txt
```

Que também merece explicações, na Tabela 21:

-cp	*ClassPath*. Indica o diretório onde se encontra a classe principal a ser executada
-mx4g	Quantidade de memória usada. No caso, 4 GB.
LexicalizedParser	*Parser* utilizado, dentre os disponibilizados
-writeOutputFiles	Indica que os testes imprimirão arquivos de saída, a serem definidos
-outputFilesDirectory	Define o diretório onde os arquivos de saída serão escritos.
-loadFromSerializedFile	Carrega a gramática serializada, gerada na execução de treinamento anterior
-testTreebank	Diretório onde se encontra o treebank a ser usado para teste. Os números no formato $a - b$ indicam o primeiro e o último arquivo, respectivamente. Números no formato $a - b, c - d$ indicam dois blocos de arquivos. Atente para não usar o mesmo bloco dos treinos, ou o parser passará por *overfitting*, e terá resultados enviesados.

Tabela 21 – Comandos para um teste simples do Stanford Parser, utilizando o terminal.

APÊNDICE B – BOSQUE

B.1 Tabelas

Tabela 22 – Tabela com resultados completos do BOSQUE

	LP	**LR**	**F1**	**Exact**	**N**
1					
pcfg LP/LR summary evalb	44.6	41.62	43.06	6.82	3650
dep DA summary evalb	68.39	68.39	68.39	14.04	3646
factor LP/LR summary evalb	47.31	45.97	46.63	8.73	3650
factor Tag summary evalb	64.63	66.08	65.35	9.28	3650
2					
pcfg LP/LR summary evalb	43.75	40.77	42.21	6.95	3652
dep DA summary evalb	67.26	67.26	67.26	12.92	3651
factor LP/LR summary evalb	46.41	45.27	45.83	8.13	3652
factor Tag summary evalb	64.24	65.68	64.96	9.44	3652
3					
pcfg LP/LR summary evalb	44.15	41.48	42.78	8.5	3657
dep DA summary evalb	67.66	67.66	67.66	13.26	3650
factor LP/LR summary evalb	46.92	45.92	46.42	8.28	3657
factor Tag summary evalb	64.18	65.59	64.88	8.12	3657
4					
pcfg LP/LR summary evalb	44.52	41.4	42.9	7.64	3661
dep DA summary evalb	67.7	67.7	67.7	13.34	3657
factor LP/LR summary evalb	46.21	44.64	45.41	8.44	3661
factor Tag summary evalb	63.95	65.36	64.65	9.09	3661
5					
pcfg LP/LR summary evalb	44.17	41.56	42.82	8.5	3667
dep DA summary evalb	67.17	67.17	67.17	13.62	3663
factor LP/LR summary evalb	46.16	45.11	45.63	8.75	3667
factor Tag summary evalb	63.76	65.16	64.45	8.8	3667
6					
pcfg LP/LR summary evalb	44.29	41.02	42.59	7.54	3656
dep DA summary evalb	67.65	67.65	67.65	13.54	3654
factor LP/LR summary evalb	47.03	45.46	46.23	8.2	3656
factor Tag summary evalb	64.05	65.5	64.77	8.78	3656

Continua na próxima página

Tabela 22 – *Continuação da página anterior*

	LP	**LR**	**F1**	**Exact**	**N**
7					
pcfg LP/LR summary evalb	44.92	42.14	43.49	8.23	3654
dep DA summary evalb	67.67	67.67	67.67	13.11	3652
factor LP/LR summary evalb	46.97	46.05	46.5	8.83	3654
factor Tag summary evalb	64.62	66.04	65.32	9.49	3654
8					
pcfg LP/LR summary evalb	44.83	41.62	43.17	8.68	3663
dep DA summary evalb	68.34	68.34	68.34	13.95	3661
factor LP/LR summary evalb	46.84	46.13	46.49	8.4	3663
factor Tag summary evalb	65.08	66.52	65.79	10.04	3663
9					
pcfg LP/LR summary evalb	43.88	40.83	42.3	7.49	3658
dep DA summary evalb	67.52	67.52	67.52	12.94	3654
factor LP/LR summary evalb	45.88	44.69	45.28	8.25	3658
factor Tag summary evalb	64.06	65.5	64.77	9.02	3658
10					
pcfg LP/LR summary evalb	43.78	40.54	42.1	8	3649
dep DA summary evalb	67.49	67.49	67.49	12.76	3644
factor LP/LR summary evalb	46.64	45.48	46.05	9.64	3649
factor Tag summary evalb	63.97	65.43	64.69	9.78	3649

Tabela 23 – Tabela de conversão completa: BOSQUE para PTB (Funções)

Tag Original (Português)	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
>A	dependente em adjp ou advp (antecede o núcleo)	>A	371	Explicado em 3.3.6
A<	dependente em adjp ou advp (segue o núcleo)	A<	272	Explicado em 3.3.6

Continua na próxima página

Tabela 23 – *Continuação da página anterior*

Tag Original (Português)	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
A<ARG	Estrutura não descrita em Freitas e Afonso (2007)	não convertida	12	
A<arg	Estrutura não descrita em Freitas e Afonso (2007)	não convertida	45	
ACC	objecto directo (incluindo alguns tipos de se)	depende da *form_tag*	4315	Explicado em 3.3.8
ACC-PASS	função do clítico se numa oração passiva (partícula apassivante)	NP	39	Refere-se ao uso de pronomes clíticos numa sentença
ADVL	adjunto adverbial	depende da form	6032	Explicado em 3.3.8
ADVL/A< [+1]	ambiguidade adjunto adverbial / adjunto adjetival	não convertida	2	
ADVL/ADVL[-3]	ambiguidade adjunto adverbial / adjunto adjetival	não convertida	2	
ADVL/N< [+1]	ambiguidade adjunto adverbial / adjunto adnominal	não convertida	70	
ADVL/N< [+2]	ambiguidade adjunto adverbial / adjunto adnominal	não convertida	25	
ADVL/N< [+3]	ambiguidade adjunto adverbial / adjunto adnominal	não convertida	5	

Continua na próxima página

Tabela 23 – *Continuação da página anterior*

Tag Original (Português)	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
ADVL/PIV	ambiguidade adjunto adverbial / obj. ind. preposicional	não convertida	1	
APP	aposição (do substantivo) [epíteto de identidade]	NP	212	
AUX	verbo auxiliar	VBP	1271	
AUX<	Em contexto de coordenção, partícula de ligação entre o auxiliar partilhado e verbos coordenados	VP	2	
CJT	elemento conjunto	_CJT_	3945	Explicado em 3.3.7
CJT&ACC	Coordenação de constituintes com funções diferentes	NP	1	Por observação de frequências
CJT&ADVL	Coordenação de constituintes com funções diferentes	PP	3	Por observação de frequências
CJT&PASS	Coordenação de constituintes com funções diferentes	PP	1	Por observação de frequências
CJT&PRED	Coordenação de constituintes com funções diferentes	ADJP	2	Por observação de frequências
CMD	enunciado imperativo	S	7	
CO	coordenador	não convertida	1753	

Continua na próxima página

Tabela 23 – *Continuação da página anterior*

Tag Original (Português)	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
COM	complementizador em estruturas de comparação (como, (do) que)	não convertida	118	
DAT	objecto indirecto pronominal (incluindo *se*)	NP	37	
EXC	enunciado exclamativo	S	36	
FOC	marcador de foco	ADJP	44	
H	núcleo	depende da form	40148	Explicado em 3.3.8
KOMP<	complemento comparativo	_KOMP_	40	Explicado em 3.3.9
MV	verbo principal	VP	7999	
>N	adjunto adnominal (antecede o núcleo)	NP	14009	Dobra do NP por adjunto, como visto em Mioto, Silva e Lopes (2013, p 67)
N<	adjunto adnominal (segue o núcleo)	NP	9208	Dobra do NP por adjunto, como visto em Mioto, Silva e Lopes (2013, p 67)
N</ADVL[-1]	ambiguidade adjunto adnominal / adjunto adverbial	não convertida	3	
N</ADVL[-2]	ambiguidade adjunto adnominal / adjunto adverbial	não convertida	1	
N</ADVL[-3]	ambiguidade adjunto adnominal / adjunto adverbial	não convertida	1	

Continua na próxima página

Tabela 23 – *Continuação da página anterior*

Tag Original (Português)	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
N</N<[+1]	ambiguidade adjunto adnominal / adjunto adnominal [1 nivel]	não convertida	2	
N</N<[+2]	ambiguidade adjunto adnominal / adjunto adnominal [2 níveis]	não convertida	20	
N</N<[-2]	ambiguidade adjunto adnominal / adjunto adnominal [2 níveis]	não convertida	1	
N</P<[+1]	ambiguidade adjunto adnominal / argumento de preposição	não convertida	1	
N<ARG	complemento nominal (complementa um substantivo não deverbal)	PP	139	
N<ARGO	complemento nominal (complementa um substantivo deverbal, relativo ao objecto)	PP	450	
N<ARGS	complemento nominal (complementa um substantivo deverbal, relativo ao sujeito)	PP	132	

Continua na próxima página

Tabela 23 – *Continuação da página anterior*

Tag Original (Português)	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
N<PRED	adjeto predicativo [epíteto predicativo]	NP	1542	
N<PRED / N<PRED[+2]	ambiguidade adjeto predicativo / adjeto predicativo	não convertida	2	
N<PRED / N<PRED[-2]	ambiguidade adjeto predicativo / adjeto predicativo	não convertida	1	
N<PRED / UTT[-4]	ambiguidade adjeto predicativo / enunciado	não convertida	1	
NUM<	dependente de numeral	não convertida	2	
OA	complemento adverbial (relativo ao objecto)	depende da form	27	
OC	predicativo do objeto	depende da form	102	Explicado em 3.3.8
>P	dependente da preposição	PP	71	Por observação, e por Mioto, Silva e Lopes (2013, p 67)
P	predicador	VP	8053	Pela Freitas (2006, p 60), O predicador é sempre de natureza verbal e, por isso, pode exibir apenas formas verbais
P<	argumento de preposição	PP	11574	Por observação, e por Mioto, Silva e Lopes (2013, p 67)
PASS	agente da passiva	PP	242	

Continua na próxima página

Tabela 23 – *Continuação da página anterior*

Tag Original (Português)	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
PAUX	em contexto de coordenação, verbo auxiliar partilhado por verbos principais com os seus próprios constituintes	não convertida	27	
PCJT	preposição conjunta (de/-desde.....a/até/para)	não convertida	20	
PIV	objecto preposicional	PP	1097	
PIV/N<[+1]	ambiguidade objecto preposicional / adjunto adnominal	não convertida	1	
PMV	em contexto de coordenação, verbo principal coordenado com os seus próprios constituintes	não convertida	52	

Continua na próxima página

Tabela 23 – *Continuação da página anterior*

Tag Original (Português)	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
PRD	Por Freitas (2006, p 123), existe normalmente uma palavra- *como*, *por*, etc. -que é uma conjunção subordinativa que inicia a oração de predicação (a função é representada por PRD)	não convertida	70	
PRED	adjunto predicativo	VP	76	
PRT-AUX	partícula de ligação verbal	não convertida	117	
QUE	enunciado interrogativo	S	64	
>S	dependente de complementizador	JJ	2	
S<	aposto da oração	NP	18	
SA	complemento adverbial [pode ser substituído por um pronome adverbial] (relativo ao sujeito)	PP	204	
SC	predicativo do sujeito	VP	1254	
STA	enunciado declarativo	S	3683	
SUB	subordinador	IN	746	

Continua na próxima página

Tabela 23 – *Continuação da página anterior*

Tag Original (Português)	**Nome da Tag**	**Tag Convertida**	**Ocorrências**	**Observações**
SUBJ	sujeito (incluindo sujeitos impessoais *se*)	depende da form	4982	depende da form
TOP	constituinte de tópico	NP	1	
UTT	enunciado	S	468	
VOC	constituinte vocativo	NP	8	
X	?	_X_	376	Explicado em 3.3.5

-cp	*ClassPath*. Indica o diretório onde se encontra a classe principal a ser executada
-mx4g	Quantidade de memória usada. No caso, 4 GB.
LexicalizedParser	*Parser* utilizado, dentre os disponibilizados
-train	Treino. Logo em seguida, um diretório e a lista de arquivos a serem usados para treinar
-saveToSerializedFile	Salva o resultado do treino num arquivo binário, cujo diretório está indicado na sequência
-saveToTextFile	Salva o resultado do treino num arquivo de texto, cujo diretório está indicado na sequência

Tabela 24 – Comandos para um treino simples do *Stanford Parser*, utilizando o terminal.

B.2 Imagens

```
...
(PP(IN que)
    (NP(DT o) (NNP Brasil)
        (NP (NNP .)))))))))
```

Figura 49 – Detalhe da sentença CF766-10, evidenciando a estrutura de comparação gerada.

Figura 50 – Erro no *FactoredParser*

B.3 Códigos

Listing B.1 – Conversão de arquivo ISO para UTF-8

```
$ iconv -f ISO-8859-1 Bosque_CF_8.0.PennTreebank.ptb -t UTF
    -8 -o Bosque_CF_PTB.txt
```

Listing B.2 – Execução de treinos do Stanford Parser para o Bosque

```
java -cp stanford-parser.jar -mx4g edu.stanford.nlp.parser.
    lexparser.LexicalizedParser -train ~/<diretorio do
    treebank> 1-421 -saveToSerializedFile ~/<diretorio de
    armazenamento>/serialGrammarBOSQUE1 > ~<diretorio de
    armazenamento>/outputs/treinoBOSQUE/treinoBr1.txt
```

São comandos análogos aos supracitados. Explicações podem ser vistas na Tabela 24.

Para a execução dos testes, foi utilizado o comando B.3

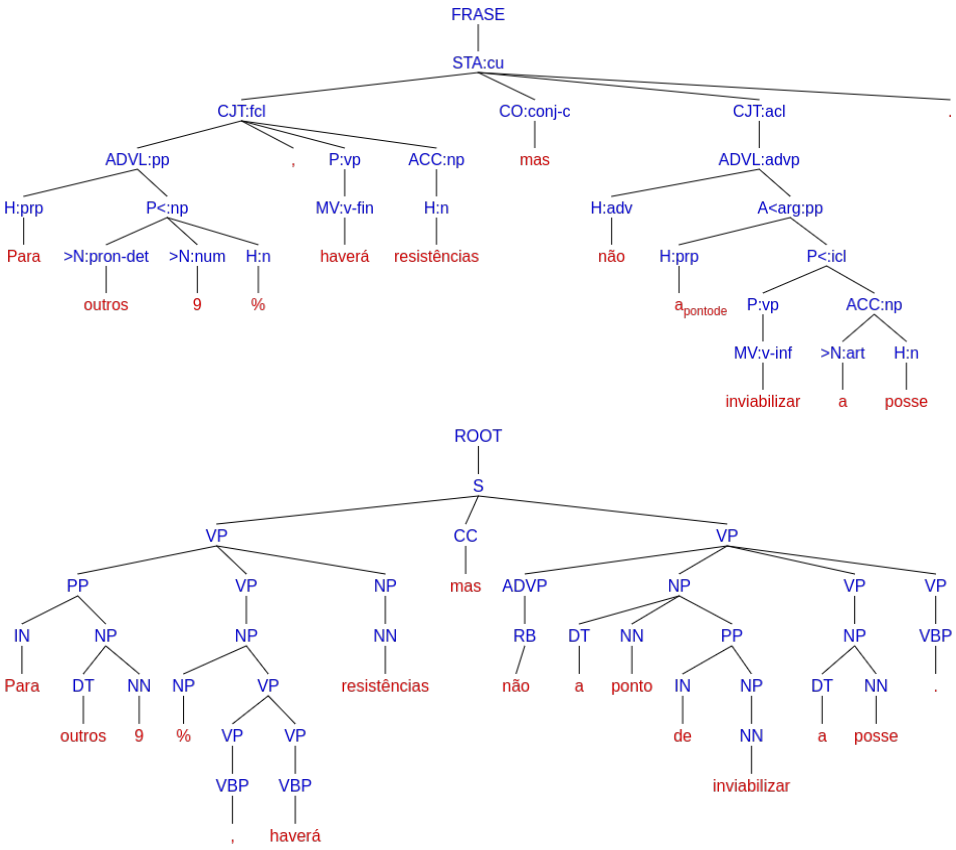


Figura 51 – Estudo da sentença CF144-5, “ Para outros 9%, haverá resistências mas não a ponto de inviabilizar a posse.”, que possui o símbolo de porcentagem. Note que, dessa vez, não há uma árvore gerada pelo SP pós treinamento.

Listing B.3 – Execução de testes do Stanford Parser para o Bosque

```
java -cp stanford-parser.jar -mx4g edu.stanford.nlp.parser.
    lexparser.LexicalizedParser -writeOutputFiles -
    outputFilesDirectory ~/<diretorio de relatorios de treino
    >/treino -loadFromSerializedFile ~/<diretorio da
    gramatica serializada>/serialGrammarBOSQUE1 -testTreebank
     ~/<diretorio dos treebanks> 422-4213 > ~/<diretorio dos
    resultados dos testes>testeBr1.txt
```

É o mesmo comando explicado na Tabela 21.

Referências

AHO, A.; SETHI, R.; LAM, S. *Compiladores: princípios, técnicas e ferramentas*. [S.l.]: LONGMAN DO BRASIL, 2008. ISBN 9788588639249. Citado na página 42.

BICK, E. The parsing system palavras. *Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*, University of Arhus, 2000. Citado 2 vezes nas páginas 25 e 49.

BIES, A. et al. *Bracketing Guidelines for Treebank II Style — Penn Treebank Project*. Philadelphia, PA, USA, 1995. v. 97, 100 p. Citado 15 vezes nas páginas 26, 63, 64, 67, 68, 70, 71, 77, 79, 81, 82, 87, 88, 89 e 105.

BRANCO, A. et al. *CINTIL TreeBank handbook: Design options for the representation of syntactic constituency*. Lisboa, 2011. Citado 10 vezes nas páginas 25, 26, 41, 60, 62, 63, 67, 71, 72 e 73.

BRANCO, A. H.; COSTA, F. A computational grammar for deep linguistic processing of portuguese: Lxgram, version a. 4.1. Department of Informatics, University of Lisbon, 2008. Citado na página 41.

CARVALHEIRO, C. *CINTIL Treebank Narrative Description*. 2012. Disponível em: <http://portulanclarin.net/repository/extradocs/CINTIL-Treebank.pdf>. Citado 2 vezes nas páginas 41 e 62.

CASTILHO, A. de. *Nova gramática do português brasileiro*. Brasil: Fapesp, 2010. ISBN 9788572444620. Citado 5 vezes nas páginas 29, 34, 60, 65 e 66.

CHARNIAK, E. Statistical techniques for natural language parsing. *AI Magazine*, v. 18, n. 4, p. 33–44, 1997. Citado 7 vezes nas páginas 21, 22, 42, 43, 44, 46 e 97.

CHEN, D.; MANNING, C. A fast and accurate dependency parser using neural networks. p. 740–750, 01 2014. Citado na página 25.

DERCZYNSKI, L. Complementarity, f-score, and nlp evaluation. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. [S.l.: s.n.], 2016. p. 261–266. Citado na página 51.

FANON, F. *Pele negra, máscaras brancas*. Salvador: Editora da Universidade Federal da Bahia, 2008. ISBN 9788523212148. Citado na página 21.

FREITAS, C.; AFONSO, S. Bíblia florestal: Um manual lingüístico da floresta sintá (c) tica. 2007. Citado 7 vezes nas páginas 26, 39, 40, 76, 83, 87 e 119.

FREITAS, C.; ROCHA, P.; BICK, E. Um mundo novo na floresta sintá (c) tica–o treebank do português. *Calidoscópio*, v. 6, n. 3, p. 142–148, 2008. Citado 2 vezes nas páginas 25 e 35.

FREITAS, S. A. C. Árvores deitadas: Descrição do formato e descrição das opções de análise na floresta sintáctica. *Texto produzido no âmbito da Floresta Sintá (c) tica*, 2006. Citado 13 vezes nas páginas 26, 39, 40, 60, 78, 79, 80, 85, 86, 87, 88, 123 e 125.

GARSIDE, R.; SAMPSON, G.; LEECH, G. *The computational analysis of English: A corpus-based approach*. University of Michigan: Longman, 1988. v. 57. Citado na página 36.

GOLD, E. M. Language identification in the limit. *Information and control*, Elsevier, v. 10, n. 5, p. 447–474, 1967. Citado na página 45.

HOPCROFT, J.; MOTWANI, R.; ULLMAN, J. *Introduction to Automata Theory, Languages, and Computation*. [S.l.]: Pearson Education International, 2003. (Addison-Wesley series in computer science). ISBN 9780321210296. Citado na página 32.

JAMES, G. et al. *An Introduction to Statistical Learning: with Applications in R*. [S.l.]: Springer New York, 2013. (Springer Texts in Statistics). ISBN 9781461471387. Citado na página 91.

LINGUATECA. *Projecto Floresta Sintá(c)tica*. 2010. Disponível em: <https://www.linguateca.pt/Floresta/>. Citado 2 vezes nas páginas 39 e 49.

LOVECRAFT, H. P. *Nyarlathotep*. The United Amateur, 1920. Disponível em: <https://www.sitelovecraft.com/multimedia.php\#Textos>. Citado na página 87.

MANNING, C. *15 5 Constituency Parser Evaluation*. 2018. Disponível em: <https://www.youtube.com/watch?v=_JtP-32keKE>. Citado na página 52.

MANNING, C. et al. The stanford corenlp natural language processing toolkit. In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. [S.l.: s.n.], 2014. p. 55–60. Citado na página 47.

MANNING, C. D.; SCHÜTZE, H. *Foundations of statistical natural language processing*. [S.l.]: MIT press, 1999. Citado 15 vezes nas páginas 25, 30, 31, 32, 33, 34, 35, 43, 44, 45, 47, 51, 52, 53 e 95.

MARCUS, M. P.; MARCINKIEWICZ, M. A.; SANTORINI, B. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, MIT Press, Cambridge, MA, USA, v. 19, n. 2, p. 313–330, jun. 1993. ISSN 0891-2017. Disponível em: <http://dl.acm.org/citation.cfm?id=972470.972475>. Citado 7 vezes nas páginas 25, 26, 35, 36, 37, 60 e 70.

MIOTO, C.; SILVA, M.; LOPES, R. *Novo manual de sintaxe*. [S.l.]: Editora Contexto, 2013. ISBN 9788572448000. Citado 8 vezes nas páginas 60, 63, 68, 76, 84, 85, 121 e 123.

MOHRI, M. Weighted finite-state transducer algorithms. an overview. In: _____. *Formal Languages and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 551–563. ISBN 978-3-540-39886-8. Disponível em: <https://doi.org/10.1007/978-3-540-39886-8_29>. Citado 2 vezes nas páginas 23 e 57.

MOOR, J. The dartmouth college artificial intelligence conference: The next fifty years. *AI Magazine*, v. 27, n. 4, p. 87, Dec. 2006. Disponível em: <https://www.aaai.org/ojs/index.php/aimagazine/article/view/1911>. Citado na página 22.

NIVRE, J. et al. The CoNLL 2007 shared task on dependency parsing. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, 2007. p. 915–932. Disponível em: <https://www.aclweb.org/anthology/D07-1096>. Citado na página 39.

NLX-GRUPO DE FALA E LINGUAGEM NATURAL. *LX-Parser*. 2010. Disponível em: <http://lxcenter.di.fc.ul.pt/tools/pt/conteudo/LXParser.html>. Acesso em: 20 nov. 2019. Citado 3 vezes nas páginas 25, 26 e 50.

OLIVEIRA, E. *Serviço social para corajosos: Entre falácias, mitos e realidade carne e osso*. [S.l.]: Viseu, 2019. ISBN 9788530010256. Citado na página 21.

PAGANI, L. A. Avaliação epistemológica de um exemplo de análise de ambigüidade num manual de introdução à semântica. 2009. Citado 2 vezes nas páginas 42 e 43.

RODRIGUES, J. *O que é o Processamento de Linguagem Natural?* 2017. Disponível em: <https://link.medium.com/pxMoml9N41>. Citado na página 21.

ROMANYSHYN, V. D. M. *The Dirty Little Secret of Constituency Parser Evaluation*. 2014. Disponível em: <https://tech.grammarly.com/blog/the-dirty-little-secret-of-constituency-parser-evaluation>. Citado na página 52.

ROSÉN, V. et al. An open infrastructure for advanced treebanking. In: HAJIČ, JAN. *META-RESEARCH Workshop on Advanced Treebanking at LREC2012*. Istanbul, Turkey, 2012. p. 22–29. Citado na página 41.

RUDER, S. *NLP-progress: Constituency parsing*. 2019. Disponível em: <www.nlpprogress.com/english/constituency_parsing.html>. Citado na página 30.

SANTORINI, B. Part-of-speech tagging guidelines for the penn treebank project (3rd revision). *Technical Reports (CIS)*, p. 570, 1990. Citado 3 vezes nas páginas 60, 64 e 76.

SANTORINI, B. Part-of-speech tagging guidelines for the penn treebank project (3rd revision, 2nd printing). *Ms., Department of Linguistics, UPenn. Philadelphia, PA*, 1990. Citado na página 37.

SASAKI, Y. et al. The truth of the f-measure. *Teach Tutor mater*, v. 1, n. 5, p. 1–5, 2007. Citado na página 51.

SILVA, J. et al. Out-of-the-box robust parsing of portuguese. In: PARDO, T. A. S. et al. (Ed.). *Computational Processing of the Portuguese Language*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 75–85. ISBN 978-3-642-12320-7. Citado 3 vezes nas páginas 23, 50 e 108.

SILVA, J.; BRANCO, A.; GONÇALVES, P. Top-performing robust constituency parsing of Portuguese: Freely available in as many ways as you can get it. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA), 2010. Disponível em: <http://www.lrec-conf.org/proceedings/lrec2010/pdf/136_Paper.pdf>. Citado na página 108.

TURING, A. M. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX, n. 236, p. 433–460, 10 1950. ISSN 0026-4423. Disponível em: <https://doi.org/10.1093/mind/LIX.236.433>. Citado na página 21.

WING, B.; BALDRIDGE, J. Adaptation of data and models for probabilistic parsing of portuguese. In: *Proceedings of the 7th International Conference on Computational Processing of the Portuguese Language*. Berlin, Heidelberg: Springer-Verlag, 2006. (PROPOR'06), p. 140–149. ISBN 3-540-34045-9, 978-3-540-34045-4. Disponível em: <http://dx.doi.org/10.1007/11751984_15>. Citado na página 80.