JOSUÉ GARCIA DE ARAÚJO

ALINHAMENTO DE ÁRVORES SINTÁTICAS PORTUGUÊS-INGLÊS

SÃO CARLOS
2011

JOSUÉ GARCIA DE ARAÚJO

ALINHAMENTO DE ÁRVORES SINTÁTICAS PORTUGUÊS-INGLÊS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação

Orientadora: Prof. Dra. Helena de Medeiros Caseli

SÃO CARLOS
2011

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

A663aa

Araújo, Josué Garcia de.
Alinhamento de árvores sintáticas português-inglês / Josué Garcia de Araújo. -- São Carlos : UFSCar, 2011.
77 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2011.

1. Processamento da linguagem natural (Computação). 2. Linguística - processamento de dados. 3. Inteligência artificial. I. Título.

CDD: 006.35 (20ª)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

"Alinhamento de árvores sintáticas português-inglês"

JOSUÉ GARCIA DE ARAÚJO

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação

Membros da Banca:

Helena de M. Caseli

Profa. Dra. Helena de Medeiros Caseli (Orientadora - DC/UFSCar)

Profa. Dra. Lúcia Helena Machado Rino (DC/UFSCar)

Leandro H. M. de Oliveira

Prof. Dr. Leandro Henrique Mendonça de Oliveira (EMBRAPA/Campinas)

São Carlos
Junho/2011

Agradecimentos

À minha familia por me apoiarem nas decisões tomadas ao longo de minha vida e por terem suportado a minha ausência durante estes dois anos.

À minha noiva Edilaine, o grande amor da minha vida, por ter demonstrado capacidade de superar momentos difíceis ao meu lado, incentivando a nunca desistir dos meus sonhos, nossos sonhos.

Aos meus colegas de Laboratório, em especial a Elen pela amizade e atenção, por saber que ao final desta jornada diremos até breve e não Adeus.

À Helena, minha cara orientadora, pelo profissionalismo, paciência e amizade. Por acreditar no meu trabalho e me dar a honra de ser seu primeiro aluno de Mestrado.

Aos colegas do NILC que me ajudaram profissionalmente, em especial ao Thiago pelas contribuições e esclarecimentos dado ao longo deste trabalho.

À CAPES pelo apoio financeiro, ao Departamento de Computação da UFSCar e seus funcionários que estiveram presentes no meu dia a dia.

Por fim, agradeço a Deus por esta experiência e saiba que precisando pode contar comigo...

muito obrigado!

Resumo

A tradução manual de uma língua natural fonte para uma língua natural alvo é uma tarefa que demanda tempo e conhecimento. Para reduzir o trabalho árduo necessário na construção manual de traduções, propôs-se realizar esta tarefa por meio de sistemas computacionais de Tradução Automática (TA). Desde a década de 1940, várias técnicas e abordagens de TA têm sido propostas, investigadas e avaliadas com o intuito de melhorar a qualidade das traduções geradas automaticamente. No momento, os métodos de tradução automática estatística são considerados o estado-da-arte em termos de medidas automáticas de avaliação comumente utilizadas na área (como BLEU e NIST), porém há uma tendência recente de que tais sistemas não conseguirão sair do patamar de desempenho no qual se encontram estagnados sem a aplicação de conhecimento linguístico mais aprofundado, por exemplo, informação sintática. Nesse sentido, como uma tentativa de auxiliar o processo de construção de tradutores automáticos, este documento apresenta a investigação, implementação e avaliação de técnicas de alinhamento de árvores sintáticas. A ferramenta computacional para alinhamento automático de árvores sintáticas resultante deste trabalho pode ser utilizada para a geração de um recurso extremamente útil para diversas técnicas de TA: as árvores sintáticas alinhadas. Esse recurso, até então inexistente para o português do Brasil, possibilitará o desenvolvimento de pesquisas inovadoras e que propiciem o avanço científico da área. Neste documento, um estudo de várias técnicas de alinhamento de árvores sintáticas é apresentado, baseado na literatura. O pré-processamento de um *corpus* para inserção de informações sintáticas a partir das quais o alinhamento é realizado também é descrito, destacando-se as fases de alinhamento lexical e análise sintática. A partir do embasamento teórico derivado do estudo das técnicas propostas na literatura, cinco modelos foram implementados para realizar a tarefa de alinhar as árvores sintáticas. Estes modelos foram avaliados usando o *corpus* pré-processado. Com base nos resultados da avaliação intrínseca do alinhamento propriamente dito, é possível concluir que o alinhamento de árvores sintáticas atingiu cerca de 97,36% de precisão e 93,48% de cobertura em pares de árvores representando sentenças paralelas em português do Brasil e inglês usando diferentes configurações. A partir desses resultados promissores pretende-se aplicar a ferramenta a um *corpus* maior de árvores sintáticas paralelas visando a obtenção de mais exemplos de tradução e permitindo, assim, sua aplicação nas técnicas de tradução automática baseada em sintaxe como os métodos estatísticos baseados em sintaxe ou a tradução orientada a dados.

Abstract

The manual translation of a source natural language into a target natural language is a task that demands time and expertise. In order to reduce the work needed for manual translations, the aim is to accomplish this task through Machine Translation (MT) systems. Since the 1940s, various approaches and techniques of MT have been proposed, investigated and evaluated in order to improve the quality of translations generated automatically. Nowadays, statistical machine translation methods are considered the state-of-art regarding the evaluation automatic measures commonly used in the area (such as BLEU and NIST), however a recent trend indicates that such systems will not improve their level of performance without the application of deeper linguistic knowledge, for instance, syntactic information. Thus, as an attempt to support the building of automatic translators, this dissertation presents the research, the implementation and the evaluation of parse trees alignment techniques. The computational tool for the automatic alignment of syntactic trees, result of this work, may be used to generate an extremely useful resource for various MT techniques: the aligned syntactic trees . This resource, so far unavailable for Brazilian Portuguese, will allow the development of new researches, which can provide the scientific advancement of the area. In this dissertation, a study of various techniques for parse trees alignment from the literature is presented. Besides, the pre-processing of a corpus for the inclusion of syntactic information from which the alignment is performed is also described, as well as the phases of lexical alignment and syntactic analysis. Some implementations and tests have been carried out with the pre-processed corpus, based on the theoretical foundations derived from the study of the techniques proposed in the literature. Based on the results of the intrinsic evaluation of the alignment, it was possible to conclude that the alignment of syntactic trees reached the accuracy of 97.36% and the coverage of 93.48% for tree pairs, representing parallel sentences in Brazilian Portuguese and in English by using different settings. Since the results have been promising, as future work, the aim is to apply the tool to a larger corpus of parallel syntactic trees, in order to obtain more examples of translation and, thus, allow its application to syntax-based machine translation techniques, such as syntax-based statistical methods or data-oriented translation.

Lista de Figuras

Lista de Tabelas

Sumário

1 Introdução

A área de Tradução Automática (TA) é uma das mais antigas em Processamento de Língua Natural. Surgiu na década de 40 e tem sido estudada ao longo dos anos. O interesse por sistemas de Tradução Automática tem aumentado, no contexto de um mundo globalizado, onde se faz necessária a tradução de forma rápida, precisa e de baixo custo. Com o advento da *web*, como um dos grandes meios de comunicação, a quantidade de informações em várias línguas fez crescer a busca por ferramentas capazes de traduzir uma língua fonte em uma língua alvo.

Segundo Caseli (2007, p. 1), a Tradução Automática pode ser entendida como a "tradução de uma língua natural (fonte) para outra (alvo) por meio de programas de computador". Basicamente, os sistemas de tradução automática se dividem em duas categorias: a tradução direta e a tradução indireta. A tradução direta se dá por meio de um dicionário bilíngue e a reordenação das palavras de acordo com as regras da linguagem alvo. Na tradução indireta, as sentenças da língua fonte são representadas em uma linguagem intermediária e posteriormente transferidas para a língua alvo.

A tradução indireta pode ser feita pelo método de transferência ou de interlínguas. O método de transferência consiste em três etapas: a análise, a transferência e a geração. O método de interlínguas extrai a representação do significado da sentença fonte e gera a sentença na linguagem alvo por meio desta representação (SPECIA; RINO, 2002).

Na etapa de análise realizada no método de transferência é comum a análise sintática e, em alguns casos, até a análise semântica. Esta análise sintática é estruturada em forma de árvore em uma representação intermediária. Na etapa seguinte (transferência), a estrutura de árvore obtida na análise sintática da língua fonte é mapeada gerando a estrutura de árvore da língua alvo. Na última etapa (geração), a estrutura de árvore da língua alvo é convertida na sentença final.

As árvores sintáticas alinhadas são um recurso que pode ser utilizado para realizar o mapeamento entre a estrutura de árvore obtida na fase de análise e a estrutura requerida na fase de geração.

A tradução automática com base nas árvores de análise sintática (ou apenas árvores sintáticas) tem sido bastante pesquisada, atualmente, devido à necessidade de melhorar o desempenho dos tradutores considerados o estado-da-arte: os modelos de tradução estatística baseada em frases (*phrase-based statistical machine translation* ou PB-SMT). Em muitas dessas técnicas baseadas em sintaxe – como (POUTSMA, 2003), (GILDEA, 2003), (LAVIE et al., 2008),(HEARNE; WAY, 2003) –, as árvores sintáticas nas línguas fonte e alvo podem ser alinhadas para que, a partir das mesmas, o "conhecimento de tradução" possa ser derivado. Nesse sentido, diversas técnicas para alinhar árvores de análise sintática estão sendo estudadas, mas não se tem conhecimento de um estudo focando a língua português do Brasil. Neste contexto, este trabalho apresenta uma investigação da aplicação de alguns métodos de alinhamento de árvores sintáticas especificamente para o idioma português do Brasil e sua tradução para o inglês.

1.1 Motivação

Este documento investiga a criação de um recurso muito útil no paradigma não linguístico de TA, ou tradução automática baseada em grandes *corpora* de textos bilíngues para treinamento e/ou base de exemplos (DORR et al., 1999). As técnicas deste paradigma não estão baseadas nas teorias linguísticas nem tão pouco nas propriedades linguísticas das línguas fonte e alvo. Mais precisamente, essas técnicas tentam encontrar características no *corpus* paralelo alinhado que possam auxiliar na tarefa de tradução automática . Um *corpus* paralelo são dois conjuntos de sentenças de línguas distintas entre si, no qual um conjunto é a tradução equivalente ao outro conjunto. Esse conjunto de sentenças paralelas pode estar alinhado lexicalmente, onde cada par de sentenças possui indicações de quais *tokens* (segmento de texto, palavras, símbolos de pontuação etc.) da sentença fonte são traduções de quais *tokens* da sentença alvo (CASELI, 2007).

Nesse projeto, as sentenças são representadas por suas árvores sintáticas e a proposta é encontrar o melhor alinhamento entre os nós dessas árvores paralelas. Assim, o projeto aqui descrito visa investigar um tipo de informação que pode ser aplicada nas pesquisas em TA, a informação sintática, servindo de base para muitas outras pesquisas futuras como, por exemplo, a tradução orientada a dados (do inglês, *Data-Oriented Translation* ou DOT). O modelo DOT, originalmente proposto por Poutsma (1998, 2003), pode ser descrito como "um modelo híbrido de tradução que combina exemplos, informação linguística e estatística" (HEARNE; WAY, 2006). Em DOT, um modelo de tradução é aprendido a partir de árvores sintáticas fonte e alvo alinhadas. Além dessa, outra técnica de TA que poderá se beneficiar dos resultados desse projeto é a tradução por meio de regras induzidas automaticamente a partir de informação sintática. A indução de regras

de tradução foi o tema do projeto de doutorado da orientadora, o ReTraTos[1], no qual sistemas de indução automática de dicionários bilíngues e de regras de tradução foram implementados com base em informação superficial (lemas, PoS, alinhamentos lexicais etc.) presente nos textos paralelos (CASELI, 2007). A partir dos alinhamentos de árvores sintáticas, uma nova versão do indutor de regras de tradução implementado no ReTraTos poderá ser implementada para induzir regras mais complexas, usando informação sintática.

1.2 Objetivos

O trabalho aqui apresentado tem como objetivo: identificar um modelo de alinhador de árvores sintáticas paralelas capaz de alinhar os conjuntos de árvores sintáticas obtidas a partir de textos paralelos em português do Brasil e inglês, por meio da implementação e avaliação de vários modelos. O recurso derivado da aplicação do alinhador (as árvores sintáticas alinhadas) pode auxiliar nos estudos sobre o uso de informação sintática na tradução automática.

1.3 Organização do texto

O restante deste documento está organizado como se segue. O capítulo 2 apresenta relatos dos métodos de alinhamento de árvores sintáticas propostos na literatura (seção 2.1) e as métricas de avaliação mais comuns utilizadas nos trabalhos relacionados (seção 2.2).

O capítulo 3 descreve o principal recurso linguístico usado na investigação dos métodos de alinhamento de árvores sintáticas: o *corpus* paralelo, assim como relata as etapas do pré-processamento do *corpus* paralelo usado neste experimento.

Além de descrever o desenvolvimento da ferramenta de alinhamento e avaliação das árvores sintáticas, o capítulo 4 apresenta os métodos aplicados na tarefa de alinhar os nós não terminais das árvores sintáticas, detalhados na seção 4.1.

Os resultados obtidos e os experimentos usados para validar este projeto estão presentes no capítulo 5. A seção 5.2 deste capítulo apresenta um exemplo de aplicação das regras extraídas das árvores sintáticas alinhadas.

Por fim, o capítulo 6 apresenta as conclusões deste trabalho.

[1] `http://www.nilc.icmc.usp.br/nilc/projects/retratos.htm`

2 Revisão Bibliográfica

Com o intuito de contextualizar o leitor nos principais conceitos, trabalhos e metodologia envolvidos com o alinhamento de árvores sintáticas, esse capítulo apresenta um relato dos métodos propostos na literatura (seção 2.1), a metodologia de avaliação empregada e os resultados obtidos (seção 2.2).

2.1 Métodos de Alinhamento de Árvores Sintáticas

O alinhamento de árvores sintáticas é o processo de encontrar as correspondências entre nós não terminais e nós terminais de duas árvores paralelas, ou seja, árvores sintáticas representando sentenças que são traduções umas das outras.

Para ilustrar esse processo, considere, por exemplo, o par de árvores sintáticas apresentado na figura 2.1. O alinhamento dos nós não terminais e nós terminais pode ser obtido seguindo diversas abordagens resultando, por exemplo, nos alinhamentos apresentados na figura 2.2 como linhas pontilhadas. Esse alinhamento, proposto por Menezes e Richardson (2001), está baseado em correspondências lexicais presentes no léxico bilíngue (identificados com a letra L) e regras de uma gramática de alinhamento (aplicadas em ordem e recursivamente até que nenhum novo alinhamento seja gerado). A regra R1, na figura 2.2, especifica o alinhamento entre traduções bidirecionais únicas como é o caso de *dirección* e *address*, *usted* e *you* e *clic* e *click*. A regra R3 alinha os filhos de pais alinhados que possuem correspondência lexical como é o caso de *hipervínculo* e *hyperlink*. Com a resolução da ambiguidade que a palavra *hipervínculo* (possível tradução de *Hyperlink_Information* e *hyperlink*), a regra R1 é novamente aplicada para determinar o alinhamento entre *información* e *hipervínculo* com *Hyperlink_Information*. Por fim, a regra R4 é aplicada para criar o alinhamento entre *hacer* e *click* já que ela especifica, grosso modo, que um nó verbo (*hacer*) cujo filho não verbo (*clic*) está alinhado com nó verbo (*click*) deve se juntar ao filho no alinhamento com nó verbo na sentença alvo.

Diversos outros métodos foram propostos na literatura com o mesmo intuito do método de

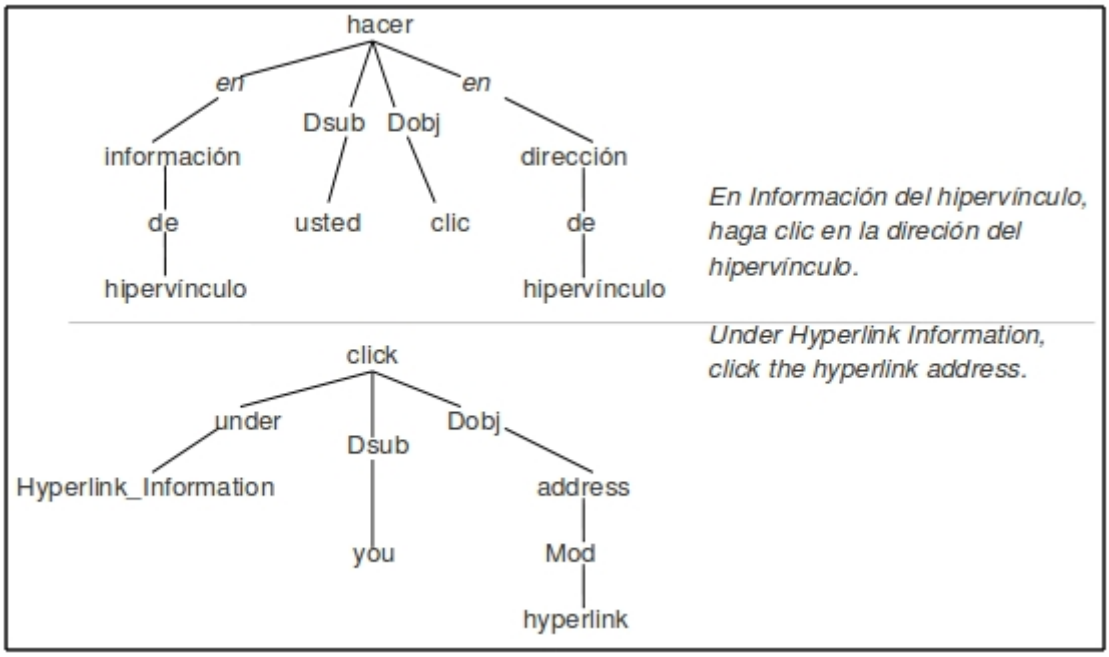


Figura 2.1: Formas Lógicas para um par de sentenças espanhol–inglês (MENEZES; RICHARDSON, 2001)

(MENEZES; RICHARDSON, 2001), como o método de Tinsley et al. (2007). Esse método, diferente do método de (MENEZES; RICHARDSON, 2001) que visa a indução de regras de tradução, tem como ponto forte o fato de estar desvinculado de qualquer aplicação, podendo ser usado como um passo prévio para diversas e não apenas uma aplicação específica. Tal método também apresenta outras vantagens como: preservar a estrutura da árvore, usar o mínimo de recursos externos e não fixar o alinhamento lexical a princípio, o que caracteriza uma independência da lingua fonte e alvo escolhida.

No modelo proposto por Tinsley et al. (2007), alguns critérios devem ser seguidos para alinhar os nós das árvores paralelas:

- Um nó só pode ser ligado uma única vez;
- Os nós descendentes de uma língua fonte só podem ser ligados aos nós descendentes de suas contrapartes na língua alvo;

Uma ligação entre dois nós equivalentes nas árvores indica que:

- As substrings representadas por esses nós são traduções equivalentes;
- Todo o sentido transportado pelo restante da frase fonte é encapsulado no restante da frase alvo, e vice-versa.

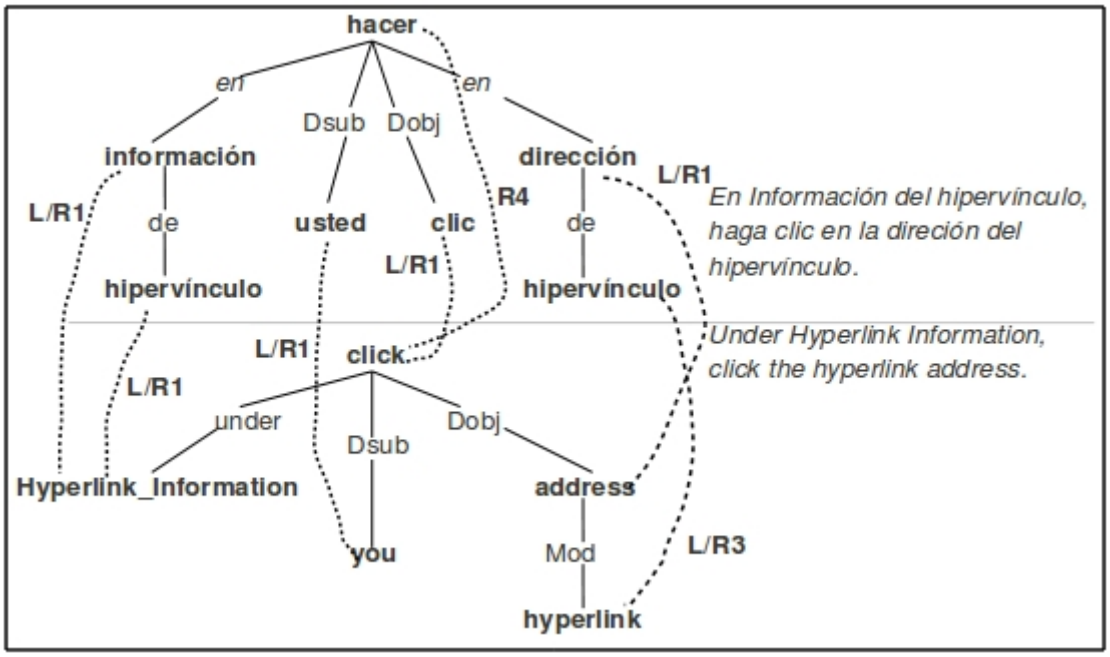


Figura 2.2: Alinhamentos das formas lógicas fonte e alvo da figura 2.1

O algoritmo analisa cada possível par de nós não terminais entre a árvore fonte e a árvore alvo, gerando uma pontuação de acordo com a probabilidade de alinhamento dos nós lexicais. Esta probabilidade é dada por um sistema de TA estatística que usa o alinhador de palavras GIZA++[1] (OCH; NEY, 2003). Os pares de nós não terminais com maior pontuação são alinhados, mantendo a estrutura da árvore de acordo com os nós descendentes e ascendentes. A ligação entre os nós respectivos ocorre seguindo um processo iterativo. Uma nova pontuação é dada a cada iteração, considerando apenas os pares de nós não alinhados.

Utilizando estes dados, a ligação de cada par de árvores $\langle S,T \rangle$ é calculada, sendo S a árvore originada pela língua fonte e o T originada pela língua alvo. O processo de alinhamento é inicializado propondo todas as ligações $\langle s,t \rangle$ entre nós em S e T como hipóteses e atribuindo pontuação $\gamma(\langle s,t \rangle)$ para eles. Todas as hipóteses pontuadas como zero são bloqueadas antes do algoritmo efetuar a ligação entre os nós relacionados. O processo de seleção, em seguida, iterativamente relaciona os resultados de maior pontuação interligando-os e bloqueando todas as hipóteses que contradizem esta ligação. Dado um par de árvores $\langle S,T \rangle$ que não possui nenhum nó lexical alinhado entre S e T, esta hipótese é pontuada com valor zero.

Um algoritmo básico é apresentado por Tinsley et al. (2007):

[1] `http://code.google.com/p/giza-pp`

```
Inicialização
Para cada nó não terminal da árvore fonte "s" faça
      Para cada nó não terminal da árvore alvo "t" faça
      Gere a hipótese de pontuação  γ(⟨s,t⟩)
      Fim do Para
Fim do Para
Bloqueie todas as hipóteses pontuadas como zero.
```

Após gerar a pontuação, o algoritmo de seleção é executado como segue:

```
Seleção
Enquanto há hipóteses não bloqueadas permaneça fazendo
 Alinhe e bloqueie a hipótese de maior pontuação
 Bloqueie todas as hipóteses contraditórias
Fim do Enquanto
```

Para gerar a pontuação, a seguinte fórmula é aplicada:

$$\gamma(\langle s,t\rangle) = \alpha(s_l|t_l) \bullet \alpha(t_l|s_l) \bullet \alpha(\bar{s_l}|\bar{t_l}) \bullet \alpha(\bar{t_l}|\bar{s_l})$$

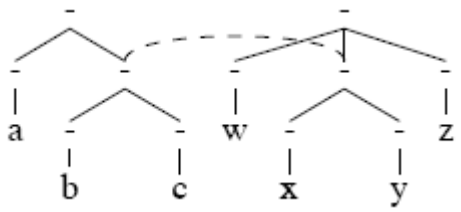
Onde:

$$s_l = s_i \ldots s_{ix} \quad e \quad t_l = t_j \ldots t_{jy}$$

denotam os terminais s e t respectivamente sendo (s, t) referente às hipóteses e

$$\bar{s_l} = S_1 \ldots s_{i-1} s_{ix+1} \ldots S_m \quad e \quad \bar{t_l} = T_1 \ldots t_{j-1} t_{jy+1} \ldots T_n$$

denotam os terminais S e T respectivamente sendo (S , T) um par das árvores.



s_l	=	b c
t_l	=	x y
$\overline{s_l}$	=	a
$\overline{t_l}$	=	w z

Figura 2.3: Valores de $s_l, t_l, \bar{s_l}$ e $\bar{t_l}$ dado um par da árvore e a hipótese de ligação (TINSLEY et al., 2007)

Na figura 2.3, podemos notar que s_l e t_l são tratados como subárvores, contendo os conjuntos de terminais b, c e x, y respectivamente.

Tinsley et al. (2007), utilizaram duas variações para calcular o valor de α, usado na fórmula

de cálculo da pontuação. Estas variações são apresentadas na figura 2.4.

Score *score1*

$$\alpha(x|y) = \prod_{j}^{|y|} \sum_{i}^{|x|} \mathrm{P}(x_i|y_j)$$

Score *score2*

$$\alpha(x|y) = \prod_{i}^{|x|} \frac{\sum_{j}^{|y|} \mathrm{P}(x_i|y_j)}{|y|}$$

Figura 2.4: Cálculos aplicados por Tinsley et al. (2007) para gerar a pontuação do relacionamento entre os nós usando a probabilidade do GIZA++

Na primeira variação, é realizada a soma das probabilidades de alinhamento para cada nó terminal pertencente à subárvore. Posteriormente é calculado o produtório destas somas. A segunda variação se difere por normalizar a soma pela quantidade de alinhamento deste nó terminal.

O estudo de Tinsley et al. (2007) foi realizado com as línguas Inglês e Francês, usando o *HomeCentre Corpus*, o qual contém 810 pares de sentenças alinhadas.

Tinsley et al. (2007) utiliza a divisão da árvore em partes, denominadas subárvores, da mesma forma que os sistemas DOT (*Data Oriented Translation*). O sistema DOT (POUTSMA, 1998, 2003) pode ser descrito como "um modelo híbrido de tradução que combina exemplos, informação linguística e estatística" (HEARNE; WAY, 2006). Em DOT, um modelo de tradução é aprendido a partir de árvores sintáticas fonte e alvo alinhadas.

O método de Menezes e Richardson (2001), já mencionado anteriormente, é um método de alinhamento e aquisição de regras de transferência utilizado no sistema WindowsMT. Esse sistema adquire o mapeamento de transferências alinhando pares de formas lógicas (FL), ou seja, após analisar sintaticamente as árvores fonte e alvo, é extraído o lema das palavras de conteúdo(*content Word*) como substantivos, verbos, adjetivos e advérbios, que juntamente com arcos direcionados e rotulados, constitui uma FL. A FL resume os diferentes aspectos de uma linguagem particular como a ordem dos constituintes, flexões morfológicas, e determinadas funções das palavras. Esses alinhamentos são obtidos por meio de análises das sentenças alinhadas em um *corpus* bilíngue. Um exemplo de forma lógica pode ser visto na figura 2.1.

Esse método se divide em duas fases. A primeira estabelece uma tentativa de correspondência lexical entre os nós fontes e alvos de um par de FL, e a segunda é o alinhamento dos demais nós baseado nesta correspondência lexical, considerando-se as estruturas das árvores. O alinhamento entre os nós é visto como um mapeamento entre um nó ou conjunto de nós relacionados de forma lógica fonte e um nó ou conjunto de nós relacionados de forma lógica alvo sendo que nenhum nó

pode participar de mais de um relacionamento. Nesse processo, é usado um dicionário bilíngue juntamente com um componente de derivação morfológica.

Esse componente de derivação morfológica aplicado por Pentheroudakis e Vanderwende (1993) nos permite identificar sistematicamente classes de palavras morfologicamente relacionadas. O componente de derivação morfológica consiste em extrair o núcleo de cada palavra do dicionário por meio de análise morfológica. Se esse núcleo possibilitar a derivação de uma ou mais palavras ele é classificado por pontuação baseado na comparação da informação semântica da forma derivada com a informação armazenada em uma base com termos supostamente similares. Por exemplo o substantivo *conversion* é analisado como [[*convert*]+*ion*] e [[*converse*]+*ion*]. Um algoritmo de pontuação é aplicado e o núcleo *convert* recebe uma pontuação mais alta que o núcleo *converse*. Ambas as possibilidades são armazenadas e as pontuações associadas são consideradas como valores de atributos, os quais expressam relações de derivação.

O dicionário bilíngue e os componentes de derivação morfológica são usados para definir as combinações, limitando-se às regras que conservam a estrutura da árvore em seu estado inicial.

Menezes e Richardson utilizaram um léxico bilíngue contendo 88.500 pares de tradução na língua Inglês-Espanhol.

O alinhamento dos nós é realizado utilizando-se as tentativas de correspondência lexical estabelecidas na primeira fase e a criação das estruturas dos alinhamentos de nós é feita com base em um conjunto de 18 regras gramaticais para alinhamento, as quais permitem somente alinhamentos com significado linguístico. Essas regras são ordenadas para criar um alinhamento de forma inequívoca, e a partir desses alinhamentos tratar os casos ambíguos. Um exemplo de alinhamento gerado com base nesse método já foi apresentado anteriormente na figura 2.2.

Outra estratégia de alinhamento apresentada por Marecek et al. (2008) propõe o alinhamento de árvores sintáticas baseado na camada tectogramatical (transição entre a camada sintática e semântica), que representa uma análise sintática mais aprofundada. Essa camada descreve as relações existentes entre o verbo principal e os elementos dependentes. De acordo com Marecek et al., é melhor trabalhar na camada tectogramatical por haver uma maior similaridade entre as árvores.

O algoritmo propõe o alinhamento em duas fases. Assim como Tinsley et al. (2007), estabelece-se a restrição de que um nó da árvore fonte só deve ser alinhado com um único nó na árvore destino, isto é, um relacionamento de $1:1$. Na segunda fase, um algoritmo busca os nós desalinhados e tenta relacioná-los aos nós já alinhados na árvore oposta permitindo, assim, um relacionamento de $1:N$.

Na primeira fase, os nós com maiores potenciais de alinhamento são considerados. Esta potencialidade é medida usando propriedades individuais de cada par das árvores, tratadas aqui como *recurso*.

Para cada par da árvore (S_i, T_j), é atribuída uma pontuação calculada da seguinte forma:

$$S(c_i\ ,\ e_j) = \vec{w} \bullet \vec{f}\ (c_i\ ,\ e_j)\ ,$$

Onde:

"c_i" é o i-ésimo nó na árvore fonte (em Tcheco) e "e_j" é o j-ésimo nó na árvore alvo (em Inglês). O "w" é um vetor de pesos do recurso obtidos por meio do treinamento de uma rede perceptron e "f" é o vetor de valores do recurso. Esses valores podem ser binários, inteiros ou reais. Em (MARECEK et al., 2008), os autores citam 15 recursos projetados para analisar: o lema, a probabilidade de tradução usando um dicionário, o alinhamento e a probabilidade de tradução usando o GIZA++, a análise do prefixo, o alinhamento dos nós ascendentes e descendentes e a similaridade da posição linear determinada pela organização das palavras na sentença. Os 15 recursos são:

1. **Par de lemas no dicionário:** retorna o valor binário 1 caso encontre termos com o lema similar no dicionário bilíngue.

2. **Probabilidade de tradução a partir do dicionário:** traz um valor real referente à probabilidade de tradução do lema a partir do dicionário. Esta probabilidade está incluída no *corpus* PCEDT.

3. **Alinhamento pelo GIZA++, intersecção:** retorna um binário igual a 1 se dois nós foram alinhados pelo GIZA++ com a simetrização de intersecção.

4. **Alinhamento pelo GIZA++, grow-diag-final:** retorna um binário igual a 1 se dois nós foram alinhados pelo GIZA++ com a simetrização grow-diag-final.

5. **Probabilidade de tradução a partir do GIZA++:** retorna um valor real referente à probabilidade de tradução do lema a partir da tabela de tradução gerada pelo GIZA++, em ambos os sentidos (Inglês-Tcheco e Tcheco-Inglês).

6. **Igualdade do Lema:** retorna o binário 1 se o lema em Tcheco for a mesma string que o lema em Inglês.

7. **Igualdade no número de prefixo:** retorna o binário 1 se o lema em Tcheco e em Inglês começarem com a mesma sequência de dígitos.

8. **5 letras iguais:** retorna o binário 1 se as cinco letras do prefixo nos lemas forem iguais.

9. **4 letras iguais:** retorna o binário 1 se as quatro letras do prefixo nos lemas forem iguais e não tenha se aplicado o recurso anterior(5 letras iguais).

10. **3 letras iguais:** retorna o binário 1 se as três letras do prefixo nos lemas forem iguais e não tenha se aplicado o recurso anterior(4 letras iguais).

11. **Alinhamento dos nós ascendentes:** retorna o binário 1 se o nó ascendente ao nó analisado na árvore em Tcheco já estiver alinhado ao nó ascendente analisado na árvore em Inglês.

12. **Alinhamento dos nós descendentes:** retorna o binário 1 se o nó descendente ao nó analisado na árvore em Tcheco já estiver alinhado ao nó descendente analisado na árvore em Inglês.

13. **CoAp:** retorna o binário 1 se ambos os nós forem raiz de construções coordenativas ou apositivas.

14. **Mesmo Part-of-Speech :** retorna o binário 1 se ambos os nós possuírem o mesmo *Part-of-Speech*.

15. **Similaridade da Posição Linear:** retorna o valor relativo à posição linear de cada nó (inicialmente armazenado no atributo *deepord*) subtraído do valor 1.

Em cada iteração, os pares com a melhor pontuação são alinhados, na próxima iteração a pontuação dos nós é atualizada, até que todos os nós estejam alinhados. Esta atualização é necessária porque alguns recursos podem sofrer influência dos pares já alinhados.

Para esse trabalho foram selecionadas sentenças de textos paralelos do *corpus* gerado pelo PCEDT (*Prague Czech-English Dependency Treebank*). As sentenças foram analisadas de forma automática usando o sistema TectoMT[2](ŽABOKRTSKÝ et al., 2008). Para a análise morfológica das expressões em Tcheco foi utilizado o *Prague Dependency TreeBank* na versão 2.0 chamado de PDT2.0 (HAJIC et al., 2006) e para a sintática, o *parser* MST (MCDONALD et al., 2005) com a posterior conversão automática em árvores. As sentenças em Inglês foram etiquetadas pelo *parser* TnT (BRANTS, 2000) e analisadas pelo *parser* de Collins (1999), também automaticamente convertidas em árvores.

[2]`http://ufal.mff.cuni.cz/tectomt/`

Os lemas foram extraídos de todas as árvores e ordenados de acordo com o atributo *deepord* e dado como entrada para a ferramenta GIZA++ (OCH; NEY, 2003). O atributo *deepord* descreve a organização das palavras em uma frase e determina a posição linear do nó na árvore. Note que não há informações sobre a estrutura da árvore ou outros atributos no processo de alinhamento da ferramenta GIZA++. Também foi usada a tabela de probabilidade gerada pelo GIZA++.

Comparando o método de Marecek et al. (2008) com o método de Tinsley et al. (2007), pode-se perceber que o primeiro busca uma similaridade maior entre os nós usando as propriedades citadas acima, enquanto o segundo analisa cada subárvore apenas com a probabilidade gerada pelo GIZA++.

Além dessas duas abordagens relevantes para o projeto aqui apresentado, há também uma proposta bastante recente de Tiedemann e Kotzé (2009) que descreve um algoritmo de alinhamento baseado em um modelo Log-Linear que prediz o relacionamento entre os nós. O modelo Log-Linear subdivide os processos de recursos e os agrupa no final para gerar o resultado. É aplicado um peso à probabilidade gerada pelos recursos associados a esses nós. Este peso é calculado usando os dados de treinamento.

Relacionar os nós considerando-os de forma independente pode causar problemas devido às dependências de relacionamento entre eles. Assim, para tentar contornar esse problema foram usados, nesta implementação, recursos baseados no histórico dos dados e um processo de classificação sequencial. Essa estratégia, chamada de predição da estrutura, utiliza a classificação global anterior como recurso de entrada para predizer a próxima classificação. Outra forma de predizer o alinhamento da estrutura da árvore é utilizando o que chamamos de *greedy Best-first*, o qual calcula o maior valor entre os nós candidatos.

Pode-se, ainda, aplicar outras restrições e critérios de boa formação como: relacionar os nós descendentes da árvore fonte apenas com nós descendentes da árvore alvo, relacionar os nós ascendentes de uma árvore fonte apenas com os nós ascendentes da árvore alvo e restringir as relações dos nós terminais apenas com nós terminais e nós não terminais apenas com nós não terminais.

O trabalho de Tiedemann e Kotzé (2009) possui uma abordagem rica em recursos apresentados da seguinte forma:

- Recursos Básicos de Alinhamento
- Recursos Contextuais
- Recursos Complexos

- Recursos de Dependência do Link

Os recursos básicos de alinhamento podem trabalhar com qualquer função de valor real sem considerar a dependência entre os nós. Isto é possível devido à flexibilidade que os modelos Log-lineares possuem, como o fato de utilizar um classificador binário, definindo a probabilidade de se relacionar dois nós. Essas probabilidades são geradas usando os recursos associados a estes nós.

Enquanto o modelo de Tinsley et al. (2007) introduz o recurso em nível lexical por meio da probabilidade gerada pelo GIZA++ como apresentado na figura 2.4, na implementação de Tiedemann e Kotzé (2009) uma pequena mudança na forma que ocorre a pontuação foi realizada como ilustrado na figura 2.5. Agora, a pontuação máxima para cada *token* é selecionada com base na probabilidade lexical, enquanto Tinsley et al. usavam a média da soma em relação a todos os possíveis relacionamentos entre os nós.

$$\begin{aligned} \gamma(s_l,t_l) &= \alpha(s_l|t_l)\alpha(t_l|s_l)\alpha(\overline{s_l}|\overline{t_l})\alpha(\overline{t_l}|\overline{s_l}) \\ \alpha(x|y) &= \prod_{x_i \geq x} max_j P(x_i|y_j) \end{aligned}$$

Figura 2.5: Cálculo aplicado por Tiedemann e Kotzé (2009) para gerar a pontuação do relacionamento entre os nós usando a probabilidade do GIZA++

Por fim, um outro recurso foi extraído em nível lexical que mede a consistência do relacionamento entre os nós com relevância como apresentado na figura 2.6.

$$\begin{aligned} align(s_i,t_j) &= \sum_{L_{xy}} consistent(L_{xy},s_i,t_j) / \sum_{L_{xy}} relevant(L_{xy},s_i,t_j) \\ consistent(L_{xy},s_i,t_j) &= \begin{cases} 1 & \text{if } s_x \geq s_i \wedge t_y \geq t_j \\ 0 & \text{otherwise} \end{cases} \\ relevant(L_{xy},s_i,t_j) &= \begin{cases} 1 & \text{if } s_x \geq s_i \vee t_y \geq t_j \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Figura 2.6: Função para calcular o grau de consistência no relacionamento entre dois nós aplicado por Tiedemann e Kotzé (2009)

Foi aplicado por Tiedemann e Kotzé (2009) o alinhamento de palavras *Viterbi*, produzido pelo GIZA++ usando o modelo IBM4 (BROWN et al., 1993), em ambas as direções, e usando a união e a intersecção dos resultados.

Seguindo a mesma estratégia de (MARECEK et al., 2008), Tiedemann e Kotzé (2009) também adotam recurso de valor binário definido entre os pares de nó terminais: caso os nós estejam relacionados é dado o valor 1. Este recurso é importante para modelos que incluem o alinhamento de nós terminais.

A posição relativa de cada nó na árvore possibilita gerar mais dois recursos, desta vez independentes de ferramentas externas. O primeiro é a similaridade dos nós junto ao nível na árvore (*Tree-Level Similarity* TLS) e o segundo, a similaridade dos nós em relação ao tamanho da árvore (*Tree Span Similarity* TSS). Para calcular o TLS é analisada a distância do nó em relação à raiz da árvore normalizada pelo tamanho da árvore. O TSS é calculado de acordo com a posição relativa do nó "na horizontal" normalizado pela quantidade de nós terminais da árvore. Nos recursos básicos, ainda é possível usar as categorias dos nós não terminais e os *part-of-speech*, de forma que, seja atribuído o valor 1 caso sejam iguais em ambas os nós comparados , ou 0 caso contrário.

No segundo conjunto de recursos definidos por Tiedemann e Kotzé (2009), os recursos contextuais visam explorar a estrutura da árvore como um todo, diferentemente dos recursos básicos que avaliam os nós diretamente ligados uns aos outros, ou seja, ascendentes ou descendentes. De modo simplificado, os nós herdam os valores atribuídos a cada nó descendente, ascendente e nós relacionados diretamente à mesma subárvore.

A combinação de alguns recursos já citados pode resultar em novas funcionalidades dando origem aos chamados recursos complexos. Para combinar os recursos "simples" no intuito de gerar recursos complexos, Tiedemann e Kotzé (2009) utilizam o produto dos valores dos recursos. Analisando a estrutura da árvore, dois novos recursos são extraídos: o recurso *children_links* e o *subtree_links* formando, assim, os recursos de dependência do Link. O primeiro recurso é o número de links diretos existentes na iteração atual entre os nós descendentes do nó analisado no momento. O segundo recurso é o número de links existente em toda a subárvore e não somente os descendentes diretos.

Para avaliar a estratégia proposta, Tiedemann e Kotzé (2009) utilizaram o *Smultron Treebank*; um *Treebank* paralelo com sentenças em três línguas: o Inglês, o Sueco e o Alemão. Tal *corpus* contém cerca de 500 sentenças, 6.671 ligações confiáveis e 1.141 duvidosas. As cem primeiras sentenças do *corpus* foram usada para treinamento e as demais para teste.

Como já mencionado anteriormente, Tinsley et al. (2007) utilizam a divisão da árvore em partes, denominadas subárvores, da mesma forma que as traduções orientadas a dados (*Data-Oriented Translation* ou DOT) (POUTSMA, 2000). Groves et al. (2004), em um método similar,

também dividem as árvores em subconjuntos aos quais dá o nome de fragmentos da árvore. Seu algoritmo alinha automaticamente fragmentos da árvore fonte com o fragmento da árvore alvo equivalente à tradução, de modo rápido e consistente. Essa abordagem, assim como Menezes e Richardson (2001), utiliza a estratégia *best-first* para alinhar a estrutura de dependência da árvore.

O algoritmo de alinhamento de árvores sintáticas de Groves et al. (2004), assim como os demais autores, se inicia com a correspondência lexical entre a língua fonte e alvo de modo *bottom-up*. Como em Menezes e Richardson (2001), é usada a estratégia *best-first* após o alinhamento lexical. Esse processo de alinhamento é recursivo e, após cada etapa, os novos pares de nós relacionados são adicionados a uma lista. As funções do algoritmo são aplicadas para cada novo par de nós até que não haja mais pares desalinhados. Essas funções são as cinco apresentadas a seguir:

1. **Alinhamento do Verbo + Objeto:** alinha-se o objeto do verbo entre ambas as árvores fonte e alvo quando os nós relacionados anteriormente forem verbos e os nós mais à esquerda das respectivas árvores. Para tanto, o nó ascendente deve estar etiquetado como VP e os nós a serem alinhados devem ter a mesma etiquetação sintática.

2. **Alinhamento dos nós pais:** alinham-se os nós ascendentes quando todos os nós descendentes a estes estão alinhados. Caso haja somente um nó desalinhado na subárvore fonte e na subárvore alvo, estes podem ser alinhados entre si.

3. **Alinhamento dos nós filhos:** é um processo similar ao do alinhamento dos nós pais, mas com a diferença que se os nós ascendentes em ambas as subárvores possuem o mesmo número de nós descendentes com a mesma etiquetação, então os descendentes podem ser alinhados.

4. **Alinhamento dos nós NP/VP:** quando se tem dois substantivos alinhados, percorresse a árvore partindo-se dos nós terminais em direção ao nó raiz buscando-se o nó etiquetado como NP localizado mais acima na árvore. O mesmo ocorre com os verbos, buscando-se a etiquetação VP mais acima na árvore.

5. **Alinhamento de subárvores:** se as subárvores de um determinado par já alinhado forem isomórficas, os demais nós são alinhado de acordo com a similaridade na etiquetação sintática. Isto ocorre devido ao fato de que uma árvore não isomórfica pode conter subárvores com esta característica.

Para os experimentos realizados por Groves et al. (2004), foi usado o *Xerox Home Centre corpus*, no par de línguas Inglês-Francês e utilizado um *gold standard* para sua avaliação, em um

total de 605 pares de sentenças na avaliação da qualidade de alinhamento. Pava avaliar a qualidade da tradução foram usados oito conjuntos de treinamento/teste (os mesmos conjuntos usados em (HEARNE; WAY, 2003)) contendo 545 pares de sentenças em cada conjunto de treinamento e 60 sentenças em cada conjunto de teste.

Dos métodos estudados, Gildea (2003) propõe uma abordagem que não prioriza a estrutura da árvore, enquanto para vários outros, como Tinsley et al. (2007), manter a estrutura original é um de seus objetivos. Essa abordagem realiza o alinhamento de árvores, mas o trata como parte da geração de um modelo estatístico de tradução.

Outra proposta que utiliza o alinhamento de árvores sintáticas como meio e não como fim foi apresentada por Lavie et al. (2008) e descreve o aprendizado de traduções equivalentes em nível subsentencial e a geração de regras a partir dos fragmentos de árvores alinhados. O foco de Lavie et al. está em extrair frases e regras a partir dos alinhamentos sintáticos. Na fase de alinhamento das árvores utiliza-se apenas o alinhamento de palavras.

Geralmente, o alinhamento lexical deixa alguns nós terminais sem alinhamento. No entanto, o algoritmo de alinhamento e fatorização prima (*Prime Factorization and Alignments, PFA*)proposto por Lavie et al. (2008) permite que um nó seja alinhado, independente da ordem das palavras expressas pela relação de precedência linear, como parte de um texto dominado por um nó que abrange o alinhamento de palavras ao mesmo nível do nó desalinhado na árvore oposta.

O PFA deve seguir os critérios de boa formação da estrutura da árvore. Este é um passo requerido por (TINSLEY et al., 2007), (MENEZES; RICHARDSON, 2001), (MARECEK et al., 2008), (TIEDEMANN; KOTZÉ, 2009) e até mesmo por (GILDEA, 2003) que permite alterar a estrutura da árvore.

O algoritmo de alinhamento e fatorização prima utiliza o mapeamento aritmético, o qual atribui um valor numérico a cada nó terminal, que serve como um identificador único. Este mesmo número é atribuído ao nó terminal alinhado na árvore alvo. São usados apenas números primos nesta atribuição. Para os nós sem correspondência lexical (nós terminais desalinhados) é atribuído o valor 1. Após este passo, os valores dados aos nós terminais se propagam aos nós não terminais ascendentes, ao qual é atribuído o produto derivado dos nós terminais descendentes pertencentes a este nó não terminal. Caso haja um nó não terminal com o mesmo valor em ambas as árvores estes são relacionados.

A figura 2.7 ilustra o alinhamento das árvores sintáticas usando o algoritmo PFA. Os pares de nós não terminais alinhados possuem a mesma forma geométrica em ambas ás árvores, enquanto os nós terminais possuem a mesma cor e são segmentados pela linha contínua.

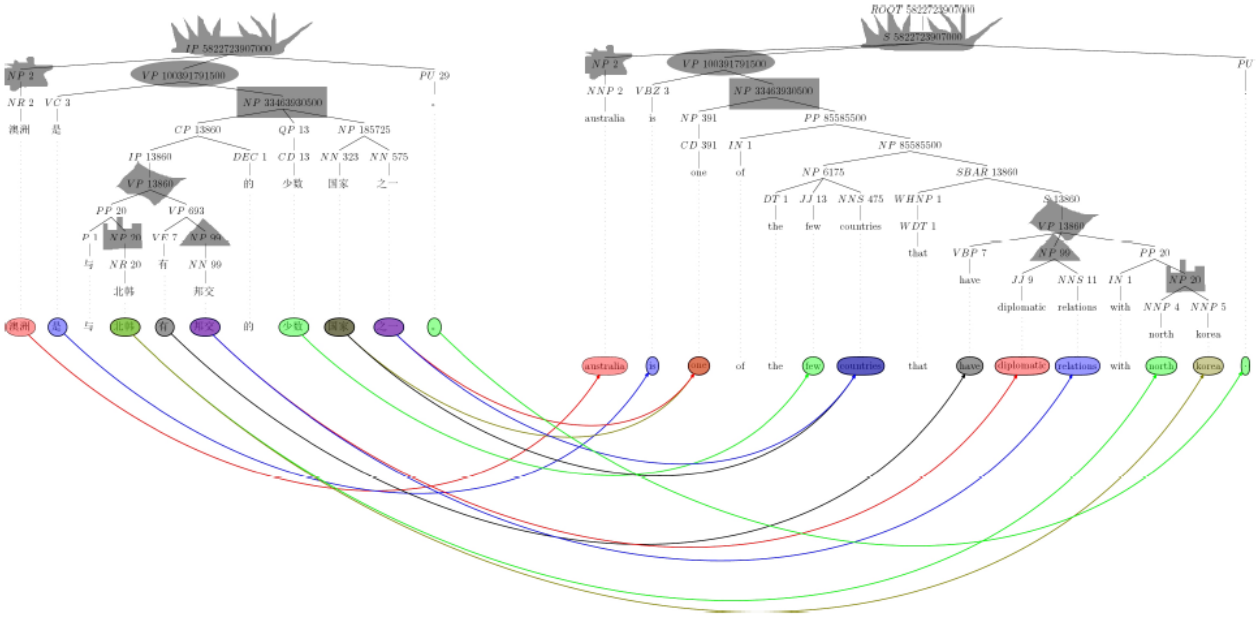


Figura 2.7: Exemplo de nós alinhados pelo algoritmo PFA (LAVIE et al., 2008)

Foi usado por Lavie et al. um *corpus* Chinês-Inglês com 3342 sentenças paralelas, alinhadas manualmente a nível lexical.

Dos trabalhos estudados, percebe-se que os autores subdividem o alinhamento de árvores sintáticas em duas etapas. Primeiramente, são alinhados os nós terminais da árvore baseado no alinhamento lexical. Esse alinhamento lexical geralmente é retirado de um léxico bilíngue – uma espécie de dicionário com os termos previamente alinhados manualmente – ou obtido automaticamente, principalmente com base em estatística. Na etapa seguinte, os demais nós são alinhados, levando-se em consideração regras de composição dos nós previamente definidas, as probabilidades de ligação de um nó fonte com um nó alvo, programação dinâmica, dentre outros.

Na proposta de Menezes e Richardson (2001), o algoritmo de alinhamento busca encontrar pares de tradução em um léxico bilíngue analisando a correspondência lexical existente entre nós fonte e alvo. Após alinhar os pares encontrados usando o léxico, baseado em uma estratégia na qual os nós com melhores correspondência lexical são considerados a princípio (*Best-first*).

Na proposta de Marecek et al. (2008) podemos perceber que seu modelo também faz uso de um léxico bilíngue e, em seguida, assim como Menezes e Richardson (2001), aplica regras de composição para alinhar os nós não terminais das árvores.

Diferentemente, os métodos de Tinsley et al. (2007) e Tiedemann e Kotzé (2009) utilizam a probabilidade de alinhamento de cada nó das árvores fonte e alvo. Com essa técnica, Tinsley et al. conseguem tornar seu modelo independente de idiomas, o que é uma das vantagens de seu algo-

ritmo, a mesma característica é encontrada em Tiedemann e Kotzé (2009). Vale dizer, ainda, que no trabalho de Tiedemann e Kotzé, o método de Tinsley et al. foi implementado apenas como um dos recursos básicos de alinhamento, sofrendo uma alteração na função de cálculo da pontuação baseada na probabilidade gerada pelo GIZA++.

É possível notar uma semelhança em relação às regras de composição mais comuns entre os autores, as quais estão direcionadas a manter a estrutura da árvore alinhada, mantendo uma dependência entre os nós descendentes e ascendentes já relacionados, não permitindo a ligação de forma cruzada na estrutura das árvores.

Para este trabalho, os métodos de Tinsley et al. e Lavie et al. foram implementados e usados como base para o desenvolvimento de 3 novos modelos. Estes métodos foram escolhidos por utilizarem recursos linguísticos disponíveis para os idiomas inglês e português do Brasil e ambos utilizam árvores de constituintes. Além disso, Tinsley et al. realizam uma abordagem Top-Down enquanto Lavie et al. utilizam a abordagem Bottom-Up, permitindo assim dois tipos distintos de abordagens não priorizando características específicas de determinados modelos. Outra vantagem de se usar estes dois métodos está na forma que ambos analisam a estrutura das árvores, Tinsley et al. consideram em seu método toda a estrutura da árvore e Lavie et al. consideram apenas os nós terminais pertencentes ao domínio do nó não terminal analisado no momento.

Tiedemann e Kotzé (2009) aplicam em seu método um classificador de máxima entropia, um toolbox chamado Megam, além de ter uma complexidade computacional maior que os métodos de Tinsley et al. e Lavie et al.. Da mesmo forma, o método de Marecek et al. (2008) não é viável por usar a camada tectogramatical, um recurso linguístico não disponível para o português do Brasil.

Por fim, Menezes e Richardson (2001) e Groves et al. (2004) usam regras estipuladas manualmente enquanto a intenção deste trabalho é automatizar toda a tarefa capaz de extrair conhecimento linguístico.

2.2 Métodos de Avaliação dos alinhamentos de árvores sintáticas

A avaliação dos métodos de alinhamento de árvores sintáticas pode ser realizada considerando-se o alinhamento dos nós fonte e alvo propriamente dito ou o uso desse alinhamento (ou possivelmente de "conhecimento" derivado desse alinhamento) em alguma aplicação de PLN, geralmente a tradução automática. No primeiro caso, dizemos que a avaliação é intrínseca enquanto, no segundo, trata-se de uma avaliação extrínseca dos alinhamentos gerados automaticamente. Além

disso, tanto na avaliação intrínseca quanto na extrínseca pode-se usar um conjunto de referência (*gold standard*) composto, por exemplo, por árvores alinhadas manualmente (avaliação intrínseca) ou sentenças traduzidas por humano (avaliação extrínseca). Esse conjunto de referência é considerado correto e, portanto, usado na comparação automática com os alinhamentos ou traduções gerados pelos métodos que se pretende avaliar.

Para avaliação de seus métodos, Tinsley et al. (2007) , Marecek et al. (2008) e Tiedemann e Kotzé (2009) usaram *gold standards* com árvores sintáticas paralelas alinhadas manualmente por especialistas na área de linguística. Menezes e Richardson (2001) submeteram as saídas de seu sistema de tradução automática à avaliação de cinco especialistas em linguística.

Na avaliação de Tinsley et al. (2007), foi utilizado um *corpus* com 810 árvores sintáticas retirado do *Corpus HomeCentre*. Oito possíveis combinações entre a forma de calcular a pontuação (*score1 e score2*) e a forma de tratar os casos com a mesma pontuação (*skip1 e skip2*), fazendo uso ou não do *Span*, foram avaliadas. O alinhamento manual necessário para a criação do *gold standard* foi realizado por um tradutor nativo do Inglês com proficiência em Francês. A tabela 2.1 demonstra os resultados obtidos na avaliação intrínseca. As medidas de Precisão (Precision) e Cobertura (Recall) são detalhadas no capítulo 5. São apresentados os valores para a avaliação considerando todos os nós alinhados (all links), sendo nós terminais e não terminais, e considerando apenas os nós não terminais (*non-lexical links*).

Tabela 2.1: Resultado da avaliação intrínseca (TINSLEY et al., 2007)

Configurations	*all links*		*non-lexical links*	
	Precision	Recall	Precision	Recall
skip1_score1	0.6096	0.7723	**0.8424**	0.7394
skip1_score2	0.6192	0.7869	0.8107	0.7756
skip2_score1	0.6162	0.7783	0.8394	0.7486
skip2_score2	0.6215	0.7867	0.8107	0.7756
skip1_score1_span1	0.6229	**0.8101**	0.8137	0.7998
skip1_score2_span1	0.6220	0.7963	0.8027	0.7871
skip2_score1_span1	**0.6256**	0.8100	0.8139	**0.8002**
skip2_score2_span1	0.6245	0.7962	0.8031	0.7871

Também foi realizada uma avaliação extrínseca na qual esses alinhamentos foram usados para treinar um sistema DOT (POUTSMA, 2003) e, em seguida, a qualidade de tradução gerada por meio desse sistema de TA foi analisada por meio de três métricas diferentes: BLEU (PAPINENI et al., 2002), NIST (DODDINGTON, 2002) e METEOR (LAVIE; AGARWAL, 2007)

apresentando os resultados da tabela 2.2.

Tabela 2.2: Resultado da avaliação extrínseca (TINSLEY et al., 2007)

Configurations	BLEU	NIST	METEOR	Coverage
manual	0.5222	6.8931	71.8531%	68.5417%
skip1_score1	0.5038	6.8673	71.3805%	71.8750%
skip1_score2	0.5296	6.8557	72.7302%	***72.5000%***
skip2_score1	0.5091	6.9145	71.7764%	71.8750%
skip2_score2	***0.5333***	6.8855	72.9615%	***72.5000%***
skip1_score1_span1	0.5258	6.9004	72.5916%	***72.5000%***
skip1_score2_span1	0.5285	6.8452	***73.0014%***	***72.5000%***
skip2_score1_span1	0.5273	***6.9384***	72.7157%	***72.5000%***
skip2_score2_span1	0.5290	6.8762	72.8765%	***72.5000%***

Para o trabalho de Menezes e Richardson (2001), a métrica de avaliação escolhida foi a análise do resultado de saída por especialistas humanos para saber a qualidade da tradução aplicando o algoritmo descrito, juntamente com uma máquina de tradução. A avaliação foi realizada por cinco indivíduos encarregados de verificar as traduções produzidas e comparar a sentença gerada com uma sentença de referência produzida manualmente (Gold Standard). Esses avaliadores humanos qualificaram o resultado em uma escala de 1 a 4, tendo como pontos de análise a precisão e a fluência da tradução. Nessa avaliação utilizou-se um *corpus* nas línguas Inglês-Espanhol, composto por 208.730 pares de sentença, onde 161.606 pares foram usados na avaliação.

No primeiro experimento, o sistema comparou a qualidade de tradução com um sistema comercial, o Babelfish[3]. No segundo experimento, foi analisado o algoritmo Best-First e comparado com a abordagem *bottom-up*. O terceiro experimento usa um algoritmo que se difere do *Best-First*, uma vez que não mantém nenhum contexto ao emitir mapeamento de transferência. O algoritmo de comparação usado no experimento 4 se difere do *Best-First* pois não aplica o *threshold* da frequência, ou seja, todos os mapeamentos de transferência são retidos. Os resultados desses quatro experimentos são mostrados na tabela 2.3.

Marecek et al. (2008), por sua vez, validou seu alinhador usando 515 sentenças (aproximadamente 13.000 *tokens*). As sentenças foram alinhadas manualmente no nível de palavras. Os anotadores foram convidados a usar três tipos de alinhamento:

- Link Correto: quando duas palavras são idênticas;

[3] `http://babelfish.yahoo.com/`

Tabela 2.3: Qualidade da tradução (MENEZES; RICHARDSON, 2001)

System-A	System-B	Num. sentences System-A rated better	Num. sentences System-B rated better	Num. sentences neither rated better	Net percent improved sentences
Best-First	BabelFish	93 (46.5%)	73 (36.5%)	34 (17%)	10%
Best-First	Bottom-Up	224 (44.8%)	111 (22.2%)	165 (33%)	22.6%
Best-First	No-Context	187 (37.4%)	69 (13.8%)	244 (48.8%)	23.6%
Best-First	No-Threshold	112 (22.4%)	122 (24.4%)	266 (53.2%)	-2.0%

- Link Frasal: quando as frases se correspondem, mas as palavras não são correspondentes;
- Link Possível: as palavras se conectam não tendo uma equivalência real com outras línguas, mas sintaticamente pertencem claramente a uma palavra próxima, como por exemplo, o artigo na língua inglês.

A partir do *gold standard* gerado conforme descrito acima, as árvores alinhadas automaticamente foram avaliadas considerando-se precisão, cobertura, e medida-F para cada iteração. A precisão foi calculada como a porcentagem de pares alinhados pelo alinhador em relação aos pares alinhados manualmente, enquanto a cobertura indica quantos pares alinhados manualmente foram alinhados pelo alinhador. A medida-F é a média harmônica entre precisão e cobertura.

Também foi utilizado o alinhamento lexical por meio da ferramenta GIZA++ para avaliar essas três métricas: precisão, cobertura e medida-F. Esta ferramenta realiza no máximo uma conexão para cada palavra (alinhamento 1 : 1). Para unir os alinhamentos produzidos por GIZA++ nos dois sentidos, foram utilizados três métodos de simetrização: intersecção, união e *grow-diag-final* (OCH; NEY, 2003). O resultado é apresentado em termos da média e do desvio padrão na tabela abaixo.

Tabela 2.4: Resultado do 10 *fold cross-validation* (MARECEK et al., 2008)

n	1	2	3	4	5	6	7	8	9	10	mean	σ
P	93.63	93.40	93.52	95.49	88.65	92.83	95.22	92.12	92.19	93.72	93.08	1.91
R	87.92	89.78	90.66	91.25	80.30	87.43	89.47	89.88	82.58	90.14	87.94	3.65
F	90.69	91.55	92.07	93.32	84.27	90.04	92.25	90.98	87.12	91.90	**90.42**	**2.73**

Tiedemann e Kotzé (2009), para avaliar seu método usaram o *Smultron Treebank*, que possui três línguas sendo o Inglês, o Sueco e o Alemão. O alinhamento das sentenças do *gold standard* foi realizado manualmente usando a ferramenta TreeAligner[4]. Este alinhamento possui links classificados como confiáveis ou duvidosos para os quais adotou-se um peso três vezes maior aos links confiáveis.

[4] `http://www.cl.uzh.ch/kitt/treealigner`

O *corpus* principal usado para avaliação possui cerca de 500 sentenças, 6.671 ligações confiáveis e 1.141 duvidosas. As cem primeiras sentenças do *corpus* foram usadas para treinamento e as demais para teste. As medidas de avaliação foram a precisão, a cobertura e a medida-F em alguns conjuntos de recursos conforme mostrado na Tabela 2.5.

Tabela 2.5: Resultados para diferentes conjuntos de recursos (TIEDEMANN; KOTZÉ, 2009)

settings	Precision	Recall	F
lexical features	64.89	51.00	57.11
+ tree features	56.80	60.70	58.68
+ alignment features	62.94	62.83	62.88
+ label features	77.20	74.44	75.79
+ context features	**79.77**	**75.66**	**77.66**
train=novel, test=economy	81.49	74.76	77.98
train=economy, test=novel	77.67	75.04	76.33

Em uma nova versão do alinhador desenvolvida com base em algumas diretrizes do trabalho de Samuelsson e Volk (2007), os nós terminais não foram considerados no modelo alinhado manualmente para treinamento. Esta decisão diminui o número de nós relacionados e perda de informações no processo de aprendizagem levando a um aumento nos valores de cobertura (de 75,66% para 86,89%) e medida-F (de 77,66% para 79,46%) sendo que este último não foi maior porque houve um decréscimo na precisão (de 79,77% para 73,20%). A partir desses resultados, Tiedemann e Kotzé concluíram que um pequeno *corpus* é suficiente para a fase de aprendizado e extração de recursos.

Para o método de Groves et al. (2004), o *Xerox Home Centre corpus* foi usado com 605 pares de sentenças na língua Inglês-Francês. Aplicaram-se dois métodos distintos para avaliação, sendo o primeiro a comparação entre a saída do algoritmo e o *gold standard* alinhado manualmente. O segundo método de avaliação foi comparar a saída do algoritmo com um modelo gerado automaticamente pelo DOT de Hearne e Way (2003). Na avaliação utilizando o *gold standard* foram testadas quatro funções de maneira individual (resultados apresentados na tabela 2.6), sendo o alinhamento dos nós pais (Par), alinhamento dos nós NP/VP(NP/VP), alinhamento dos nós filhos (Child) e o alinhamento do verbo+objeto (Verb+Object), além de avaliar os valores para o alinhamento lexical (lex). As funções também foram avaliadas de maneira conjunta (resultados apresentados na tabela 2.7) aplicando também o alinhamento de subárvores.

Em relação aos valores das tabelas 2.6 e 2.7, é importante dizer que a baixa cobertura se deu por consequência do baixo desempenho do alinhamento de palavras. Além disso, das funções analisadas individualmente, o alinhamento dos nós pais (Par) alcançou o melhor desempenho com

Tabela 2.6: Resultado da análise individual das funções (GROVES et al., 2004)

	PRECISION	RECALL	F-SCORE
Lex	0.6800	0.3057	0.4212
Par	**0.7471**	**0.4983**	**0.5978**
NP/VP	0.7206	0.4879	0.5819
Child	0.7045	0.3856	0.4984
Verb + Object	0.6843	0.3191	0.4352

Tabela 2.7: Resultado da análise conjunta das funções (GROVES et al., 2004)

	PRECISION	RECALL	F-SCORE
Par + Child	0.7525	0.5588	0.6414
Par + NP/VP	0.7373	0.6106	0.6680
Par + Child + NP/VP	0.7411	0.6587	0.6974
All	**0.7430**	0.6686	0.7039
All + Subtree	0.7370	**0.6784**	**0.7064**

59,78% de medida-F e o melhor resultado foi alcançado usando todas as funções de forma conjunta (medida-F igual a 70,64%).

Um *gold standard* também foi usado para avaliar a abordagem de Lavie et al. (2008). Em uma primeira avaliação foram usadas 30 sentenças do *corpus* alinhadas manualmente por um especialista bilíngue. Esse *gold standard* foi comparado com a saída do algoritmo PFA usando o alinhamento manual a nível lexical com os resultados apresentados na tabela 2.8.

Tabela 2.8: Resultado da avaliação usando o alinhamento lexical manual (LAVIE et al., 2008)

Precision	Recall	F-1	F-0.5
0.8129	0.7325	0.7705	0.7841

Em um segundo momento de avaliação utilizou-se o alinhamento automático lexical (no lugar do alinhamento manual usado na primeira avaliação). O resultado é demonstrado na tabela 2.9.

Tabela 2.9: Resultado da avaliação usando o alinhamento lexical automático (LAVIE et al., 2008)

Comb Method	Prec	Rec	F-1	F-0.5
Intersection	0.6382	0.5395	0.5846	0.6014
Union	0.8114	0.2915	0.4288	0.5087
Sym1	0.7142	0.4534	0.5546	0.5992
Sym2	0.7135	0.4631	0.5616	0.6045
Grow-Diag-Final	0.7777	0.3462	0.4790	0.5493
Grw-Diag-Fin-And	0.6988	0.4700	0.5619	0.6011

A partir do exposto nessa seção, vê-se que a avaliação dos alinhamentos sintáticos para os métodos propostos pelos autores citados anteriormente necessita do conhecimento humano na geração de modelos alinhados de referência (Gold Standard). Tais modelos são usados não só

para avaliar os métodos de alinhamento como também para extrair o conhecimento na fase de aprendizado. Para tanto, a próxima seção apresenta um levantamento bibliográfico a respeito do processo de pré-processamento dos *corpora* usados no alinhamento de árvores sintáticas (treinamento, referência/teste e avaliação).

3 Tratamento do *Corpus*

Este capítulo descreve o principal recurso linguístico a ser usado na investigação dos métodos de alinhamento de árvores sintáticas: o *corpus* paralelo. Para tanto, a seção 3.1 apresenta os formalismos de representação da informação sintática, em especial o formalismo adotado neste projeto: o TigerXML. A seção 3.2, por sua vez, apresenta as ferramentas utilizadas na análise sintática do *corpus* a ser empregado no treinamento, teste e avaliação dos cinco modelos de alinhamento derivados desse trabalho.

3.1 Estudo dos formalismos de representação da informação sintática

Existem vários formatos para codificação de *corpora* anotados sintaticamente, entre eles podemos citar: Penn TreeBank[1], Suzanne[2] e NeGra[3]. Como as aplicações não suportam todos os tipos de codificação existentes, um possível formato para importação e exportação desses dados codificados é o XML. A seguir são apresentados os formatos usados pelas ferramentas de análise sintática utilizadas neste projeto: TigerXML (usado pelo PALAVRAS) e Penn TreeBank (usado pelo Collins).

3.1.1 O formato de codificação *Treebank* TigerXML

O formato TigerXML[4] foi designado como um formato de representação. Baseado em XML, ele é estruturado em etiquetas (tags). Uma ferramenta que processa a codificação no formato TigerXML é a TigerSearch[5]. Esta ferramenta permite a realização de consultas na estrutura do documento XML.

[1] `http://www.cis.upenn.edu/~treebank/`
[2] `http://www.grsampson.net/RSue.html`
[3] `http://www.grsampson.net/RSue.html`
[4] `http://www.ims.uni-stuttgard.de/projekte/TIGER/`
[5] `http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/`

Um documento TigerXML é composto por duas partes: o cabeçalho contendo informações sobre o *corpus* e alguns metadados, e o corpo do documento com definições do grafo de sintaxe que são grafos direcionados (da raiz para as folhas) de forma acíclica a partir de um único nó raiz e as anotações. O corpo do documento por sua vez, pode ser dividido em partes chamadas *subcorpora*.

O cabeçalho possui informações de metadados como: nome do *corpus*, autor, data, descrição, formato e história. A figura 3.1 traz o esboço da estrutura geral do *corpus* de teste usado neste projeto e que foi analisado sintaticamente pelo *parser* do Collins.

```
<corpus>
   <head>
      <meta>
         <name> Corpus Fapesp en com 108 arvores sintaticas geradas
                pelo parser de Collins (1999)
         </name>
         <format> Penn-Treebank Format </format>
         ...
      </meta>
      <annotation>
       Informações sobre as etiquetas e seus valores apresentadas na figura 3.2
      </annotation>
   </head>
   <body>
     Informações sobre as sentenças e suas árvores sintáticas (veja figura 3.3)
   </body>
</corpus>
```

Figura 3.1: Esboço do *corpus* de teste no formato TigerXML

As outras informações do cabeçalho são relativas aos atributos usados no *corpus*. Exemplos de atributos seriam "word" para os nós terminais e "cat" para os nós não terminais como apresentado na figura 3.2.

Logo após o cabeçalho, o corpo (`<body>`) segue um modelo de dados baseado nos grafos de sintaxe. Na figura 3.3 é apresentado um exemplo de saída do TigerXML para a sentença "*The faults of the Spheres*" e na figura 3.4 a representação gráfica dessa árvore acompanhada de sua tradução para o português. Na estrutura do documento percebe-se que os nós terminais (`<terminals>`) e os nós não terminais (`<nonterminals>`) aparecem como subelementos do nó `<s>`. O atributo "id" identifica esta árvore como "s7". Dentro do *corpus*, cada árvore recebe uma identificação única. Além disso, vemos que valores dos atributos representados por pares atributo-valor não podem ser omitidos.

```
<head>
   <meta>
    Informações sobre o corpus (veja figura 3.1)
   </meta>
   <annotation>
      <feature name="word" domain="T"/>
      <feature name="pos"  domain="T">
         <value name="!"/>
         <value name="CC"> Coordinating conjunction </value>
         <value name="CD"> Cardinal number </value>
         <value name="DT"> Determiner </value>
         ...
      </feature>
      <feature name="cat" domain="NT">
         <value name="ADJP"> Adjective Phrase </value>
         <value name="ADVP"> Adverb Phrase </value>
         ...
      </feature>
      <edgelabel>
         <value name="--"> not assigned </value>
      </edgelabel>
   </annotation>
</head>
```

Figura 3.2: Exemplo de informações referentes ao *corpus*, contidas no cabeçalho

Os nós terminais possuem um ou mais subelementos `<t>` conforme o número de *tokens* na sentença. Cada subelemento destes contém atributos como o "id" que se refere ao identificador do *token*, o "*word*" cujo valor atribuído é a palavra que está sendo disponibilizada no elemento, o "*pos*" que nos mostra sua categoria gramatical. Além desses atributos, alguns etiquetadores fornecem informações como o "*lemma*" que apresenta o lema da palavra, , o "*morph*" com informações morfológicas, a "*sem*" com informações semânticas e o "*extra*" com alguns dados extras do *token*.

Por sua vez, os nós não terminais possuem o subelemento `<nt>` que compõe a estrutura sintática de uma sentença. Para os nós `<nt>` são fornecidos os seguinte atributos:

- id – que identifica o nó `<nt>`
- cat – define a sua categoria, indicando o tipo de estrutura.

Os nós `<nt>` podem ter um ou mais subelementos etiquetados como `<edge>`. Este subelemento indica a estrutura interna da árvore por meio do atributo "idref", uma referência ao identi-

```
<s id="s7">
   <graph root="s7_500">
      <terminals>
         <t id="s7_1" word="The" pos="DT"/>
         <t id="s7_2" word="faults" pos="NNS"/>
         <t id="s7_3" word="of" pos="IN"/>
         <t id="s7_4" word="the" pos="DT"/>
         <t id="s7_5" word="spheres" pos="NN"/>
      </terminals>
      <nonterminals>
         <nt id="s7_501" cat="NP">
            <edge idref="s7_1" label="--"/>
            <edge idref="s7_2" label="--"/>
         </nt>
         <nt id="s7_503" cat="NP">
            <edge idref="s7_4" label="--"/>
            <edge idref="s7_5" label="--"/>
         </nt>
         <nt id="s7_502" cat="PP">
            <edge idref="s7_3" label="--"/>
            <edge idref="s7_503" label="--"/>
         </nt>
         <nt id="s7_500" cat="NP">
            <edge idref="s7_501" label="--"/>
            <edge idref="s7_502" label="--"/>
         </nt>
      </nonterminals>
   </graph>
</s>
```

Figura 3.3: Exemplo de uma sentença e suas anotações correspondentes à árvore sintática

ficador de um outro nó na estrutura da árvore. Para exemplificar, observe, na figura 3.3, que o nó não terminal com o atributo id="s7_501", pertencente à categoria "NP", possui dois subelementos `<edge>` nos quais os atributos "idref" fazem referência aos subelementos `<t>` dos nós terminais, com o atributo id="s7_1" e id="s7_2". Em outras palavras, os nós não terminais constituem um grafo onde cada subelemento `<edge>` é uma aresta. O nó `<edge>` pode fazer referência não somente a nós terminais como também a um outro nó não terminal como ilustrado na figura 3.3 para o nó identificado como "s7_500".

Quando um *corpus* é muito extenso, esse documento XML precisa ser dividido em vários arquivos. Para isto, o formato TigerXML incorpora um *link* para arquivos externos chamados subcorpora. Para incorporar os arquivos ao arquivo principal é utilizado o elemento `<subcorpus>` e os atributos "name" e "external" são setados pelo nome do *subcorpora* e a URL respectivamente

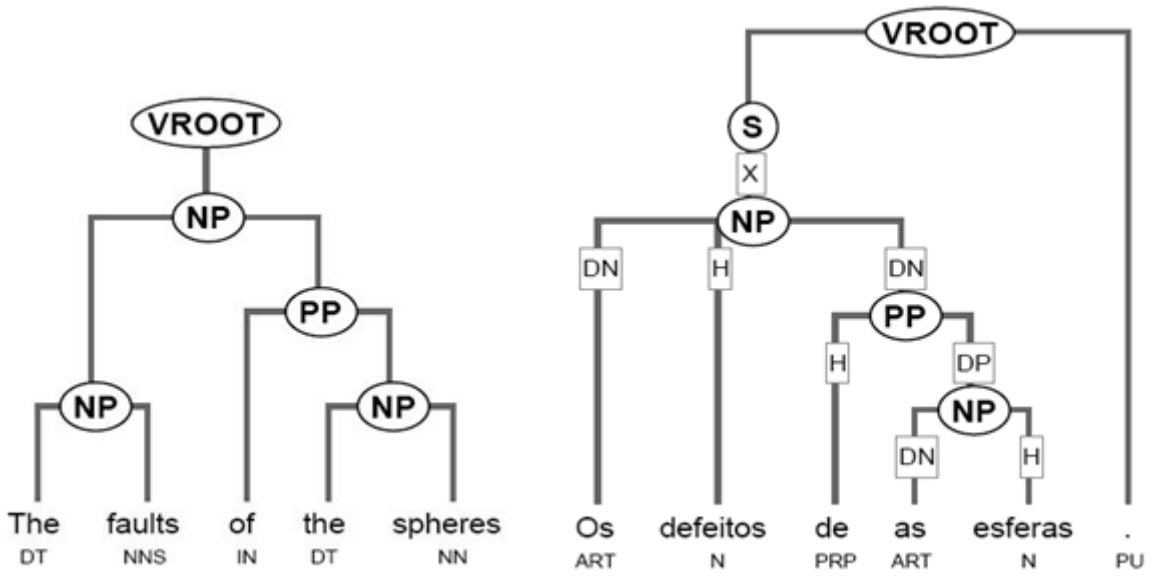


Figura 3.4: À esquerda, árvore da sentença em inglês relativa ao código da figura 3.3, e à direita, a mesma sentença em português

como apresentado na figura 3.5.

```
<corpus>
   <head>
   ...
   </head>
   <body>
      <subcorpus name="corpus Fapesp" external="file:subcorpus.xml"/>
   </body>
</corpus>
```

Figura 3.5: Exemplo de incorporação do subcorpora ao arquivo principal

3.1.2 O formato Penn TreeBank

O Penn Treebank, utilizado como base para o *parser* de Collins (1999), é um grande *corpus* anotado na língua inglesa com informações sintáticas e semânticas. O formato Penn Treebank consiste em etiquetas de *part-of-speech* e informações sintáticas, as quais são apresentadas em textos entre parênteses, como ocorre nos *corpora Wall Street Journal* e o *Corpus Brown*. Os seguintes *part-of-speech* podem ser percebidos na primeira sentença do exemplo na figura 3.6:

- **NP** – substantivo próprio no singular (Proper noun, singular)
- **DT** – artigos (Determiner)

```
(NP (NP (DT The) (NNS faults)) (PP (IN of) (NP (DT the) (NN spheres))))
(NP (NP (DT The) (NNS teeth)) (PP (IN of) (NP (DT the) (JJS oldest)
 (NN orangutan))))
(S (NP (NP (DT A) (JJ new) (NNS species)) (PP (IN of) (NP (NNP hominid))))
(VP (VBD found) (PP (IN in) (NP (NNP Thailand)))
```

Figura 3.6: Sentenças no formato Penn TreeBank. A primeira sentença é relativa à árvore da esquerda na figura 3.4

- **NNS**– substantivo no plural (Noun, plural)
- **PP** – pronome pessoal (Personal pronoun)
- **IN** – preposição ou conjunção subordinada (Preposition or subordinating conjunction)
- **NN** – substantivo no singular ou plural (Noun, singular or mass)

Neste projeto, a saída do *parser* de Collins em formato Penn TreeBank foi convertida para o formato TigerXML pela ferramenta TigerRegistry[6]. Esta ferramenta realiza a conversão de vários formatos como Penn TreeBank, Suzanne e NeGra para o formato TigerXML.

3.2 Pré-processamento do *corpus* para inserir informação sintática

A partir da década de 90, com a ascensão da aquisição do conhecimento por meio do aprendizado de máquina, tornou-se possível construir uma gramática sem a necessidade do conhecimento de um especialista, utilizando grandes bases de exemplo.

Um exemplo do uso dessas grandes bases de exemplo em processamento de língua natural pode ser dado na recuperação da estrutura sintática das sentenças. Essa análise sintática é realizada pelo *parser*, uma ferramenta capaz de recuperar esta estrutura sintática, utilizando uma gramática ou um *corpus*.

Utilizando técnicas de aprendizado e cálculos estatísticos baseados em *corpus*, é possível obter estas informações sintáticas usando um processo chamado de *parsing* probabilístico, como no modelo de Collins.

Também é possível gerar esta estrutura usando abordagens baseadas em regras, como é o caso do PALAVRAS (BICK, 2000). Este esquema de anotação se baseia no formalismo da gramática restritiva, introduzido por Karlsson (1990), Karlsson et al. (1995).

[6]http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/doc/html/TIGERRegistry.html

3.2.1 O *Parser* Palavras

O *parser* PALAVRAS foi desenvolvido por Bick (2000) com a finalidade de analisar sintaticamente estruturas em português. Faz parte do projeto VISL (*Visual Interactive Sintax Learning*)[7].

O sistema é baseado no formalismo *Constraint Grammar* (Gramática Constritiva CG) ou também conhecido como gramática restritiva e apoiado em um léxico com 50.000 lemas e milhares de regras gramaticais, permitindo a análise morfológica e sintática.

Determinadas palavras, quando analisadas isoladamente, são ambíguas quanto a flexão, função sintática, classe de palavras, conteúdo semântico, etc. Ao analisar o contexto é possível determinar a função e o significado em que estas palavras se encontram. Por meio de condições contextuais, a gramática restritiva tenta desambiguar as palavras usando um conjunto de regras selecionando a etiqueta correta no nível morfológico e semântico, ou seja, regras dependentes do contexto são compiladas em uma gramática que atribui etiquetas gramaticais a palavras ou símbolos.

A CG é uma gramática dependencial, utiliza símbolos como "@" para introduzir etiquetas de função sintática, e marcadores como "<" e ">" para indicar a direção do núcleo sintático de que os constituintes são dependentes. O principal fundamento da gramática de dependência é que a estrutura sintática é formada por dependências (relações binárias) ligando os nós lexicais. Se uma palavra é sensível ou dependente de alguma propriedade de outra palavra, então existe uma dependência entre as duas palavras.

A notação usada pela gramática restritiva, com marcadores de dependência em todos os níveis e um conjunto de etiquetas bem definido, não permite a apresentação gráfica de estrutura sintática. Para tanto, é necessário inserir marcadores de limite de constituintes (*constituint boundaries*) por uso de regras e transformar em uma notação de árvore verticalizada, atribuindo novas etiquetas para os sintagmas.

O esquema de anotações se difere do usado no Penn TreeBank. São anotadas as etiquetas funcionais, seguindo o paradigma das gramáticas de restrições. Possui um conjunto de etiquetas com 14 classes principais de categorias de palavras em conjunto com 24 etiquetas para categorias de inflexão. Alguns exemplos de etiquetas de categoria de palavras são: N (Nouns), PROP (Proper Names), DET (Determiners)[8], V (Verbs). E os exemplos para as inflexões são: M (Male), P (Plural), ACC (accusative), PR (Present Tense).

[7] `http://visl.sdu.dk/`

[8] Em (BICK, 2000), página 69, o artigo é etiquetado como DET, porém no corpus utilizado neste trabalho esta classe de palavra é etiquetada como ART.

Por fim, o PALAVRAS também apresenta informações sobre funções sintáticas e forma sintática (estrutura de constituinte). Seus resultados, segundo o autor, são de 99% de precisão para análise morfológica (POS e inflexões) e cerca de 97% para funções sintáticas.

A seguir, na figura 3.7, apresenta-se um exemplo de uma sentença em português analisada por esse *parser* no formato de saída TigerXML.

```
<s id="s7" ref="7" source="Running text" forest="1" text="Os defeitos das
 esferas.">
   <graph root="s7_500">
      <terminals>
         <t id="s7_1" word="Os" lemma="o" pos="art" morph="MP" sem="--"
          extra="--"/>
         <t id="s7_2" word="defeitos" lemma="defeito" pos="n" morph="MP"
          sem="ac" extra="--" />
         <t id="s7_3" word="de" lemma="de" pos="prp" morph="--" sem="--"
          extra="sam-np-close" />
         <t id="s7_4" word="as" lemma="o" pos="art" morph="FP" sem="--"
          extra="-sam" />
         <t id="s7_5" word="esferas" lemma="esfera" pos="n" morph="FP"
          sem="Labs" extra="--"  />
         <t id="s7_6" word="." lemma="--" pos="pu" morph="--" sem="--"
          extra="--" />
      </terminals>
      <nonterminals>
         <nt id="s7_500" cat="S">
            <edge label="X" idref="s7_501" />
         </nt>
         <nt id="s7_501" cat="NP">
            <edge label="DN" idref="s7_1" />
            <edge label="H" idref="s7_2" />
            <edge label="DN" idref="s7_502" />
         </nt>
         <nt id="s7_502" cat="PP">
            <edge label="H" idref="s7_3" />
            <edge label="DP" idref="s7_503" />
         </nt>
         <nt id="s7_503" cat="NP">
            <edge label="DN" idref="s7_4" />
            <edge label="H" idref="s7_5" />
         </nt>
      </nonterminals>
   </graph>
</s>
```

Figura 3.7: Exemplo da sentença no formato TigerXML em português. Formato de saída do *parser* PALAVRAS

3.2.2 O *Parser* de Collins

Collins (1999), em seu modelo inicial, utiliza como base o método Cocke-Younger-Kasami (CYK), também conhecido como CKY, apoiado por uma gramática livre de contexto probabilística (GLC-P). A GLC-P é uma extensão da gramática livre de contexto em que cada regra gramatical possui uma probabilidade associada. As árvores sintáticas são desmembradas em suas regras constituintes e associa-se a probabilidade de ocorrência a cada regra da sentença observada. O modelo proposto por esta gramática supõe uma independência que considera a probabilidade de cada regra de sintagma de forma isolada aos demais sintagmas da sentença. A GLC-P possui algumas limitações, dentre elas o problema de gerar suposições fracas de independência e a falta de informação lexical.

Para aumentar a sensibilidade estrutural ou ao contexto, Collins introduz, em seu modelo, dependências lexicais entre bigramas, usando informações lexicais para modelar relações núcleo-modificador. Esta dependência é encontrada na relação entre as palavras na sentença reduzida, definindo um núcleo e um modificador. Uma sentença reduzida é formada a partir da sentença S inicial, sem as pontuações e com apenas os núcleos dos sintagmas nominais. Esta dependência é basicamente constituída por um par modificador-núcleo, a indicação da posição do modificador (se o modificador está à direita ou à esquerda em relação ao núcleo) e o pai desta subestrutura.

Para gerar o núcleo destas relações, Collins se baseia na teoria X-barra de Chomsky, onde projeta o núcleo do sintagma ao nó ascendente, de forma recursiva, até alcançar a raiz preenchendo assim o seu núcleo. Desta forma, cada núcleo é gerado antes de toda a estrutura dependente deste nó. Este é um modelo gerativo que tem como elemento principal do processo de geração o núcleo.

Também foi introduzido o conceito de distância, uma variável importante quando se decide a existência de relacionamento entre duas palavras. A distância é um vetor contendo duas informações: a adjacência entre os bigramas e a existência de um verbo entre eles.

Para o treinamento do *parser* de Collins foram usadas 40.000 sentenças do corpus *Wall Street journal* e para teste um conjunto de 2.416 sentenças.

A partir deste modelo inicial, Collins propõe mais três modelos, onde cada um estende o modelo anterior. Os melhores resultados foram obtidos nos modelos 2 e 3 alcançando aproximadamente 88,3% de precisão e 88,0% de cobertura.

A saída desse *parser* tem o formato Penn Treebank visto na figura 3.6.

3.2.3 O *Parser* de Jason

Outro *parser* que tem a finalidade de analisar sintaticamente estruturas em português é o *parser* desenvolvido por Wing e Baldridge (2006). Embora não seja utilizado neste trabalho, é citado por trabalhar com a implementação do modelo 2 de Collins (1999). Vale ressaltar, aqui, que os modelos propostos neste trabalho são independentes do *parser* usado para análise sintática. A única limitação quanto aos *parsers* é que seu formato de saída seja a codificação *TreeBank* TigerXML ou que possa ser convertido para este formato. A escolha em usar o *parser* PALAVRAS (BICK, 2000) se deu pelo fato da media harmônica apresentada no trabalho de Bick (2000) ser melhor que o resultado obtido por Wing e Baldridge (2006) além da disponibilidade da ferramenta.

Com o intuito de desenvolver um *parser* para a língua portuguesa, Wing e Baldridge (2006) realizaram algumas alterações nas configurações de parâmetros e mudança simples nos dados para adaptarem o *parser* implementado por Bikel (2004)[9]. Collins (1999) utilizou o corpus *Wall Street Journal*, constituído por sentenças na língua inglesa, enquanto Wing e Baldridge (2006) usaram o Floresta Sintática[10], um *corpus* da língua portuguesa, para treinamento da ferramenta.

Um pré-processamento no *corpus* foi necessário para converter o formato original da floresta sintática para o formato do Penn TreeBank. Alteração nas etiquetas de cláusulas conjuntivas e sinais de pontuação, adição de marcadores para explicitar o núcleo, distinção das clausulas relativas foram algumas das alterações necessárias.

A informação sobre o núcleo do constituinte na árvore é fundamental para derivar as relações de dependência e parametrizar o modelo de *parser*. Esta informação comumente ocorre no *corpus* Floresta Sintática, mas nem todos os sintagmas possuem esta informação. Para complementar estas informações, foram utilizadas regras heurísticas para inferir os núcleos dos sintagmas faltantes.

Dos três modelos de *parser* apresentados em (COLLINS, 1999), o modelo 2 foi escolhido para a implementação de Wing e Baldridge (2006), usando um conjunto de configurações específicas para a língua portuguesa (regras para definir o núcleo, características morfológicas, marcações de argumentos e algumas configurações do Floresta Sintática).

Na avaliação realizada pelos autores do *parser*, foi utilizado um conjunto de 5620 sentenças para treino e 1877 sentenças para teste. Seu melhor resultado obteve 63,2% de medida-F.

[9] Uma implementação do *parser* de Collins disponível em: `http://www.cis.upenn.edu/~dbikel/software.html`.

[10] `http://linguateca.dei.uc.pt/Floresta/`

3.2.4 A ferramenta TreeAligner

A última ferramenta descrita neste capítulo não realiza análise sintática automática como os *parsers* descritos anteriormente, mas possibilita a visualização das árvores geradas por eles. Os documentos nos formatos TigerXML e Penn TreeBank são documentos que possuem estrutura em forma de árvore, linearizada e de fácil processamento computacional, porém de difícil visualização. Para visualização gráfica das árvores usadas neste trabalho foi utilizada a ferramenta TreeAligner[11].

O TreeAligner é uma ferramenta para criação e busca em *treebanks* paralelos. Esta ferramenta é utilizada para anotações e criação de correspondências entre os nós nas árvores sintáticas paralelas. O TreeAligner permite realizar ligações entre os nós correspondentes em árvores sintáticas de diferentes idiomas. Estas ligações podem ser úteis para diversas aplicações na área de linguística, mais notoriamente em Tradução Automática.

O TreeAligner mostra graficamente árvores de arquivos no formato TigerXML e sua licença de uso é GNU GPL.

Para visualizar um documento em formato diferente do TigerXML é necessário, antes, fazer uso de uma ferramenta chamada TigerRegistry para a conversão de formatos, conforme já citado anteriormente.

Por meio de filtros de importação específicos para cada formato, o documento é indexado e convertido para o formato TigerXML. O filtro de conversão para o formato Penn Treebank trabalha com o *corpora* no estilo UPenn. Funções sintáticas são modeladas como *edge Labels*, e as arestas como *edges* secundários. Esse filtro foi testado com Penn Treebanks como o *Wall Street Journal*, *Penn Helsink Parsed Corpus of Middle English* e o *Chinese TreeBank*.

Exemplos de árvores sintáticas visualizadas pela ferramenta são apresentadas na figura 3.4, para os idiomas inglês e português do Brasil, respectivamente. Agora, na figura 3.8, ambas as árvores são apresentadas juntamente com os alinhamentos (correspondências), definidos manualmente com o auxílio da TreeAligner, para seus nós terminais e não terminais.

Como já mencionado anteriormente, o projeto de mestrado aqui apresentado foi desenvolvido com o objetivo de identificar um modelo de alinhador de árvores sintáticas paralelas e avançar nos estudos sobre o uso de informação sintática na tradução automática. Mais especificamente, com o intuito de investigar a construção de um sistema computacional para a produção de um recurso linguístico-computacional extremamente útil para várias técnicas de TA: *as árvores*

[11] `http://www.cl.uzh.ch/kitt/treealigner`

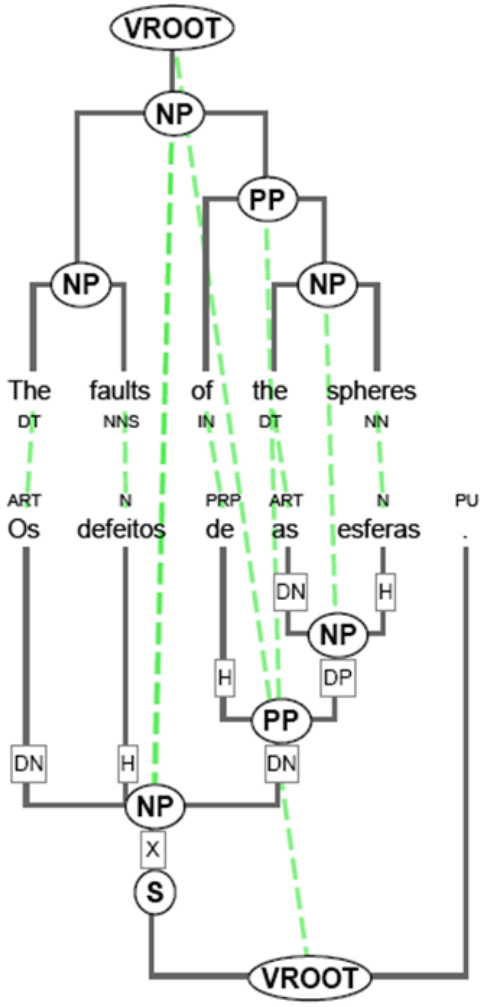


Figura 3.8: Exemplo das árvores em paralelo alinhadas por um especialista usando a ferramenta TreeAligner

sintáticas alinhadas.

No capítulo anterior, foram apresentados alguns métodos empregados para esta tarefa, assim como as métricas de avaliação e seus resultados. Também foram apresentadas as ferramentas e os recursos linguísticos necessários para desenvolver e avaliar o trabalho proposto. A partir de tudo o que foi exposto, neste mestrado foram propostos, implementados e avaliados alguns métodos híbridos de alinhamento de árvores sintáticas baseados, basicamente, nos trabalhos de Lavie et al. (2008) e Tinsley et al. (2007).

Enquanto Lavie et al., em seus estudos, utilizaram os idiomas inglês e chinês, Tinsley et al. trabalharam com os idiomas inglês e francês. Neste trabalho variações destes métodos foram aplicadas e avaliadas em *corpora* paralelos nos idiomas inglês e português do Brasil.

Para permitir o entendimento de todo o trabalho desenvolvido, na seção 4.1 são apresentadas as implementações baseadas nas propostas de Lavie et al. (2008) e Tinsley et al. (2007), assim como métodos híbridos que as combinam. Contudo, antes de citar como o processo de alinhamento de árvores sintáticas foi implementado, a seção 3.3 descreve o pré-processamento do *corpus* paralelo utilizado nos experimentos. Esse *corpus*, construído no projeto ReTraTos (CASELI, 2007), foi pré-processado para incluir informação proveniente da análise sintática. Nesse pré-processamento, além da informação sintática, outras informações relevantes para cada técnica de TA estão presentes como: formas superficiais, lemas, *part-of-speech*, etc.

O *corpus* na forma de árvores sintáticas (processado como descrito na seção 3.3) foi utilizado na avaliação dos métodos implementados (conforme descrito na seção 4.1), com base nas métricas padrão na área, obtendo os resultados apresentados no capítulo 5.

3.3 Pré-processamento do *corpus* português-inglês

Para o pré-processamento do *corpus* usado neste projeto foram utilizados analisadores sintáticos para os idiomas português do Brasil (pt) (BICK, 2000) e inglês (en) (COLLINS, 1999) já descritos no capítulo 2. O *corpus* pt-en usado nos experimentos desse projeto foi pré-processado por meio da análise sintática das sentenças fonte e alvo, separadamente. Deste *corpus* extraiu-se dois conjuntos: um para treinamento e outro para teste/referência. O *corpus* de *treinamento* foi usado em um estudo manual para extrair informações essenciais na fase de planejamento e desenvolvimento dos algoritmos de alinhamento. Outra parte desse *corpus* pré-processado foi separada para ser usada como *teste* e *referência* na avaliação dos métodos de alinhamento de árvores sintáticas como explicado na seção 3.3.1.

3.3.1 Os *corpora* de treinamento, teste e referência

Os métodos de alinhamento de árvores sintáticas implementados neste projeto foram avaliados em um *corpus* paralelo de textos escritos em português do Brasil (pt) traduzidos para o inglês (en). Esses textos são, originalmente, artigos da revista científica Pesquisa FAPESP[12] de nove seções diferentes: ciência, editorial, estratégia, humanidade, linha de produção, memória, opinião, política e tecnologia.

Para ser utilizado neste projeto, esse conjunto completo de textos paralelos passou por três etapas de pré-processamento descritas brevemente a seguir: alinhamento sentencial, análise sintática e alinhamento lexical.

Segundo Caseli (2007), o alinhamento sentencial de dois textos paralelos é o processo no qual são estabelecidas as correspondências entre as sentenças do texto fonte e as sentenças do texto alvo. O alinhamento de sentenças, do modo como foi gerado por Caseli (2007), foi utilizado neste projeto. Tal alinhamento foi obtido, primeiramente, por meio do alinhador automático TCAalign desenvolvido durante o projeto PESA[13], baseado no *Translation Corpus Aligner* (HOFLAND, 1996). Esse alinhador utiliza programação dinâmica para escolher o melhor alinhamento, usando critérios como iniciais maiusculas, caracteres especiais, tamanho da sentença, lista de palavras ancoras e palavras cognatas. Após o processo de alinhamento automático de sentenças uma verificação manual foi realizada para corrigir os erros de alinhamento ocorridos em casos menos prováveis (diferentes de 1 : 1).

A análise sintática das sentenças paralelas foi realizada de maneira independente de acordo com características de cada língua. Para tanto duas ferramentas foram usadas: o *parser* PALAVRAS (BICK, 2000) para etiquetar as sentenças em português e o *parser* do Collins (1999), para as sentenças em inglês.

Como já mencionado no capítulo 2, o *parser* PALAVRAS realiza a etiquetação morfológica, sintática e semântica, para os textos com o idioma português. Essa etiquetação é necessária para a extração das informações sintáticas e lexicais. Uma característica interessante desse *parser* é a de desmembrar as preposições em sua forma contraída, o que facilita o alinhamento de alguns *tokens* como a preposição "do", que na realidade está composta por "de" + "o". Desse modo, os *tokens* em inglês "of" e "the" podem ser relacionados com suas correspondências em português, por exemplo: o *token* "de" em português, com o *token* "of" em inglês, e o *token* "o" em português, com o *token* "the" em inglês. A saída do PALAVRAS está no formato TigerXML,

[12] URL da versão online da revista Pesquisa FAPESP: `http://revistapesquisa.fapesp.br`.
[13] `http://www.nilc.icmc.usp.br/nilc/projects/pesa.htm`

como já apresentado no capítulo 2.

O *parser* do Collins permite a etiquetação morfológica e sintática de textos na língua inglesa. Ele é baseado em métodos estatísticos e sua saída tem o formato Penn Treebank conforme apresentado na figura 3.6 do capítulo 2.

Por fim, o alinhamento lexical foi realizado por outra ferramenta automática: o GIZA++[14] (OCH; NEY, 2003). O GIZA++ utiliza modelos estatísticos (BROWN et al., 1993) e o modelo de Markov oculto (HMM) para determinar as correspondências mais prováveis entre palavras fonte e alvo. Neste projeto, o GIZA++ foi executado com sua configuração padrão e a união dos alinhamentos gerados separadamente nos sentidos fonte-alvo e alvo-fonte foi utilizada.

Após as três etapas de pré-processamento descritas anteriormente, o *corpus* final disponível para uso neste trabalho está composto por 16.994 pares de árvores sintáticas representando sentenças paralelas alinhadas lexicalmente. Desse conjunto, 108 pares de árvores foram separados para *teste*.

Esse conjunto de teste deu origem ao *corpus* de referência (*Gold Standard*) composto pelos mesmos 108 pares de árvores sintáticas de teste porém manualmente alinhadas. O alinhamento manual das árvores sintáticas paralelas foi realizado por um especialista da área de linguística e ocorreu tanto em nível lexical quanto sub-estrutural, sendo alinhados os nós terminais e os não terminais. Esse *corpus* de referência contém 3.273 nós terminais e 2.743 nós não terminais em inglês; e 3.115 nós terminais e 1.784 nós não terminais em português.

Para os nós terminais em inglês, 3.131 nós possuem ao menos um alinhamento com algum nó terminal na árvore em português, enquanto para os nós terminais em português, 2.849 nós terminais possuem ao menos um alinhamento com algum nó terminal na árvore em inglês.[15] Estes dados indicam a cobertura do alinhamento lexical realizado pelo especialista em linguística que foi de cerca de 96% para os nós terminais em inglês (3.131 de 3.273) e de 92% para os nós terminais em português (2.849 de 3.115).

Para os nós não terminais, a quantidade de nós com no mínimo um alinhamento foi bem menor do que a de terminais, uma vez que há bem mais nós não terminais em inglês do que em português (2.743 X 1.784). Assim, apenas 952 nós não terminais na árvore inglês e 1.032 nós não terminais na árvore português foram alinhados pelo especialista humano resultando em uma cobertura de cerca de 35% para os nós não terminais em inglês (952 de 2.743) e de 58% (1.032 de

[14] `http://code.google.com/p/giza-pp`

[15] Uma vez que tanto o alinhamento lexical quanto o alinhamento de nós das árvores sintáticas gerados pelo especialista humano podem envolver mais do que um nó fonte ou alvo, para o cálculo de cobertura aqui apresentado, considerou-se a ocorrência de pelo menos 1 alinhamento para cada nó.

1.784) em português.

Na geração manual desses alinhamentos, o especialista humano contou com o auxílio da ferramenta TreeAligner[16], descrita no capítulo 2. Apesar de bastante útil, essa ferramenta de alinhamento manual de árvores sintáticas possui algumas limitações; uma delas é que as árvores de entrada tenham o formato TigerXML. As árvores em português, geradas pelo *parser* PALAVRAS, já estavam nesse formato, porém com algumas discrepâncias que precisaram ser resolvidas para manter o formato de entrada exigido pelo TreeAligner. Assim, algumas etiquetas foram adicionadas no cabeçalho do documento XML (principalmente aquelas referentes às categorias gramaticais) para ser reconhecido como um documento TigerXML.

O texto em inglês, etiquetado pelo *parser* do Collins, precisou ser convertido do formato de Penn TreeBank para o formato TigerXML exigido pela ferramenta TreeAligner. Para essa conversão, foi usado o TigerRegistry[17], ferramenta que realiza a conversão de vários formatos como Penn TreeBank, Suzanne e NeGra para o formato TigerXML, conforme apresentado no capítulo 2.

Assim, com o auxílio da TreeAligner, o linguista especialista em ambos os idiomas alinhou manualmente as 108 árvores sintáticas paralelas dando origem ao *gold standard*. Mais especificamente, a partir da representação gráfica das árvores sintáticas paralelas (originalmente no formato TigerXML) apresentada pela ferramenta TreeAligner, o especialista as alinhava partindo dos nós terminais (folhas contendo as formas superficiais das palavras) alinhando, em seguida, os nós não terminais (que representam a estrutura sintática das árvores). Exemplos de textos com esse padrão, sendo um com os dados das árvores no idioma inglês e outro com as árvores no idioma em português, são apresentados nas figuras 3.3 e 3.7 respectivamente.

Uma melhor descrição da ferramenta TreeAligner, bem como do *parser* PALAVRAS, *parser* do Collins, TigerRegistry e os formatos usados por essas ferramentas pode ser revista no capítulo 2.

[16] `http://www.cl.uzh.ch/kitt/treealigner`

[17] `http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/doc/html/TIGERRegistry.html`

4 Alinhamento de Árvores Sintáticas

No contexto deste projeto de mestrado, foram implementados cinco modelos para alinhamento de árvores sintáticas. Para essa tarefa, foi usada a linguagem JAVA e o banco de dados MySql. A linguagem Java foi escolhida pelo fato de trabalhar com uma biblioteca desenvolvida especificamente para o formato de arquivo TigerXML, a Tiger-API. Esta API é um projeto *open source* e, assim como a linguagem Java e o banco de dados MySql, possui a licença GPL (*General Public License*). A ferramenta para desenvolvimento Java usada foi o NetBeans[1] na versão 6.7.1, o padrão de projeto utilizado foi o MVC (*model-view-control*), onde a camada de visualização foi feita usando *Swing*.

Este protótipo foi dividido em módulos por questão de organização facilitando, assim, a sua manutenção e até mesmo a fase de desenvolvimento. A estrutura deste protótipo está, portanto, dividida em 3 módulos:

- **módulo de entrada**: este módulo é o responsável pela leitura dos arquivos de entrada e armazenamento em uma base de dados;
- **módulo de alinhamento**: este módulo executa os algoritmos implementados realizando o alinhamento das árvores sintáticas previamente carregadas. Neste módulo são aplicados os critérios de alinhamento implementados para relacionar os nós entre a árvore fonte e a árvore alvo;
- **módulo de avaliação**: este módulo tem como tarefa avaliar a saída produzida pelo módulo de alinhamento. Neste módulo são aplicadas métricas como precisão, cobertura e medida-F, como descritas no capítulo5.

A figura 4.1 mostra a interface gráfica do módulo de alinhamento, onde é possível definir o tipo de alinhamento lexical, os recursos usados e o modelo a ser aplicado na tarefa de alinhar as árvores sintáticas.

[1] `http://netbeans.org/`

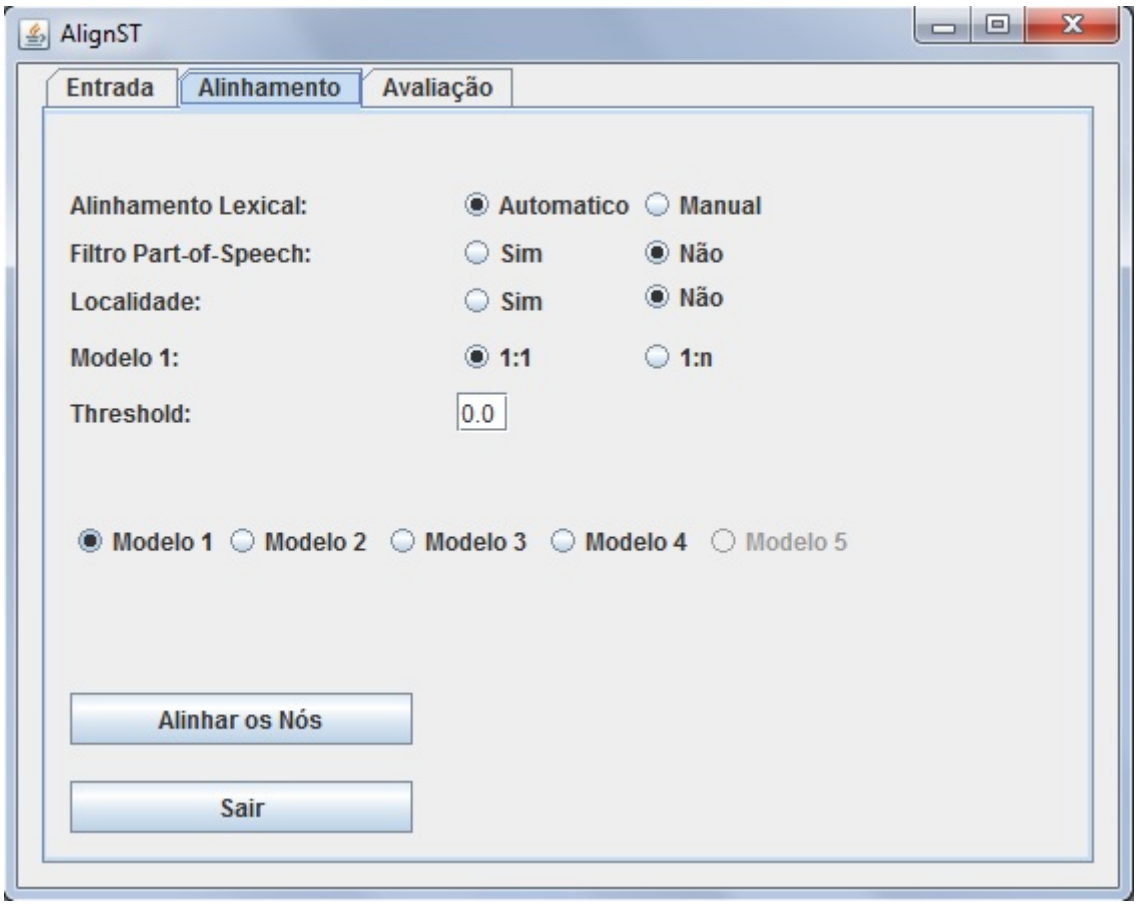


Figura 4.1: Módulo de Alinhamento

O módulo de entrada processa as informações contidas nos *corpora* após serem pré-processados como descrito na seção 3.3. Após processadas, estas informações são armazenadas na base de dados. Para auxiliar no processo de leitura do arquivo no formato TigerXML fornecido como entrada, a biblioteca TigerAPI foi usada.

Três arquivos de entrada são processados pelo módulo de entrada e usados pelos modelos de alinhamento: as árvores sintáticas fonte etiquetadas e no formato TigerXML, as árvores sintáticas alvo etiquetadas no formato TigerXML e o arquivo contendo o alinhamento lexical no formato XML.

Os nós não terminais alinhados pertencentes ao *Gold Standard* também são processados e armazenados pelo módulo de entrada. O arquivo *Gold Standard* está no formato XML, gerado pela ferramenta TreeAligner. Estas informações são usadas posteriormente pelo módulo de avaliação.

A TigerAPI facilita o acesso à estrutura de qualquer *corpus* no formato TigerXML para programadores Java. A API fornece métodos para percorrer toda a estrutura da árvore sintática e acessar elementos dentro de etiquetas como $< s > \ldots < /s >$, $< t > \ldots < /t >$, $< nt > \ldots < /nt >$.

A modelagem do banco de dados foi baseada na estrutura do documento TigerXML conforme apresentada na figura 4.2.

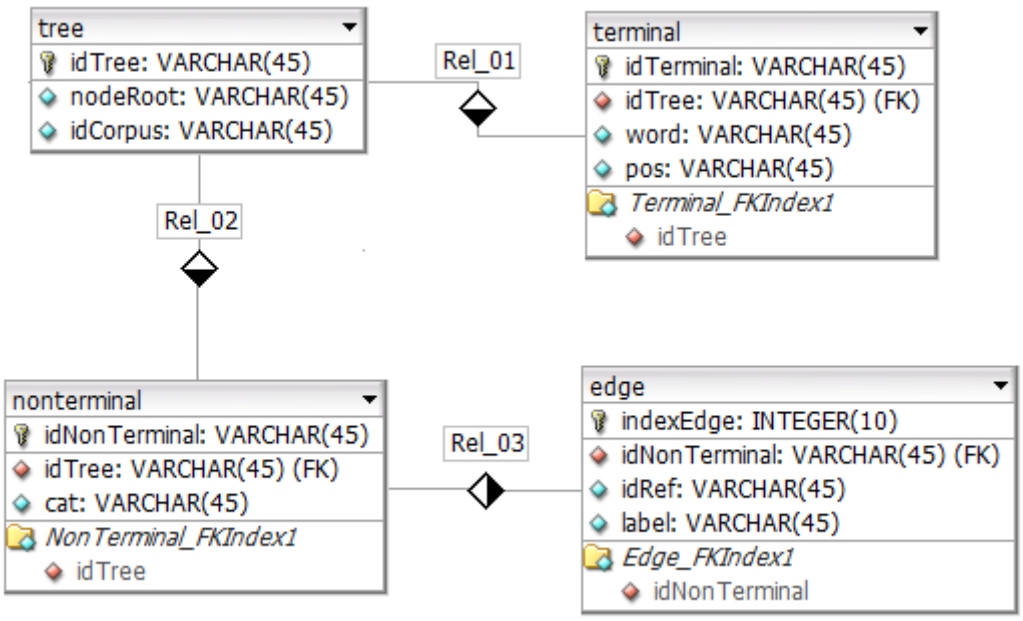


Figura 4.2: Modelagem do banco de dados na estrutura TigerXML

O campo `idTree` na tabela `tree` é referente ao atributo "id" e o `nodeRoot` referente ao atributo "root" na figura 4.3. O campo `idCorpus` é extraído no cabeçalho do documento TigerXML.

```
< s id="s7">
< graph root="s7_500">
```

Figura 4.3: Relação entre o TigerXML e a tabela `tree` no banco de dados

A tabela `terminal` contém os campos `idTerminal`, `word` e `pos` referentes aos atributos "id', "word" e "pos" respectivamente na figura 4.4. O campo `idTree` é proveniente do atributo "id" na tag $< s >$.

```
<terminals>
<t id="s7_1" word="The" pos="DT"/>
...
</terminals>
```

Figura 4.4: Relação entre o TigerXML e a tabela `terminal` no banco de dados

A estrutura da tabela `nonTerminal` está formada pelo campo `idNonTerminal` relativo ao atributo "id" e o campo `cat` relativo ao atributo "cat" na figura 4.5. Assim como na tabela `Terminal`, o campo `idTree` é proveniente do atributo "id" na etiqueta $< s >$.

```
<nonterminals>
<nt id="s7_501" cat="NP">
...
</nt>
</nonterminals>
```

Figura 4.5: Relação entre o TigerXML e a tabela `nonTerminal` no banco de dados

A construção da tabela `Edge` foi baseada nos atributos "idref" e "label', dando origem aos campos `idref` e `label`, respectivamente, como visto na figura 4.6. O campo `idNonTerminal` faz referência ao atributo "id" da figura 4.5. Um campo chamado `IndexEdge` é usado para criar um identificador único para cada registro na tabela `Edge`. Isto se dá devido ao fato de que a etiqueta $< edge >$ não possui um identificador próprio.

```
<nt id="s7_500" cat="NP">
<edge idref="s7_501" label="–"/>
...
</nt>
```

Figura 4.6: Relação entre o TigerXML e a tabela `Edge` no banco de dados

Para cada tabela criada no banco de dados, uma classe é gerada na linguagem Java. Um programa em Java pode ser considerado como uma coleção de objetos relacionados entre si por meio da invocação de métodos. Cada classe corresponde a um tipo de objeto. Neste módulo, quatro classes foram geradas sendo: Tree, Terminal, NonTerminal e Edge. Foi usada a especificação JPA (*Java Persistence API*) para realizar a persistência dos dados.

Após o armazenamento dos dados na base, o módulo de alinhamento utiliza estas informações como entrada para os algoritmos de alinhamento. Cada par de árvore alvo e fonte é processado de forma independente. Sendo assim, é instanciado um objeto para a árvore fonte e outro objeto para a árvore alvo, contendo informações referentes a sua estrutura e informações morfossintáticas. Os nós não terminais e nós terminais também são tratados como objetos. Estes objetos podem ser estendidos para conter outras informações, por exemplo, armazenar um número primo ou a probabilidade de alinhamento entre os nós lexicais, conforme as necessidades dos métodos de alinhamento implementados.

Cada método de alinhamento implementado neste projeto é uma classe em Java. Isto permite alterar um determinado método sem modificar ou danificar os métodos restantes. A saída destas classes são os pares de nós alinhados, que são armazenados na base de dados. Uma melhor descrição de cada método implementado pode ser vista na seção 4.1.

O módulo de avaliação, por sua vez, analisa os nós alinhados armazenados na base, apli-

cando as métricas de avaliação descritas no capítulo5. A saída do alinhador é comparada com os nós alinhados no *Gold Standard*. É importante destacar que esse módulo pode avaliar tanto o alinhamento de nós não terminais retornado pelo módulo de alinhamento, quanto o alinhamento de nós terminais alinhados pelo GIZA++ ou qualquer outro método de alinhamento lexical automático ou manual. Este módulo foi desenvolvido paralelo ao módulo de alinhamento.

Por fim, a saída do módulo de alinhamento é um arquivo XML no formato usado pela ferramenta TreeAligner, permitindo o seu uso em outros aplicativos.

```
<alignments>
     <align type="good" last_change="2011-03-24" author="STAlign">
        <node treebank_id="en" node_id="s93_501" />
        <node treebank_id="pt" node_id="s93_503" />
     </align>
     <align type="good" last_change="2011-03-24" author="STAlign">
        <node treebank_id="en" node_id="s93_502" />
        <node treebank_id="pt" node_id="s93_504" />
     </align>
</alignments>
```

Figura 4.7: Exemplo do formato de saída gerado pelo módulo de Avaliação

Na figura 4.7 é apresentado um trecho do arquivo de saída gerado pelo módulo de Alinhamento, mesmo formato XML usado pela ferramenta TreeAligner. Podemos notar no esquema de marcação os pares de nós alinhados por meio do elemento “node”. Os atributos deste elemento indicam o idioma e a identificação de cada nó terminal.

4.1 Implementação dos modelos de alinhamento de árvores sintáticas

Diante do estudo dos métodos relatados no capítulo 2, foram definidas algumas restrições e técnicas de alinhamento a serem aplicadas neste projeto. Assim, com base nos trabalhos citados anteriormente, esse trabalho adota como critérios de boa formação de um alinhamento de árvores sintáticas os seguintes:

- Os nós descendentes de uma língua fonte só podem ser ligados aos nós descendentes de suas contrapartes na língua alvo;
- Os nós ascendentes de uma língua fonte só podem ser ligados aos nós ascendentes de suas contrapartes na língua alvo;

- Os nós terminais só podem ser ligados aos nós terminais e os não terminais só podem ser ligados aos não terminais.

Estas restrições são adotadas para manter a estrutura da árvore alinhada, criando uma dependência entre os nós descendentes e ascendentes já relacionados, não permitindo a ligação de forma cruzada na estrutura da árvore.

Assim como Tinsley et al. (2007), Marecek et al. (2008), Tiedemann e Kotzé (2009), Menezes e Richardson (2001) e Groves et al. (2004), o processo de alinhamento é dividido em duas etapas: a primeira realiza o alinhamento dos nós terminais e, na segunda, alinham-se os nós restantes.

Para a primeira etapa, os nós terminais foram alinhados usando a ferramenta GIZA++ (OCH; NEY, 2003), assim como fizeram Lavie et al. (2008), Tinsley et al. (2007) e Tiedemann e Kotzé (2009).

Após obter o alinhamento dos nós terminais, a segunda etapa é alinhar os nós não terminais. Para esta tarefa, foram escolhidos dois modelos relatados no capítulo 2 para serem a base deste trabalho: o modelo que utiliza fatoração de números primos, usada por Lavie et al. (2008) e o modelo de Tinsley et al. (2007), que utiliza a probabilidade de alinhamento entre os nós lexicais. O principal motivo para a escolha destes dois modelos foi o fato de serem eles os de melhor custo-benefício quando considera-se o bom desempenho relatado com o uso de recursos relativamente simples, por exemplo, quando comparados aos recursos de Marecek et al. (2008).

Da mesma forma que Tiedemann e Kotzé (2009), o cruzamento de várias técnicas em busca de um melhor resultado foi realizado. Além dos modelos bases, foram aplicadas algumas variações e agrupamentos entre estes modelos, com o intuito de melhorar o desempenho na tarefa de alinhar as árvores sintáticas.

A estrutura da árvore analisada foi fragmentada, como nos modelos de Tinsley et al. (2007), Tiedemann e Kotzé (2009), Gildea (2003), Lavie et al. (2008) e Groves et al. (2004). Usando o conceito de Groves et al., no sentido de que as árvores não são necessariamente isomórficas, mas seus fragmentos podem ser. Cada nó não terminal da árvore é um fragmento, onde este nó é considerado raiz de uma subárvore.

Cinco modelos foram implementados no módulo de alinhamento, sendo os dois primeiros a base para o desenvolvimento dos demais. Cada um dos cinco modelos é descrito em uma subseção a seguir.

4.1.1 Modelo 1 – baseado no algoritmo de Lavie et al.

Seguindo uma ideia semelhante à descrita em Lavie et al. (2008), esta implementação atribui números primos para cada par de nós terminais alinhados previamente (por alguma ferramenta específica para esse fim ou com auxílio de um especialista humano). Além disso, atribui o valor 1 aos nós terminais sem alinhamento.

Para os nós terminais com alinhamentos múltiplos, o produto dos números primos de cada alinhamento é atribuído. Diferente do modelo original, o modelo implementado neste trabalho permite mais de um alinhamento para cada nó não terminal. Esta alteração foi adotada para que o modelo automático fosse capaz de lidar com os alinhamentos dos nós não terminais da mesma maneira que a especificada no *Gold Standard*, na qual a restrição de apenas um alinhamento para cada nó não foi seguida.

Este modelo tem como entrada um conjunto de árvores sintáticas paralelas com seus alinhamentos lexicais indicando a correspondência entre os nós terminais. A cada par de árvores sintáticas os nós não terminais são alinhados seguindo 3 passos como no modelo original. Primeiramente, cada par de nó terminal alinhado recebe um número primo, no segundo passo, os valores dos nós terminais são propagados para os nós não terminais, em uma abordagem *bottom-up*. O valor atribuído ao nó pai é o produto dos valores atribuídos a seus nós filhos. No último passo, o valor de cada nó na árvore fonte é comparado com os nós da árvore alvo. Caso estes valores sejam iguais então os nós não terminais em questão são alinhados.

Na figura 4.8 podemos ver duas árvores sintáticas alinhadas pelo modelo 1, note que o nó terminal “oldest” da árvore fonte está alinhado com os nós terminais “mais” e “antigo” da árvore alvo, o que indica mais de um alinhamento para o nó terminal “oldest”. Neste caso, podemos perceber que o produto entre os valores 13 e 17 ($13 \times 17 = 221$) é atribuído ao nó terminal da árvore fonte.

Uma outra configuração possível para este modelo é restringir a quantidade de nós que podem ser alinhados com cada nó não terminal. Assim como no modelo original, esta configuração permite apenas um alinhamento para cada nó não terminal. O modelo 1 usa a abordagem *bottom-up*, desta forma, é selecionado o nó alinhado mais próximo aos nós terminais, nos casos com mais de um alinhamento para este nó não terminal. Por exemplo, na figura 4.8, o nó fonte NP está alinhado com dois nós alvo NP e S. Com a restrição de alinhamentos 1:1, apenas o alinhamento entre o nó fonte NP e o nó alvo NP seriam mantidos. Desse modo, ganha-se em precisão porém perde-se em cobertura quando comparado ao alinhamento do *gold standard*.

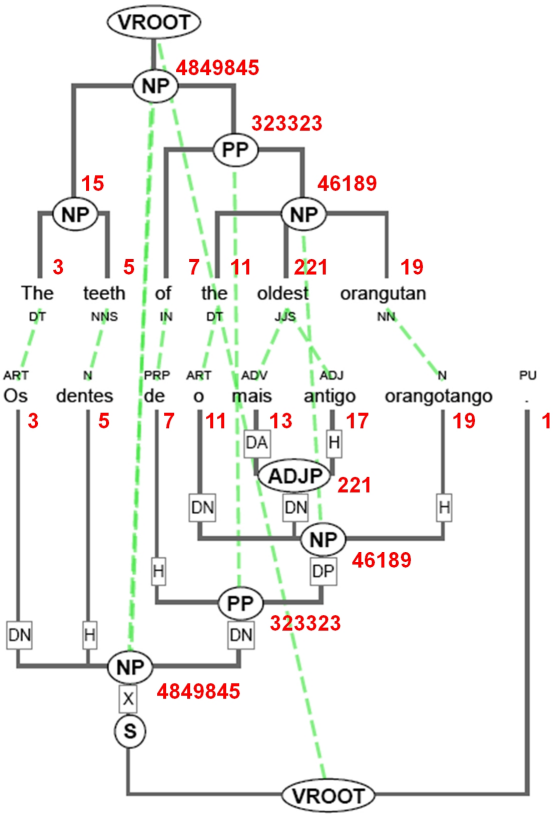


Figura 4.8: Exemplo de um par de árvores sintáticas paralelas alinhadas pelo modelo 1

4.1.2 Modelo 2 – baseado no algoritmo de Tinsley et al.

Semelhante ao método descrito em (TINSLEY et al., 2007), o modelo 2 utiliza a probabilidade gerada pelo GIZA++ (OCH; NEY, 2003) para analisar quais nós devem ser alinhados entre a árvore sintática fonte e a árvore sintática alvo. Nesta implementação, assim como no modelo original de Tinsley et al., cada nó não terminal da árvore só pode ser alinhado com um único nó na árvore oposta. Esta é uma das diferenças entre o modelo 1 e o modelo 2. Por trabalhar com a probabilidade, este modelo não permite alinhamentos multiplos, devido aos casos ambiguos possuirem a mesma probabilidade, o que não ocorre no modelo 1.

Para cada nó na árvore fonte, é calculada a probabilidade de alinhamento em relação a cada nó na árvore alvo. Estes valores são organizados em uma matriz e, a cada iteração, o par de nós com maior pontuação é alinhado. Quando dois pares de nós possuem a mesma pontuação, o modelo 2 segue uma abordagem gulosa selecionando o que está mais próximo da raiz seguindo a abordagem *top-down*.

Assim como no modelo 1, este modelo utiliza como entrada um conjunto de árvores sintáticas paralelas e o alinhamento lexical com as probabilidades gerada pelo GIZA++ (OCH;

NEY, 2003) para cada nó terminal alinhado.

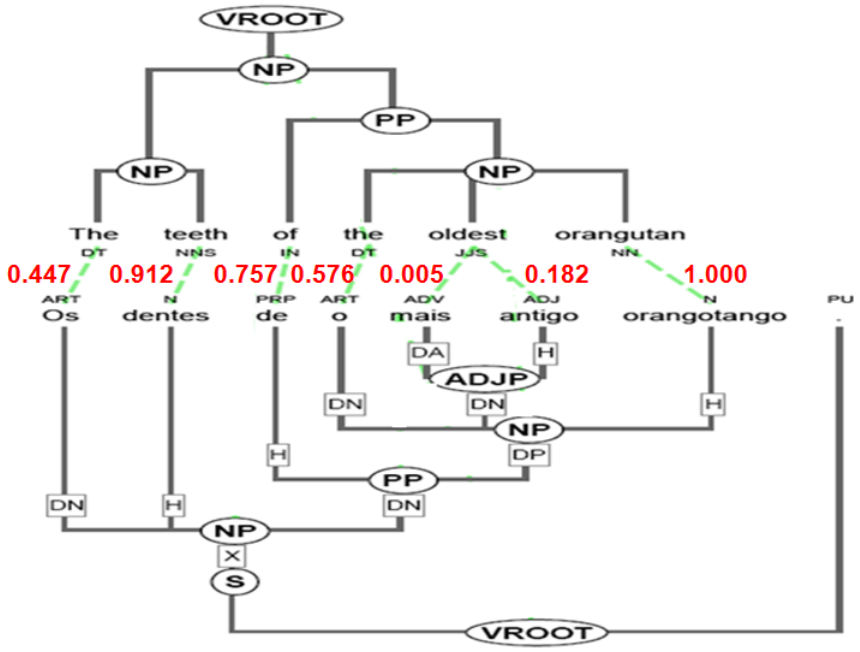


Figura 4.9: Probabilidades geradas pelo GIZA++ atribuidas a cada nó terminal alinhado.

Na figura 4.9 temos os valores das probabilidades geradas pelo GIZA++ para cada nó terminal alinhado. Após este passo, é calculada a pontuação a todos os possíveis pares de nós não terminais entre as árvores fonte e alvo.

Para gerar a pontuação, a seguinte fórmula é aplicada:

$$\gamma(\langle s,t\rangle) = \alpha(s_l|t_l) \bullet \alpha(t_l|s_l) \bullet \alpha(\bar{s_l}|\bar{t_l}) \bullet \alpha(\bar{t_l}|\bar{s_l})$$

Usando as sentenças da figura 4.9, considerando o nó não terminal NP da árvore fonte contendo os terminais "*the*", "*oldest*" e "*orangutan*" denotando s_l e o nó terminal NP da árvore alvo contendo os terminais "o", "mais", "antigo" e "orangotango" denotando t_l, temos:

- s_l = *the oldest orangutan*
- t_l = o mais antigo orangotando
- $\overline{s_l}$ = *The teeth of*
- $\overline{t_l}$ = Os dentes de

Aplicada a medida *score1* da figura 4.10 tendo x como s_l e y como t_l, obtemos:

$\alpha(s_l|t_l)$ = (0.576)*(0.005 + 0.182)* (1.000) = 0.1077

A probabilidade de alinhamento dos nós terminais é propagada para os nós não terminais usando a medida *score1* apresentada na figura 4.10. A medida *score1* apresentou melhor resultado que a medida *score2* no trabalho desenvolvido por Tinsley et al. (2007).

Score *score1*
$$\alpha(x|y) = \prod_{j}^{|y|} \sum_{i}^{|x|} P(x_i|y_j)$$

Score *score2*
$$\alpha(x|y) = \prod_{i}^{|x|} \frac{\sum_{j}^{|y|} P(x_i|y_j)}{|y|}$$

Figura 4.10: Cálculos aplicados por Tinsley et al. (2007) para gerar a pontuação do relacionamento entre os nós usando a probabilidade do GIZA++

Nesta equação, o nó não terminal da árvore fonte é representado por x e o nó não terminal da árvore alvo, por y; e nela calcula-se o produto da soma das probabilidades do alinhamento lexical dos nós terminais (x_i e y_i) contidos dentro dos fragmentos x e y.

Diferente do modelo 1, no qual o produto dos números primos é único, este modelo permite a mesma pontuação para dois pares de nós paralelos. Por esse motivo, várias iterações ocorrem, sempre alinhando o par de nós com maior pontuação na iteração, até que alguma condição de parada seja satisfeita. O número de iteração é variável de acordo com a quantidade de nós não terminais que a árvore possui em sua estrutura sintática. A classe nós não terminais, quando instanciada em Java, é extendida atribuindo a sua estrutura algumas propriedades, como por exemplo, a variável controle. Esta variável é importante na condição de parada. Quando a variável controle estiver setada com o valor 'bloqueado' em todos os nós não terminais, é satisfeita a condição de parada. O valor 'bloqueado' é atribuído à variável controle de cada nó não terminal no momento em que o nó é alinhado ou quando o resultado de $\gamma(\langle s,t \rangle)$ for zero.

Baseado no modelo 1 e modelo 2, três novos modelos foram implementados. Com base nas propriedades matemáticas dos conjuntos, foi possível usar variações como a união e a intersecção destes dois modelos base. Além disso, outra variação foi a combinação (*merge*) entre os modelos como explicado a seguir. A figura 4.11 mostra a entrada e a saída de cada novo modelo. Note que as entradas destes novos modelos são as saídas dos modelos 1 e 2.

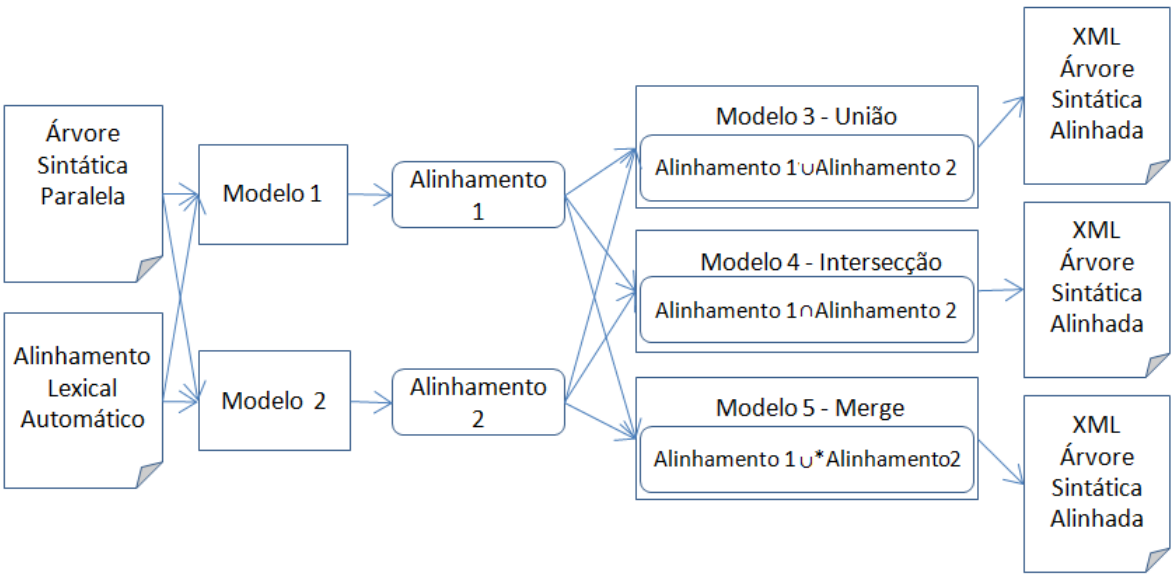


Figura 4.11: Entrada e saída dos Modelos implementados como variações dos modelos base 1 e 2

4.1.3 Modelo 3 – União entre os modelos 1 e 2

Após implementar os dois modelos, foi desenvolvida a união usando as saídas dos modelos 1 e 2 como entradas para o modelo 3. Assim como Tiedemann e Kotzé (2009) utilizam o algoritmo de Tinsley et al. (2007) como um dos recursos em seu modelo, com uma pequena alteração na equação de pontuação para cada nó não terminal, o modelo 3 utiliza o modelo 2 (descrito na seção 4.1.2) como um recurso juntamente com o modelo 1 (descrito na seção 4.1.1).

No conceito matemático, a união de dois conjuntos A e B, representada por $A \cup B$, é o conjunto dos elementos x, tais que x pertence a pelo menos um destes conjuntos A ou B:

$$x \in A \cup B \text{ se e somente se } x \in A \textbf{ ou } x \in B.$$

Sendo $A = (s_1,t_1);(s_1,t_2);(s_2,t_3);(s_4,t_5)$ e $B = (s_1,t_2);(s_2,t_3);(s_3,t_4)$ então $A \cup B = (s_1,t_1);(s_1,t_2);(s_2,t_3);(s_3,t_4);(s_4,t_5)$.

No contexto do alinhamento de árvores sintáticas, pode-se considerar que os conjuntos A e B representam os alinhamentos gerados pelos modelos 1 e 2, respectivamente. Além disso, os nós fonte são identificados como "s_i"e os nós alvo como "t_j". Assim, para entender melhor esse processo, a figura 4.12 ilustra o resultado dos modelos implementados como a combinação dos modelos base.

Nesta figura é possível perceber que a união dos alinhamentos gerados pelos modelos 1 e 2 são todos os nós alinhados em pelo menos um dos dois modelos base, eliminando-se os alinhamen-

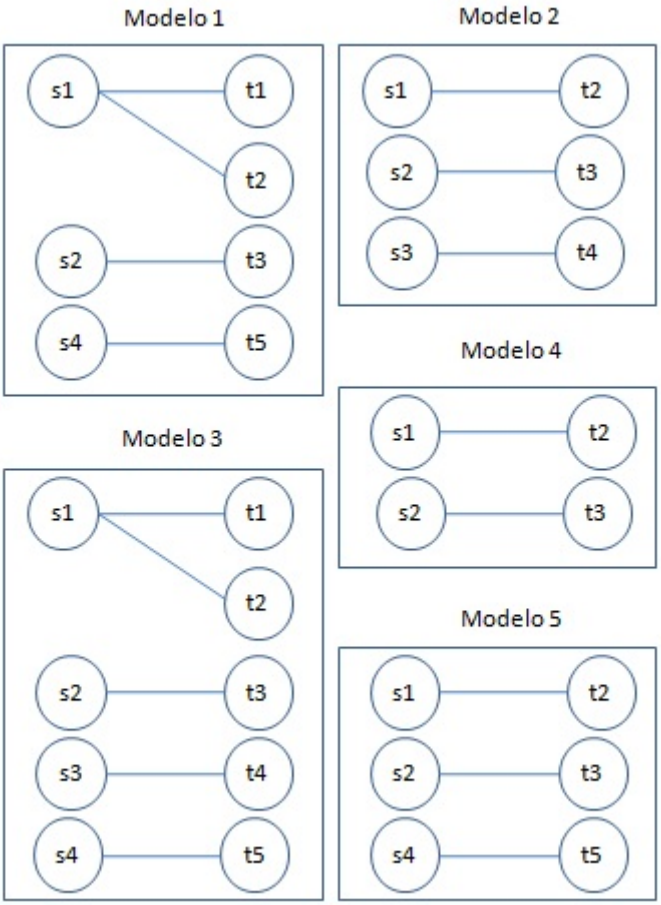


Figura 4.12: Ilustração da união (modelo 3), intersecção (modelo 4) e *merge* (modelo 5) dos alinhamentos dos modelos 1 e 2

tos redundantes. Note que o alinhamento dos nós s_3 e t_4 pertencentes ao conjunto de nós alinhados pelo modelo 2 é agregado aos nós alinhados pelo modelo 1. Como os alinhamentos $s_1 <=> t_2$ e $s_2 <=> t_3$ do modelo 2 também se encontram no modelo 1, estes não são agregados novamente no resultado gerado pelo modelo 3.

A união entre os modelo 1 e o modelo 2 foi desenvolvida com o intuito de melhorar a cobertura do processo de alinhamento das árvores sintáticas paralelas.

4.1.4 Modelo 4 – Intersecção entre os modelos 1 e 2

Este modelo implementa a intersecção entre os modelos 1 e 2. Assim como no modelo 3, a entrada para este modelo é dada pela saída dos dois modelos base (1 e 2). Novamente, da matemática sabe-se que a intersecção entre dois conjuntos A e B, denotada por $A \cap B$, é o conjunto dos elementos x tais que x pertence a ambos os conjuntos A e B.

$$x \in A \cap B \text{ se e somente se } x \in A \textbf{ e } x \in B.$$

Sendo $A = (s_1,t_1);(s_1,t_2);(s_2,t_3);(s_4,t_5)$ e $B = (s_1,t_2);(s_2,t_3);(s_3,t_4)$ então $A \cap B = (s_1,t_2);(s_2,t_3)$, como pode ser visto no modelo 4 da figura 4.12.

Desse modo, o conjunto de saída do modelo 4 está composto por apenas os nós não terminais alinhados em ambos os modelos 1 e 2, excluindo aqueles que foram alinhados por apenas um desses modelos. Na figura 4.12, a saída do modelo 4 contém apenas os alinhamentos $s_1 <=> t_2$ e $s_2 <=> t_3$, pois estes são os únicos que aparecem tanto na saída do modelo 1 quanto na saída do modelo 2.

A ideia de usar a intersecção entre os modelos base foi de melhorar a precisão do processo de alinhamento das árvores sintáticas paralelas.

4.1.5 Modelo 5 – *Merge* entre os modelos 1 e 2

Finalmente, o último modelo gerado é a combinação (*merge*) dos modelos 1 e 2 no qual aplica-se o modelo 2 para filtrar os alinhamentos múltiplos (com mais de um nó) gerados pelo modelo 1. Nesse filtro, apenas um dos nós do alinhamento múltiplo é escolhido e mantido na saída. O *merge* é denotado, neste trabalho, como $A \cup *B$ sendo A e B dois conjunto representando as saídas dos modelos 1 e 2, respectivamente. O resultado do modelo 5 é, portanto, o conjunto de elementos x tais que x pertence a A e, caso esteja envolvido em um alinhamento múltiplo (tenha mais de um alinhamento), x pertence a A e B.

$$x \in A \cup *B \text{ se } x_i \in A \text{ para } i = 1,$$
$$x \in A \cup *B \text{ se } x_i \in A \textbf{ e } x_i \in B \text{ para } i > 1,$$
$$\text{sendo } i \text{ o número de vezes que o nó } x \text{ é alinhado no conjunto } A.$$

Dado os conjuntos $A = (s_1,t_1);(s_1,t_2);(s_2,t_3);(s_4,t_5)$ e $B = (s_1,t_2);(s_2,t_3);(s_3,t_4)$ então $A \cup *B = (s_1,t_2);(s_2,t_3);(s_4,t_5)$, como é mostrado no modelo 5 da figura 4.12.

Veja que nesse exemplo da figura 4.12, o modelo 1 alinhou $s_1 <=> t_1$ e $s_1 <=> t_2$ (um alinhamento múltiplo de s_1 com dois nós: t_1 e t_2) e o modelo 5 eliminou o alinhamento $s_1 <=> t_1$ pelo fato de ele não ter sido alinhado pelo modelo 2, mantendo apenas o alinhamento $s_1 <=> t_2$. Os nós que possuem apenas um alinhamento não sofrem exclusão, permanecendo no conjunto de nós alinhados pelo modelo 5.

Este modelo tem a intenção de melhorar a precisão do modelo 1, assim como no modelo 4, ao mesmo tempo que tenta amenizar a diminuição da medida de cobertura. Em outras palavras, o

modelo 5 busca uma precisão tão boa quanto a do modelo 4 sem uma perda tão grande na cobertura. Vale dizer, também, que na implementação do modelo 5, o modelo 1 foi escolhido para receber este filtro por ter apresentado melhor resultado que o modelo 2.

5 Avaliação dos resultados

Na avaliação de forma intrínseca, o *corpus* de referência alinhado manualmente pelo especialista da área de linguística (*gold standard*), contendo 108 pares de árvores sintáticas alinhadas, foi comparado com a saída dos 5 modelos de alinhamento de árvores sintáticas descritos anteriormente, após processar, automaticamente, as mesmas árvores do *corpus* de referência.

Assim como Tiedemann e Kotzé (2009), o *gold standard* foi criado para conter duas categorias de alinhamento: os alinhamentos para os quais se tem certeza (*good*) e os alinhamentos para os quais não se tem tanta certeza (*fuzzy*). De acordo com o especialista da área de linguística, foram alinhados 3.137 nós terminais como "*good*" e 44 nós terminais alinhados como "*fuzzy*", enquanto para os nós não terminais foram alinhados 1.027 nós como "*good*" e apenas 2 nós não terminais como "*fuzzy*". Dada a pequena quantidade de nós não terminais alinhados como "*fuzzy*" estes foram considerados "*good*". Assim, nas equações a seguir, o conjunto G é formado pelos nós não terminais alinhados no *gold standard* independentemente de sua classificação como "*good*" ou "*fuzzy*" e o conjunto A são os nós alinhados automaticamente.

Três métricas foram usadas para avaliar intrinsecamente os nós alinhados, sendo precisão, cobertura e média harmônica (medida-F) apresentadas respectivamente nas equações 5.1, 5.2 e 5.3.

$$\text{Precisão} = \frac{|\, G \cap A \,|}{|\, A \,|} \tag{5.1}$$

$$\text{Cobertura} = \frac{|\, G \cap A \,|}{|\, G \,|} \tag{5.2}$$

$$\text{medida-F} = 2\frac{\text{Precisão} \times \text{Cobertura}}{\text{Precisão} + \text{Cobertura}} \tag{5.3}$$

A precisão é calculada como a porcentagem de alinhamentos corretos em relação a todos os alinhamentos gerados pelo método automático, enquanto a cobertura indica quantos alinhamentos do *gold standard* foram alinhados pelo módulo de alinhamento.

5.1 Avaliação dos alinhamentos de nós não terminais gerados pelos modelos 1-5

A avaliação descrita neste trabalho foi projetada para verificar o desempenho, e possivelmente a melhoria, dos modelos usados para o alinhamento das árvores sintáticas. Para tanto, cada modelo implementado foi avaliado com base no *corpus* de referência/teste descrito na seção 3.3 e usando o *corpus* de treinamento para gerar os alinhamentos lexicais automáticos dos nós terminais com o auxílio da ferramenta GIZA++ (OCH; NEY, 2003). O desempenho de cada modelo foi avaliado segundo as equações 5.1, 5.2 e 5.3.

A tabela 5.1 apresenta os resultados obtidos por cada um dos modelos. É importante destacar que nesta avaliação, o modelo baseado em Lavie permite mais de um alinhamento para cada nó não terminal.

Tabela 5.1: Valores de precisão, cobertura e medida-F dos 5 modelos implementados como descrito nas seção 4.1

	Precisão	Cobertura	Medida-F
Modelo 1	94,09%	82,63%	87,99%
Modelo 2	91,47%	76,96%	83,59%
Modelo 3	91,10%	**91,88%**	**91,49%**
Modelo 4	**95,22%**	67,71%	79,14%
Modelo 5	94,59%	72,62%	82,16%

A partir dos valores da tabela 5.1 é possível notar que, como esperado, o modelo 3 (união) foi o de melhor cobertura enquanto o modelo 4 (intersecção) foi o de melhor precisão. O modelo 3 também foi o que apresentou a melhor medida-F. Veja que o modelo 5 confirmou a hipótese de uma boa precisão (a segunda melhor precisão, perdendo apenas para o modelo 4) sem tanta perda na cobertura, como a do modelo 4 (o modelo 5 melhorou em 5% a cobertura do modelo 4).

Comparando somente o resultado individual de cada modelo base, podemos ver que o modelo 1 obteve melhor precisão e cobertura que os resultados apresentados para o modelo 2. Isto pode ocorrer pelo fato de o modelo 2 restringir o número de alinhamento a apenas um para cada nó não terminal. É bom lembrar que o *Gold Standard* permite mais de um alinhamento para cada nó não terminal.

A partir dessa primeira análise de desempenho dos modelos implementados, novos experimentos foram propostos e realizados com alterações nos modelos originais ou com a utilização de recursos adicionais no intuito de tentar melhorar os valores obtidos com as implementações

básicas. Assim, as próximas subseções relatam esses experimentos e seus resultados.

5.1.1 Restrição de alinhamentos para apenas 1 : 1

Para verificar o desempenho dos modelos permitindo apenas um alinhamento para cada nó não terminal, o modelo 1 foi alterado para seguir esta restrição. A tabela 5.2 mostra os resultados considerando-se, agora, a alteração no modelo 1. Os resultados do modelo 2 se mantiveram os mesmos, uma vez que sua implementação é completamente independente da implementação do modelo 1. Contudo, os modelos 3 e 4 foram influenciados já que o número de nós alinhados pelo modelo 1 decaiu. Veja que o modelo 5 não foi aplicado neste experimento uma vez que seu objetivo é filtrar os alinhamentos múltiplos gerados pelo modelo 1, os quais deixaram de existir na alteração avaliada aqui.

Tabela 5.2: Valores de precisão, cobertura e medida-F do modelo 1 restrito a alinhamentos 1 : 1 e seu impacto nos modelos 3 e 4

	Precisão	Cobertura	Medida-F
Modelo 1	96,84%	66,67%	78,97%
Modelo 2	91,47%	76,96%	83,59%
Modelo 3	91,81%	**87,91%**	**89,82%**
Modelo 4	**97,36%**	55,71%	70,87%

A partir da tabela 5.2 é possível notar que, como esperado, a precisão do modelo 1 subiu de 94,09% na versão original deste projeto para 96,84% na versão que restringe os alinhamentos a 1 : 1. A precisão do modelo 4 (intersecção) também melhorou passando de 95,22% quando o modelo 1 permitia alinhamentos 1 : n para 97,36% na implementação restrita a alinhamentos 1 : 1. Com a restrição de alinhamentos 1 : 1 imposta para o modelo 1, o impacto na cobertura e, consequentemente, na medida-F, foi sentido pelos modelos 1, 3 e 4 como é possível notar pela comparação desses valores nas tabelas 5.1 e 5.2.

5.1.2 Avaliação do alinhamento lexical (nós terminais)

Além de avaliar o alinhamento dos nós não terminais realizado pelos 5 modelos implementados, também avaliou-se a qualidade do alinhamento lexical gerado pela ferramenta GIZA++ (OCH; NEY, 2003) comparando-o com o alinhamento dos nós terminais presente no *gold standard* e, portanto, gerado manualmente.

As mesmas métricas de precisão, cobertura e medida-F foram utilizadas e o resultado pode ser visto na tabela 5.3.

Tabela 5.3: Avaliação do alinhamento lexical gerado por GIZA++ (OCH; NEY, 2003), união de ambos os sentidos de alinhamento: fonte-alvo e alvo-fonte

	Precisão	Cobertura	Medida-F
GIZA++ união	74,63%	93,42%	82,97%

Vale lembrar que nesse alinhamento, GIZA++ foi treinado com todo o *corpus* de treinamento e não apenas o pequeno *corpus* de teste usado nos experimentos. Além disso, os alinhamentos de GIZA++ considerados nesse trabalho e utilizados pelos cinco modelos, são resultantes da união dos alinhamentos de GIZA++ em ambas as direções de tradução: fonte-alvo (pt-en) e alvo-fonte (en-pt).

5.1.3 Avaliação do impacto da qualidade do alinhamento dos nós terminais no alinhamento dos nós não terminais: automático X manual

Após constatar a qualidade real dos alinhamentos automáticos, uma série de experimentos foram desenvolvidos para se avaliar o impacto da qualidade do alinhamento lexical no alinhamento dos nós não terminais. Uma primeira avaliação foi realizada para verificar esse impacto no modelo 1 comparando o alinhamento dos nós não terminais gerados com base no alinhamento manual e automático dos nós terminais. Para tanto, o modelo 1 recebeu como entrada as árvores sintáticas paralelas e o alinhamento lexical manual derivado do *gold standard*, ou seja, os nós terminas alinhados pelo especialista em linguística.

Os resultados, avaliando-se apenas o alinhamento dos nós não terminais, são apresentados na tabela 5.4. Nessa tabela são apresentadas as duas configurações do modelo 1, onde restringe-se apenas um alinhamento para cada nó não terminal ($1:1$) ou permite-se mais de um alinhamento para cada nó não terminal ($1:n$). Veja que em ambas as variações do modelo 1, o uso do alinhamento lexical manual gerou perda na precisão dos alinhamentos dos nós não terminais e ganho na cobertura. Isto ocorre porque uma melhor precisão no alinhamento lexical produz um número maior de alinhamentos de nós não terminais, aumentando assim a cobertura. Usando o alinhamento lexical manual, o modelo 1 alinhou 1049 nós não terminais e usando o alinhamento automático do GIZA++ alinhou 930 nós não terminais. Sendo assim, quanto maior a precisão do alinhamento lexical, maior a cobertura do alinhamento dos nós não terminais usando o modelo 1.

Uma análise de impacto semelhante não pôde ser feita com o modelo 2 porque o alinha-

Tabela 5.4: Avaliação do impacto da qualidade do alinhamento lexical dos nós terminais no alinhamento dos nós não terminais gerado pelo modelo 1

	Precisão	Cobertura	Medida-F
Modelo 1 1 : 1 (lexical manual)	96,12%	84,23%	89,78%
Modelo 1 1 : 1 (GIZA++ união)	96,84%	66,67%	78,97%
Modelo 1 1 : *n* (lexical manual)	93,33%	92,45%	92,89%
Modelo 1 1 : *n* (GIZA++ união)	94,09%	82,63%	87,99%

mento lexical manual não possui a probabilidade de alinhamento necessária para o algoritmo de alinhamento dos nós não terminais. No alinhamento automático, essa probabilidade é gerada por cálculos estatísticos realizados pelo GIZA++, o que não faz sentido no alinhamento lexical manual gerado pelo especialista humano. A impossibilidade de reprodução dessa análise ao modelo 2 se aplica também aos demais modelos que usam seu resultado como entrada: modelos 3, 4 e 5.

5.1.4 Avaliação do impacto da qualidade do alinhamento dos nós terminais no alinhamento dos nós não terminais: filtro de *part-of-speech*

Ainda com o propósito de analisar o impacto da qualidade do alinhamento lexical dos nós terminais no alinhamento dos nós não terminais, um novo experimento foi realizado para tentar melhorar a qualidade do alinhamento lexical automático. Nesse experimento foi proposto e implementado um filtro para restringir o alinhamento entre os nós terminais àqueles com categorias de *part-of-speech* pertencentes ao mesmo grupo de alinhamentos possíveis. Para tanto, o *corpus* de referência foi analisado para extrair informações de *part-of speech* dos nós terminais. Com a análise do alinhamento lexical do *Gold Standard* foram definidos grupos de etiquetas que permitem alinhamentos entre si.

Dessa análise surgiu um filtro de *part-of-speech* que verifica se as etiquetas de cada par de nós terminais alinhados automaticamente estão, ambas, dentro de um grupo de etiquetas possíveis. Caso as etiquetas pertençam a grupos diferentes, este alinhamento é excluído do conjunto de nós alinhados pelo GIZA++. Desse modo o filtro traz um aumento na precisão dos alinhamentos gerados por GIZA++.

A tabela 5.5 compara os valores do alinhamento de GIZA++ sem e com o filtro de *part-of-speech* aplicado como passo posterior.

Como é possível perceber pelos valores da tabela 5.5, o filtro de *part-of-speech* realmente melhorou a precisão do alinhamento lexical de GIZA++ ao passo que manteve sua cobertura inal-

Tabela 5.5: Avaliação do alinhamento lexical gerado por GIZA++ união sem e com o filtro de *part-of-speech*

	Precisão	Cobertura	Medida-F
GIZA++ união sem Filtro	74,63%	93,42%	82,97%
GIZA++ união com Filtro	**80,56%**	93,42%	**86,51%**

terada, comprovando que o filtro excluiu apenas nós alinhados erroneamente. A partir desse resultado, novos experimentos foram realizados para verificar o impacto do uso de um alinhamento lexical mais preciso (GIZA++ união com filtro de *part-of-speech*) no alinhamento dos nós não terminais.

Os 5 modelos de alinhamento de nós não terminais foram, então, avaliados usando o alinhamento lexical de GIZA++ união com filtro de *part-of-speech* e os resultados são mostrados na tabela 5.6

Tabela 5.6: Valores de precisão, cobertura e medida-F dos 5 modelos e alinhamento lexical de GIZA++ união com filtro de *part-of-speech*

	Precisão	Cobertura	Medida-F
Modelo 1	93,40%	84,14%	88,53%
Modelo 2	92,28%	76,77%	83,81%
Modelo 3	91,50%	**92,54%**	**92,02%**
Modelo 4	**94,76%**	68,37%	79,43%
Modelo 5	93,87%	73,75%	82,60%

Ao comparar os resultados apresentados sem o uso do filtro de *part-of-speech* (tabela 5.1) com os resultados obtidos usando tal filtro (tabela5.6) podemos notar que o modelo 1 diminuiu a precisão e melhorou a cobertura, uma vez que o alinhamento lexical com melhor precisão produz um maior número de nós não terminais alinhados, como visto na tabela 5.4. Os casos de nós terminais sem alinhamentos no modelo 1 são tratados de forma a amenizar o impacto aos nós não terminais (recebem o valor 1), enquanto o modelo 2 necessita da probabilidade desses alinhamentos. O alinhamento lexical com filtro de *part-of-speech* aumentou a quantidade de nós terminais sem alinhamento de 144 para 179 nas árvores em inglês e de 252 para 268 nós terminais nas árvores em português.

Assim como na avaliação dos modelos 1-5 usando o alinhamento lexical automático sem filtro, foi avaliado o desempenho usando o filtro de *part-of-speech* e permitindo apenas um alinhamento para cada nó não terminal (1 : 1) alinhado pelo modelo 1. O resultado de cada modelo pode ser visto na tabela 5.7.

Tabela 5.7: Valores de precisão, cobertura e medida-F do modelo 1 restrito a alinhamentos 1 : 1, usando alinhamento lexical de GIZA++ união com filtro de *part-of-speech*, e seu impacto nos modelos 3 e 4

	Precisão	Cobertura	Medida-F
Modelo 1	96,78%	68,18%	80,00%
Modelo 2	92,28%	76,77%	83,81%
Modelo 3	92,60%	**88,57%**	**90,54%**
Modelo 4	**97,23%**	56,37%	71,37%

Embora o modelo 4 tenha uma pequena queda na precisão comparado ao resultado demonstrado na tabela 5.2, melhorou a cobertura e medida-F. Como o modelo 1 obteve uma melhor cobertura e o modelo 2 uma melhor precisão, consequentemente o modelo 3 alcançou um aumento nas 3 medidas. Porém, todas as alterações nos valores das medidas obtidas com a aplicação do filtro de *part-of-speech* foram de menos de 2%, o que não pode ser considerada uma melhora significativa no alinhamento de nós não terminais.

5.1.5 Avaliação do impacto da qualidade do alinhamento dos nós terminais no alinhamento dos nós não terminais: recurso de Localidade

Nos recursos usados por Marecek et al. (2008), é evidente a relação com o alinhamento lexical, quando usado o prefixo dos nós terminais, a probabilidade de tradução do GIZA++, o uso de um dicionário bilíngue, similaridade do *part-of-speech* e similaridade da posição linear.

Tiedemann e Kotzé (2009) também usam a posição relativa de cada nó na árvore, considerando o nível de profundidade do nó e o tamanho da árvore.

Seguindo a mesma ideia desses autores, neste trabalho adotou-se um recurso que analisa a posição do nó terminal na árvore. Este recurso é chamado de Localidade. O recurso de Localidade consiste em calcular a posição de cada nó na árvore fonte e na árvore alvo, permitindo que o nó terminal seja alinhado com um outro nó terminal de acordo com uma determinada proximidade. Desse modo, o recurso de Localidade é aplicado a um conjunto de pares de nós terminais alinhados para excluir aqueles pares que não respeitam o critério de proximidade conforme especificado no algoritmo:

```
Inicialização
Para cada par de nó terminal(X,Y) alinhado faça
     posicao Xi = ((Xi * tamanhoArvoreY)/tamanhoArvoreX)
     limite = ((tamanhoArvoreY * 20)/100)
     Se Yj >= (posicaoXi − limite) e Yj <= (posicaoXi + limite) faça
          mantém (X,Y)
     senão
          exclui (X,Y)
     Fim do Se
Fim do Para
```

Note que no algoritmo é feita a normalização de acordo com o tamanho da árvore, onde X_i é a posição linear do nó lexical (posição relativa que o nó terminal ocupa na sentença) e o limite é referente a uma taxa de aproximação, neste caso o limite é de 20% de proximidade inferior ou superior em relação à posição do nó X_i. Esta taxa de aproximação foi escolhida após analisar valores superiores e inferiores a 20%, os quais obtiveram menor precisão que esta taxa.

Com o intuito de avaliar o desempenho do recurso de Localidade, foram calculadas as medidas de precisão, cobertura e medida-F aos nós lexicais e aos nós não terminais alinhados por cada um dos cinco modelos conforme apresentado nas tabelas 5.8 e 5.9, respectivamente.

Tabela 5.8: Avaliação do alinhamento lexical gerado por GIZA++ união com o recurso de Localidade.

	Precisão	Cobertura	Medida-F
GIZA++ união sem Localidade	74,63%	**93,42%**	82,97%
GIZA++ união com Localidade	**80,03%**	91,85%	**85,53%**

Ao aplicar o recurso de Localidade, o alinhamento lexical melhorou a precisão e medida-F, porém eliminou alguns nós alinhados corretamente, reduzindo a medida de cobertura, diferente do filtro de *part-of-speech* que eliminou apenas nós não terminais alinhados erroneamente. O resultado pode ser visto na tabela 5.8.

Já em relação ao alinhamento dos nós não terminais, como mostram os valores da tabela 5.9, assim como no filtro de *part-of-speech* (tabela 5.6), os modelos apresentaram melhora na medida de cobertura, com exceção do modelo 2 que melhorou a precisão. O mesmo ocorre usando o modelo 1 restrito a alinhamentos 1 : 1, conforme pode ser verificado na tabela 5.10. Novamente, o ganho nas medidas com o uso do recurso de Localidade foi de, no máximo, 2%.

Tabela 5.9: Valores de precisão, cobertura e medida-F dos 5 modelos e alinhamento lexical de GIZA++ união com o recurso de Localidade

	Precisão	Cobertura	Medida-F
Modelo 1	92,19%	84,70%	88,29%
Modelo 2	92,24%	76,30%	83,52%
Modelo 3	90,61%	**92,07%**	**91,33%**
Modelo 4	**94,44%**	68,93%	79,69%
Modelo 5	93,56%	74,13%	82,72%

Tabela 5.10: Valores de precisão, cobertura e medida-F do modelo 1 restrito a alinhamentos 1 : 1, usando alinhamento lexical de GIZA++ união com o recurso de Localidade

	Precisão	Cobertura	Medida-F
Modelo 1	95,47%	67,71%	79,23%
Modelo 2	92,24%	76,30%	83,52%
Modelo 3	92,00%	**88,01%**	**89,96%**
Modelo 4	**96,58%**	56,00%	70,89%

5.1.6 Avaliação do impacto da qualidade do alinhamento dos nós terminais no alinhamento dos nós não terminais: filtro de *part-of-speech* e recurso de Localidade

Como apresentado anteriormente, a qualidade do alinhamento lexical impacta na precisão e cobertura dos alinhamentos dos nós não terminais. Aplicando o filtro de *part-of-speech* é possível melhorar a medida de precisão do alinhamento lexical, assim como ao aplicar o recurso de Localidade.

Com o intuito de verificar o impacto de ambos os recursos como filtros do alinhamento lexical automático, primeiro foi aplicado o filtro de *part-of-speech* no conjunto de nós alinhados pelo GIZA++ e, posteriormente, o recurso de Localidade neste mesmo conjunto de nós lexicais. Foram selecionados apenas os nós terminais que possuem mais de um alinhamento ($1 : n$) para ser aplicado o recurso de Localidade com o objetivo de reduzir os casos de ambiguidade.

Na tabela 5.11 são apresentados os valores do alinhamento de GIZA++ somente com o filtro de *part-of-speech* e o alinhamento de GIZA++ com ambos os recursos: filtro de *part-of-speech* e Localidade.

Note que a limitação da taxa de aproximação (em 20%) eliminou alguns nós alinhados corretamente diminuindo a cobertura, porém, o número de nós incorretos eliminados foi maior aumentando, assim, a precisão. No geral, a média harmônica (medida-F) do alinhamento lexical melhorou de 86,51% para 88,27% sendo esta a melhor medida alcançada neste trabalho para o

Tabela 5.11: Avaliação do alinhamento lexical gerado por GIZA++ união com o filtro de *part-of-speech* e o recurso de Localidade

	Precisão	Cobertura	Medida-F
GIZA++ união com Filtro de *part-of-speech*	80,56%	**93,42%**	86,51%
GIZA++ união com Filtro de *part-of-speech* + Localidade	**84,91%**	91,91%	**88,27%**

alinhamento lexical automático.

Para o alinhamento dos nós não terminais, os resultados usando o alinhamento lexical GIZA++ união com filtro de *part-of-speech* e recurso de Localidade são apresentados na tabela 5.12.

Tabela 5.12: Valores de precisão, cobertura e medida-F dos 5 modelos e alinhamento lexical de GIZA++ união com filtro de *part-of-speech* e recurso de Localidade

	Precisão	Cobertura	Medida-F
Modelo 1	91,64%	86,97%	89,24%
Modelo 2	92,81%	76,77%	84,03%
Modelo 3	90,91%	**93,48%**	**92,18%**
Modelo 4	**93,94%**	70,25%	80,39%
Modelo 5	93,05%	75,83%	83,56%

A aplicação de ambos os recursos o número de nós alinhados aumentou em relação à aplicação de apenas um o que levou o modelo 3 a alcançar o melhor resultado em termos de cobertura neste trabalho, sendo também a melhor medida-F (tabela 5.12). Entretanto, a melhor precisão foi apresentada pelo modelo 4, permitindo apenas um alinhamento para cada nó (1 : 1), conforme visto na tabela 5.2.

Tabela 5.13: Valores de precisão, cobertura e medida-F do modelo 1 restrito a alinhamentos 1 : 1, usando alinhamento lexical de GIZA++ união com filtro de *part-of-speech* e recurso de Localidade

	Precisão	Cobertura	Medida-F
Modelo 1	95,59%	69,59%	80,54%
Modelo 2	92,81%	76,77%	84,03%
Modelo 3	92,74%	**89,24%**	**90,96%**
Modelo 4	**96,34%**	57,13%	71,73%

Considerando apenas os nós não terminais restritos a um alinhamento (1 : 1), a melhor medida-F é alcançada para cada modelo (1-4), como pode ser visto na tabela 5.13.

Tabela 5.14: Quantidade total de nós alinhados por cada modelo e a quantidade de nós corretamente alinhados

	GIZA++ união		Filtro POS		Localidade		POS+Localidade	
	Alinhado	Acerto	Alinhado	Acerto	Alinhado	Acerto	Alinhado	Acerto
Modelo 1 (1 : 1)	729	706	746	722	751	717	771	737
Modelo 1 (1 : *n*)	930	875	954	891	973	897	1005	921
Modelo 2	891	815	881	813	876	808	876	813
Modelo 3 (1 : 1)	1014	931	1013	938	1013	932	1019	945
Modelo 3 (1 : *n*)	1068	973	1071	980	1076	975	1089	990
Modelo 4 (1 : 1)	606	590	614	597	614	593	628	605
Modelo 4 (1 : *n*)	753	717	764	724	773	730	792	744
Modelo 5	813	769	832	781	839	785	863	803

A quantidade de nós alinhados por cada modelo, de acordo com o alinhamento lexical e os recursos aplicados, é apresentada na tabela 5.14, assim como a quantidade de acertos (nós alinhados corretamente). Observando os resultados apresentados, é possível notar que a precisão aumenta conforme a quantidade de nós alinhados diminuiu, enquanto a cobertura aumenta de forma inversa. Para a melhor precisão obtida pelo modelo 4 (tabela 5.2), o número de nós não terminais alinhados foi de 606, como mostra a tabela 5.14, e a melhor cobertura (tabela 5.12 teve 1.089 nós não terminais alinhados.

5.2 Regras extraídas a partir dos Alinhamentos

As árvores sintáticas paralelas, após alinhadas, podem gerar recursos capazes de auxiliar na tradução automática. Um desses recursos, as regras de composição, podem ser geradas a partir dos nós alinhados na árvore sintática.

Para extrair estas regras, neste trabalho foram usados os nós alinhados entre as árvores fonte (inglês) e as árvores alvo (português do Brasil) pelos modelos 3 (união) e 4 (intersecção). Neste caso, a direção escolhida é a tradução do inglês para o português do Brasil, mas não é vetada a direção inversa (português do Brasil-inglês). Assim, para cada par de nós não terminais alinhados por cada um dos modelos citados foi gerada uma regra de composição, resultando em dois conjuntos de regras: um para o modelo 3 e outro para o modelo 4. O primeiro conjunto de regras gerado a partir do alinhamento de melhor cobertura (93,48%), ou seja, o modelo 3 aplicando o filtro de *part-of-speech* e o recurso de Localidade ao alinhamento lexical do GIZA++ para a configuração 1 : *n*. A quantidade de regras de composição equivale ao número de nós alinhados por este modelo, sendo geradas 1.089 regras, como visto na tabela 5.14. O segundo conjunto de regras, por sua vez, foi gerado a partir do alinhamento de melhor precisão (97,36%): o modelo

4 somente com o alinhamento lexical do GIZA++, sem o filtro de *part-of-speech* e o recurso de Localidade, para a configuração 1 : 1. Foram geradas 606 regras, de acordo com o número de nós não terminais alinhados, visto na tabela 5.14. Vale destacar que cada regra pode ocorrer mais de uma vez em cada conjunto.

Cada regra é composta pelas derivações da árvore fonte e alvo. O processo de derivação consiste em buscar na árvore os nós (não terminais ou terminais) que se encontram um nível abaixo do nó não terminal alinhado. No exemplo das árvores sintáticas alinhadas (figura 4.8 na seção 4.1), os sintagmas preposicionais (PP) fonte e alvo estão alinhados, gerando a seguinte regra:

$$PP -- > \ in\,NP \mid PP -- > \ prp\,NP$$

O delimitador "|" é usado para separar a derivação extraída da árvore fonte à esquerda e a derivação extraída da árvore alvo à direita. O símbolo "$--\ >$" é usado para indicar os filhos derivados do nó não terminal alinhado.

Após extraídas as regras, foram calculadas suas probabilidades com base na frequência em que cada regra ocorre no determinado conjunto. A equação 5.4 demonstra como é calculada a probabilidade agregada a cada regra: conta-se a quantidade de vezes que o par (LE|LD) se repete no conjunto e divide-se pelo número de regras que contém o lado esquerdo (LE) desta regra.

$$\text{Probabilidade} = \frac{count(LE|LD)}{count(LE)} \tag{5.4}$$

onde LE (lado esquerdo) é a derivação da árvore fonte e LD (lado direito) é a derivação da árvore alvo.

Ao agregar a probabilidade, cada regra se torna única no conjunto diminuindo o número de regras de 1.089 para 552 no primeiro conjunto e de 606 para 229 no segundo conjunto de regras. Como esperado, o conjunto de 229 regras derivado do modelo 4 (intersecção) está contido no conjunto de 552 regras derivado do modelo 3 (união).

Comparando os dois conjuntos de regras e suas probabilidades, é possível notar que as regras geradas pelo segundo conjunto (intersecção) tem maior probabilidade, uma vez que o modelo 4 prioriza a precisão.

Na tabela 5.15 são apresentadas algumas regras extraídas usando o modelo 3 e a probabilidade de cada regra calculada de acordo com a equação 5.4. Juntamente com a probabilidade são apresentadas as frequências de LE e LD, separadas pelo símbolo /.

Tabela 5.15: Regras geradas pelo modelo 3 (união) e suas probabilidades

União	
Regra	Probabilidade
$NP -- > cd\ nn \mid NP -- > num\ n$	3/5 0,60
$NP -- > cd\ nn \mid NP -- > art\ n$	1/5 0,20
$NP -- > cd\ nn \mid PP -- > prp\ NP$	1/5 0,20

O modelo 4 gerou um número menor de regras, como se pode notar pelos valores na tabela 5.16, porém estas regras possuem uma probabilidade mais alta. Este conjunto (intersecção) eliminou duas regras, mas a regra de maior frequência na tabela 5.15 foi mantida com um leve aumento em sua probabilidade.

Tabela 5.16: Regras geradas pelo modelo 4 (intersecção) e suas probabilidades

Intersecção	
Regra	Probabilidade
$NP -- > cd\ nn \mid NP -- > num\ n$	2/3 0,66
$NP -- > cd\ nn \mid NP -- > art\ n$	1/3 0,33

Embora seja mais preciso o segundo conjunto, o modelo 4 penalizou a cobertura eliminando regras corretas com menor frequência, como a regra:

$$NP -- > dt\ \ jj\ \ jj\ \ nns \mid NP -- > art\ \ n\ \ adj$$

Para conhecer o desempenho destas regras é necessário aplicá-las em um modelo de tradução automática. Antes, porém, um exemplo de aplicação destas regras pode ser observado na figura 5.1 Para a árvore fonte (de entrada) em inglês "*The pressure of modern life*".

Dada a regra ($LE|LD$), as derivações LE e LD devem ser utilizadas de forma síncrona, ou seja, ao aplicar a derivação LE, a derivação equivalente LD deve ser aplicada. A derivação LE é extraída da árvore dada como entrada, neste caso a árvore fonte (inglês). Ao analisar todas as regras que possuem a derivação LE é escolhida a derivação LD presente na regra de maior probabilidade. Verificando passo a passo como a árvore alvo foi gerada tem-se:

1. O processo de produção da árvore alvo correspondente à árvore fonte inicia com o filho único do nó raiz VROOT. Desse modo, para todas as regras com LE = $NP -- > NP\ \ PP$ escolhe-se a de maior probabilidade:

Regra 1 $\quad NP -- > NP\ \ PP \mid NP -- > art\ \ n\ \ PP\ \ \ 41/110\ \ \ 0,37$

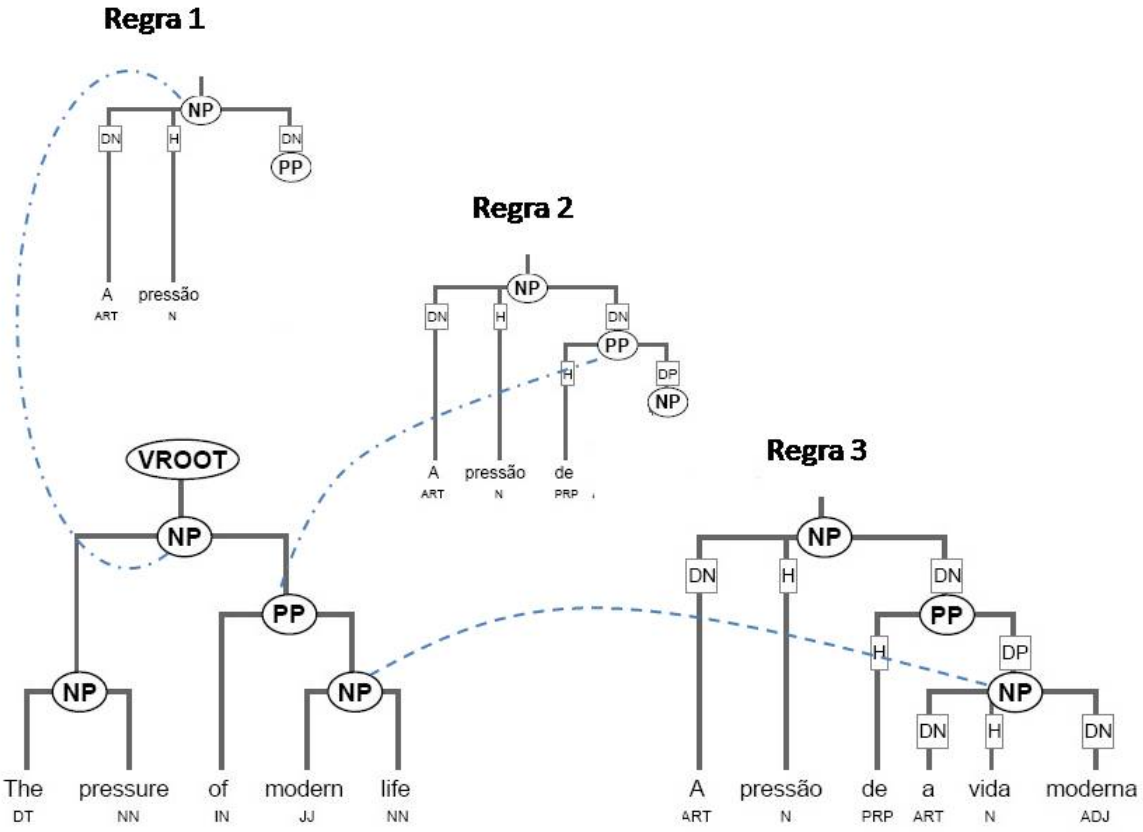


Figura 5.1: Regras de composição aplicadas para gerar a árvore alvo

2. Em seguida, o não terminal NP derivado no passo anterior na primeira posição mais à esquerda é examinado. Nesse momento, verifica-se que os filhos de NP alvo são equivalentes aos nós terminais já derivados fonte ($art|dt$ e $n|nn$) como especificado na regra:

$$\textbf{Regra 4} \quad NP --> dt\ nn \mid NP --> art\ n \quad 60/82 \quad 0,73$$

Nesse caso, a regra 4 é ignorada uma vez que está contida na regra 1. A derivação LD da regra 4 está implícita na derivação LD da regra 1, ou seja, possuem o mesmo nó pai (NP) e os nós filhos (art e n) se apresentam em ambas as derivações.

3. Em seguida, aplica-se a regra de maior probabilidade para o não terminal alvo PP:

$$\textbf{Regra 2} \quad PP --> in\ NP \mid PP --> prp\ NP \quad 164/215 \quad 0,76$$

4. Por fim, aplica-se a regra de maior probabilidade para o não terminal alvo NP:

$$\textbf{Regra 3} \quad NP --> jj\ nn \,|\, NP --> art\ n\ adj\ \ 3/8\ \ 0,37$$

Desta forma, a árvore alvo é formada de acordo com a árvore fonte. A figura 5.1 mostra a formação da árvore alvo conforme são aplicadas as regras de composição. As regras foram aplicadas sequencialmente, no sentido da raiz aos nós terminais. Foram escolhidas somente as regras de maior probabilidade.

É importante ressaltar que este exemplo de uso das regras foi aplicado a um modelo simples e pequeno de árvores sintáticas paralelas, retirado do *corpus* de treinamento. Apesar do processo de composição ter ocorrido com sucesso neste exemplo, não se pode afirmar que ele se aplica a todas as árvores sintáticas paralelas. Como visto anteriormente, a quantidade de nós não terminais que possuem ao menos um alinhamento é menor que o número total de nós não terminais, isto significa que as regras de composição extraídas podem não cobrir toda a gramática usada para composição das árvores. Para tanto, é necessário o estudo de modelos de tradutores automáticos que fazem uso de informações sintáticas, o qual foge ao escopo deste trabalho e é proposto como trabalho futuro.

6 Conclusões

Os sistemas de tradução automática que utilizam *corpus* na aquisição do conhecimento geralmente são limitados a domínios específicos, uma vez que, esses *corpora* precisam ser enriquecidos com um número maior de informações sintáticas e até semânticas para que bons resultados sejam alcançados. No entanto, a dificuldade de criação de tais recursos é justificada pela melhor qualidade de tradução nestes sistemas quando comparada a modelos simples que só utilizam a tradução de palavras individuais.

Neste contexto, um *corpus* de árvores sintáticas paralelas alinhadas é um recurso muito útil para melhorar a qualidade da tradução para sistemas de Tradução Automática baseados em transferência.

Diversas propostas para alinhar árvores sintáticas podem ser encontradas na literatura, mas para o português do Brasil não se tem conhecimento, até o momento, de nenhum trabalho. É importante destacar que dado o fato de ser uma área recente, as pesquisas sobre alinhamento de árvores sintáticas têm muito a serem exploradas.

Os resultados obtidos até o momento na comunidade científica têm revelado que a área é promissora, embora os métodos de avaliação aplicados sejam, em sua maioria, relacionados à precisão na fase de alinhamento (avaliação intrínseca) e poucos diretamente na aplicação de tradução (avaliação extrínseca).

A ferramenta desenvolvida neste trabalho possui uma flexibilidade para priorizar a precisão (modelo 4) ou a cobertura (modelo 3), o que é importante para uma futura avaliação extrínseca. Os resultados apresentados por este trabalho alcançaram 97,36% de precisão ao usar o modelo 4 e 93,48% de cobertura usando o modelo 3.

A ferramenta GIZA++ conseguiu alinhar os nós lexicais com uma precisão de 74,63% e cobertura de 93,42% usando um *corpus* com 16.994 pares de sentenças português-inglês. Porém, com a aplicação do filtro de *part-of-speech* e o recurso de Localidade a precisão melhorou mais de 10% chegando a 84,91% de precisão com uma perda de menos de 2% na cobertura que chegou a

91,91%.

Os trabalhos derivados desta pesquisa incluem aqueles nos quais as árvores sintáticas alinhadas serão usadas na tradução automática propriamente dita. Para isso, novos módulos para geração e aplicação de regras de tradução, seguindo a estratégia apresentada na seção 5.2, deverão ser implementados. Na extração dessas regras, os métodos de melhor precisão (modelo 4) e cobertura (modelo 3) serão utilizados para alinhar os 16.994 pares de árvores paralelas do *corpus* de treinamento. Em seguida, esses pares de árvores alinhadas serão fornecidos como entrada para o módulo de extração de regras. Por fim, as regras extraídas serão aplicadas na tradução automática de novas sentenças, de modo semelhante ao apresentado no exemplo da seção 5.2, e medidas de avaliação da qualidade da tradução como BLEU (PAPINENI et al., 2002) e NIST (DODDINGTON, 2002) serão utilizadas. A qualidade da tradução trará um indício da qualidade das regras e, consequentemente, do alinhamento de árvores sintáticas utilizado como entrada no processo de extração.

Referências Bibliográficas

BICK, E. The Parsing System “Palavras”: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. In: *PhD thesis - Aarhus University*. Aarhus, Denmark: [s.n.], 2000.

BIKEL, D. M. Intricacies of collins' parsing model. *Comput. Linguist.*, MIT Press, Cambridge, MA, USA, v. 30, p. 479–511, December 2004. ISSN 0891-2017. Disponível em: <http://dx.doi.org/10.1162/0891201042544929>.

BRANTS, T. TnT - A Statistical Part-of-Speech Tagger. In: *Proceedings of the 6th Applied Natural Language Processing Conference*. Seattle: [s.n.], 2000. p. 224–231.

BROWN, P. F.; PIETRA, S. A. D.; PIETRA, V. J. D.; MERCER, R. L. The mathematics of statistical machine translation: Parameter estimation. In: *Computational Linguistics*. [S.l.: s.n.], 1993. v. 19, n. 2, p. 263–311.

CASELI, H. M. *Indução de léxicos bilíngües e regras para a tradução automática*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação (ICMC), Universidade de São Paulo (USP), 2007. 158 p.

COLLINS, M. Headdriven statistical models for natural language parsing. In: *PhD thesis - University of Pennsylvania*. verificar: [s.n.], 1999.

DODDINGTON, G. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: *Proceedings of ARPA Workshop on Human Language Technology*. San Diego: [s.n.], 2002. p. 128–132.

DORR, B. J.; JORDAN, P. W.; BENOIT, J. W. A survey of current research in Machine Translation. In: *M. Zelkowitz (Ed.), Advances in Computers*. [S.l.: s.n.], 1999. p. 1–68.

GILDEA, D. Loosely tree-based alignment for machine translation. In: *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2003. p. 80–87.

GROVES, D.; HEARNE, M.; WAY, A. Robust Sub-Sentential Alignment of Phrase-Structure Trees. In: *Proceedings of the 20th International Conference on Computational Linguistics (COLING) 2004*. [S.l.: s.n.], 2004. p. 1072–1078.

HAJIC, J.; HAJICOVA, E.; PANEVOVA, J.; SGALL, P.; PAJAS, P.; STEPANEK, J.; HAVELKA, J.; MIKULOVA, M. Prague Dependency Treebank 2.0. In: *Linguistic Data Consortium, LDC Catalog No.: LDC2006T01*. Philadelphia: [s.n.], 2006.

HEARNE, M.; WAY, A. Seeing the Wood for the Trees: Data-Oriented Translation. In: *MT Summit IX*. New Orleans, LO: [s.n.], 2003. p. 165–172.

HEARNE, M.; WAY, A. Disambiguation Strategies for Data-Oriented Translation. In: *Proceedings of EAMT-2006*. Oslo, Norway: [s.n.], 2006. p. 59–68.

HOFLAND, K. A program for aligning English and Norwegian sentences. In: HOCKEY, S.; IDE, N.; PERISSINOTTO, G. (Ed.). *Research in Humanities Computing*. Oxford: Oxford University Press, 1996. p. 165–178.

KARLSSON, F. Constraint grammar as a framework for parsing running text. In: *COLING*. [S.l.: s.n.], 1990. p. 168–173.

KARLSSON, F.; VOUTILAINEN, A.; HEIKKILA, J.; ANTTILA, A. *Constraint Grammar, A Language-independent System for Parsing Unrestricted Text*. [S.l.]: Mouton de Gruyter, 1995.

LAVIE, A.; AGARWAL, A. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In: *Proceedings of the 2nd Workshop on Statistical Machine Translation*. Prague: [s.n.], 2007. p. 228–231.

LAVIE, A.; PARLIKAR, A.; AMBATI, V. Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In: *SSST '08: Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation*. Morristown, NJ, USA: Association for Computational Linguistics, 2008. p. 87–95.

MARECEK, D.; ZABOKRTSKY, Z.; NOVAK, V. Automatic Alignment of Czech and English Deep Syntactic Dependency Trees. In: *Proceedings of XII EAMT conference*. Hamburg, Germany: [s.n.], 2008.

MCDONALD, R.; PEREIRA, F.; RIBAROV, K.; HAJIC, J. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HTL/EMNLP)*. Vancouver, BC, Canada: [s.n.], 2005. p. 523–530.

MENEZES, A.; RICHARDSON, S. A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In: *Proceedings of the Workshop on Data-driven Machine Translation at ACL-2001*. Toulouse, France: [s.n.], 2001. p. 39–46.

OCH, F. J.; NEY, H. A systematic comparison of various statistical alignment models. *Computational Linguistics*, v. 29, n. 1, p. 19–51, 2003.

PAPINENI, K.; ROUKOS, S.; WARD, T.; ZHU, W. BLEU: a method for automatic evaluation of machine translation. In: *Proceedings of ACL-2002*. Philadelphia, PA: [s.n.], 2002. p. 311–318.

PENTHEROUDAKIS, J.; VANDERWENDE, L. Automatically identifying morphological relations in machine-readable dictionaries. In: *Ninth Annual conference of the University of Waterloo Center for the new OED and Text Research*. [S.l.: s.n.], 1993.

POUTSMA, A. Data-Oriented Translation. In: *Ninth Conference of Computational Linguistics in the Netherlands*. Leuven, Belgium: [s.n.], 1998.

POUTSMA, A. Data-Oriented Translation. In: *Proceedings of the 18th conference on Computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2000. p. 635–641.

POUTSMA, A. Machine translation with Tree-DOP. In: *Bod, R.; Scha, R.; Sima'an, K. (Eds.), (2003) Data-Oriented Parsing*. Stanford, CA: [s.n.], 2003. p. 339–359.

SAMUELSSON, Y.; VOLK, M. Alignment Tools for Parallel Treebanks. In: *Proceedings of GLDV Frühjahrstagung 2007*. Tübingen, Germany: [s.n.], 2007.

SPECIA, L.; RINO, L. H. M. *Introdução aos Métodos e Paradigmas de Tradução Automática*. Série de relatórios do NILC (NILC-TR-02-04), São Carlos-SP, 2002. 22 p. Disponível em: <http://www.nilc.icmc.usp.br/nilc/download/TR0204-SpeciaRino.zip>.

TIEDEMANN, J.; KOTZÉ, G. Building a large machine-aligned parallel treebank. In: *Proceedings of the 8th International Workshop on Treebanks and Linguistic Theories (TLT'08)*. Milão, Italy: [s.n.], 2009. p. 197–208.

TINSLEY, J.; ZHECHEV, V.; HEARNE, M.; WAY, A. Robust language pair-independent sub-tree alignment. In: *Proceedings of the MT Summit XI*. Copenhagen, Denmark: [s.n.], 2007. p. 467–474.

WING, B.; BALDRIDGE, J. Adaptation of data and models for probabilistic parsing of portuguese. In: *PROPOR*. [S.l.: s.n.], 2006. p. 140–149.

ŽABOKRTSKÝ, Z.; PTÁČEK, J.; PAJAS, P. Tectomt: highly modular mt system with tectogrammatics used as transfer layer. In: *StatMT '08: Proceedings of the Third Workshop on Statistical Machine Translation*. Morristown, NJ, USA: Association for Computational Linguistics, 2008. p. 167–170. ISBN 978-1-932432-09-1.