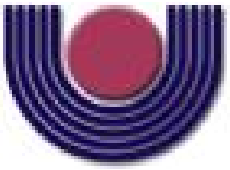


Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

Um Sistema de Simplificação Automática de Textos escritos em Inglês por meio de Transdução de Árvores

Gustavo Henrique Paetzold

CASCAVEL
2013

GUSTAVO HENRIQUE PAETZOLD

Um Sistema de Simplificação Automática de Textos escritos em Inglês por meio de Transdução de Árvores

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientador: Prof. Jorge Bidarra

CASCAVEL
2013

GUSTAVO HENRIQUE PAETZOLD

Um Sistema de Simplificação Automática de Textos escritos em Inglês por meio de Transdução de Árvores

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Jorge Bidarra (Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Lucia Specia (Co-Orientadora)
Department of Computer Science, The University of Sheffield

Prof. Clodis Boscarioli
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Márcio Seiji Oyamada
Colegiado de Ciência da Computação,
UNIOESTE

Cascavel, 19 de novembro de 2013

DEDICATÓRIA

Dedico este trabalho às pessoas que me forneceram a inspiração necessária para concluí-lo. À minha mãe Jane Shirlei Wilchen Paetzold, meu pai Waldemar Antonio Paetzold, e minha irmã Maira Gabriela Paetzold.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

CFG	*Context-Free Grammar*
GLC	Gramática Livre de Contexto
HMM	*Hidden Markov Model*
IBM	*International Business Machines Corporation*
PCFG	*Probabilistic Context-Free Grammar*
PLN	Processamento de Linguagem Natural
STSG	*Synchronous Tree Substitution Grammar*
T3	*Tree Transduction Toolkit*
BLEU	*Bilingual Evaluation Understudy*
NIST	*National Institute of Standards and Technology*

Sumário

Resumo

Enquanto a simplificação automática em nível lexical visa à substituição e remoção de palavras/termos complexos em sentenças, a simplificação automática em nível sintático tem o objetivo de reconstruir a estrutura sintática de uma sentença com alto grau de elaboração de forma a manter seu significado e aumentar sua legibilidade. O principal objetivo da simplificação automática de texto é tentar substituir palavras, expressões e estruturas sintáticas sentenciais de textos escritos, consideradas complexas ou de difícil compreensão, de modo facilitar a sua leitura. Neste trabalho, apresentamos e discutimos a especificação e implementação de um sistema de simplificação voltado para a simplificação de textos escritos em inglês por meio da transdução de árvores. Para tanto, experimentos com diferentes configurações do sistema sobre os dados da *Simple English Wikipedia* foram realizados. Os resultados obtidos até o momento vêm-se mostrando promissores, revelando, entretanto, a necessidade de técnicas mais elaboradas de produção de regras de simplificação e de métodos de ranqueamento de sentenças mais eficientes.

Palavras-chave: Processamento de Linguagem Natural, Simplificação Automática de Texto, Transdução de Árvores.

Capítulo 1

Introdução

É bastante comum nós, leitores, nos depararmos com textos de jornais, revistas e artigos científicos, nos quais encontramos palavras, expressões e estruturas sentenciais pouco conhecidas ou muito elaboradas, o que, muitas vezes, prejudicam a nossa compreensão sobre o texto lido. Exemplos de situações são, de um lado, as ocorrências nos textos de termos ambíguos ou para os quais não encontramos termos equivalentes na língua. De outro, o surgimento de períodos longos, muitas vezes entrecortados por orações internas, ligadas ou não por conectivos, cujo resultado final é a dificuldade de compreensão por parte de quem está lendo o texto. Estes tipos de documento constituem desafios não apenas para muitos leitores, mas também para sistemas computacionais de Processamento de Linguagem Natural (PLN) - Sistemas de Tradução Automática, Derivação Sintática e Sumarização.

A simplificação automática de texto surge como alternativa para contornar estes problemas. O objetivo principal da simplificação automática de texto é produzir versões simplificadas de sentenças complexas por meio de procedimentos computacionais. Investir no desenvolvimento e aprimoramento de estratégias de simplificação automática de texto pode, além de aumentar a qualidade dos resultados produzidos por diferentes sistemas computacionais de PLN, tornar acessível um grande conjunto de documentos que outrora trariam grandes desafios a indivíduos com dificuldade na leitura.

O desenvolvimento de sistemas de simplificação automática de texto comumente envolve não só profissionais da área da Ciência da Computação, mas também de diversas outas áreas do conhecimento. A perspectiva dos profissionais da área da Linguística, por exemplo, ajuda a esclarecer quais são as características intrínsecas de uma sentença complexa, permitindo assim que sejam projetadas e codificadas as mudanças que precisam ser feitas na estrutura da sen-

tença complexa para que a mesma seja simplificada. Já os profissionais da área da Psicologia, Medicina e Pediatria permitem que se compreenda com maior riqueza de detalhes quais são as principais dificuldades na leitura encontradas por portadores de patologias da linguagem, como a Dislexia e a Afasia.

A simplificação de uma dada sentença complexa pode ser feita de várias formas, por exemplo por meio de modificações em sua estrutura sintática ou lexical. Alguns dos efeitos que podem ser atingidos por meio de modificações na estrutura sintática de uma sentença complexa são: a segmentação de períodos muito longos em um número maior de períodos mais curtos e diretos, a transformação de trechos na voz passiva para a voz ativa e também a remoção de trechos que contenham conteúdo não-essencial ou irrelevante ao contexto geral de seu significado. Já modificações no conteúdo lexical de uma sentença permitem que sejam substituídas palavras/termos complexos de sua estrutura.

Com base nestas duas técnicas e também nos resultados apresentados pelos principais trabalhos publicados na área da simplificação de texto nas últimas duas décadas, propomos o desenvolvimento de um sistema que emprega uma estratégia de simplificação automática em nível sintático-lexical, que simplifica sentenças complexas pela aplicação de regras de simplificação confeccionadas automaticamente pelo processo de transdução de árvores[1]. O caráter sintático-lexical da estratégia de simplificação implica que serão empregadas regras de simplificação que modificam tanto a estrutura sintática, quanto a estrutura lexical de sentenças complexas.

Para discutirmos esse assunto e apresentarmos a solução proposta, esse documento assume a seguinte organização:

- **Capítulo 2:** Apresenta uma descrição detalhada com respeito às definições, motivações e estratégias referentes à simplificação automática de texto. Neste Capítulo são também descritas duas subcategorias de estratégias de simplificação: as em nível sintático, e as de nível lexical.

- **Capítulo 3:** Para auxiliar o leitor a compreender todos os conceitos e termos técnicos mencionados neste trabalho, o Capítulo 3 apresenta esclarecimentos com respeito aos

[1] A transdução de árvores é uma tarefa cujo objetivo é identificar o conjunto de procedimentos necessários para que se transforme uma estrutura de árvore fonte em uma estrutura de árvore alvo. Os detalhes desta tarefa são descritos na Seção 3.4

principais conceitos comumente empregados no desenvolvimento de estratégias de simplificação.

- **Capítulo 4:** Neste Capítulo são relatados os principais trabalhos publicados na área de simplificação automática de texto, e também de que forma eles se relacionam a este trabalho.

- **Capítulo 5:** Provê todos os detalhes com respeito a especificação, modelagem e programação do sistema de simplificação automática proposto. Este Capítulo tem o objetivo de guiar o leitor nos processos de idealização e construção do sistema que aplica a estratégia de simplificação proposta neste trabalho.

- **Capítulo 6:** Descreve quais foram os experimentos conduzidos com a estratégia de simplificação desenvolvida, e também provê análises qualitativas e quantitativas sobre os resultados obtidos.

- **Capítulo 7:** Apresenta as considerações finais com respeito às tarefas concluídas do presente trabalho, bem como uma discussão com respeito aos resultados obtidos e possíveis trabalhos futuros.

Capítulo 2

Simplificação Automática de Texto

Em [Wubben, Bosch e Krahmer 2012] a Simplificação Automática de Texto é definida como "o processo de produzir uma versão retrabalhada das sentenças/períodos que compõem os textos, por meio de alterações nas suas estruturas sintáticas e lexicais, com vista a tornar o texto mais claro, direto e objetivo para o leitor". Sentenças simplificadas são comumente reconhecidas por possuírem poucos casos de ambiguidade, lexical ou sintática, e também por serem estruturadas de uma forma que facilita a interpretação de seu conteúdo. O exemplo abaixo, extraído de [Chandrasekar e Srinivas 1997], ilustra uma sentença complexa em inglês cujos traços de linguagem são comumente encontrados em trechos de jornais, revistas e artigos: "*The embattled Major government survived a crucial vote on coal pits closure as its last-minute concessions curbed the extent of Tory revolt over an issue that generated unusual heat in the House of Commons and brought the miners to London streets.*"[1].

É fácil notar que a sentença de exemplo não é apenas longa, como também apresenta palavras, expressões e estruturas sintáticas mais elaboradas e, muitas vezes, de difícil compreensão por parte do leitor. Simplificando a sentença complexa anterior, se obtém a seguinte sentença: "*The Major government survived a crucial vote on coal pits closure. Its last-minute concessions reduced the Tory revolt over the coal-mine issue. This issue generated unusual heat in the House of Commons. It also brought the miners to London streets.*"[2]. Observa-se que a versão simplificada carrega o mesmo significado da versão complexa, porém possui uma estrutura gra-

[1]N.T.: A confrontada prefeitura sobreviveu a um voto crucial na conclusão do caso das minas de carvão, uma vez que as concessões de última hora comprimiram a extensão da revolta de Tory sobre um problema que gerou uma polêmica incomum na Câmara Parlamentar e levou os mineiros de Londres às ruas.

[2]N.T.: A prefeitura sobreviveu a um voto crucial na conclusão caso das minas de carvão. Concessões de última hora reduziram a revolta de Tory sobre o caso das minas de carvão. Este caso gerou uma polêmica incomum na Câmara Parlamentar. O caso também levou os mineiros de Londres às ruas.

matical que facilita sua compreensão pelo leitor. Essa transformação foi possível mediante a aplicação dos seguintes procedimentos:

1. **Segmentação de Sentença:** Para aumentar sua legibilidade, a sentença original foi segmentada em um conjunto equivalente de múltiplas sentenças mais curtas e diretas. Este processo tem como objetivo tornar mais explícitas as informações referentes a cada objeto da frase.

2. **Substituição de Termos Complexos:** Em textos escritos, é muito frequente o aparecimento de palavras/expressões difíceis, muitas vezes, desconhecidas pelo leitor. Uma das técnicas é justamente a substituição dessas palavras/expressões por equivalentes mais conhecidos e, portanto, mais claros para o leitor, ou, dependendo da importância delas no contexto, a sua própria remoção do texto. Retomando o exemplo mostrado na Tabela 1.2, a palavra "*embattled*" foi removida na simplificação da sentença complexa, enquanto o termo "*curbed the extent*" foi substituído por "*reduced*".

O trabalho de simplificação de textos pode tanto ser feito manualmente (pelas pessoas) ou por sistemas automatizados. Embora as simplificações feitas pelo homem tendam ser muito satisfatórias, é uma atividade de alto custo. Considere, por exemplo, a tarefa de simplificação manual de um livro de centenas de páginas. Seria necessária a contratação de dezenas de linguistas trabalhando por grandes períodos de tempo para se conseguir tal material. Além da demora e altos custos, pode não existir total concordância entre as simplificações de cada linguista, o que pode agregar ainda mais custos relacionados a revisão e até resimplificação total do material caso os resultados de simplificação manual iniciais sejam insatisfatórios.

Uma alternativa que, cada vez mais, vem ganhando espaço na sociedade, tem sido a simplificação automática de textos. O grande desafio para esse tipo de desenvolvimento, no entanto, é tentar reproduzir computacionalmente as operações de simplificação manual. A representação computacional destes procedimentos permite que a simplificação de texto seja aplicada de maneira mais ágil e menos custosa em textos extensos, artigos, jornais e outros.

2.1 Motivações

São muitas as razões pelas quais a simplificação automática de texto é uma tarefa a ser explorada ao máximo pelos pesquisadores da área da Ciência da Computação. Nas Seções seguintes serão discutidos os dois principais cenários nos quais a simplificação automática de texto tem sido uma técnica bastante empregada.

2.1.1 Simplificação Textual para Usuários Finais

O desenvolvimento de sistemas computacionais de simplificação, sejam eles assistivos ou não, vem ganhando um espaço de destaque nos trabalhos de PLN e Linguística Computacional nos últimos vinte anos. No avançado estágio da tecnologia dos meios de comunicação atual, a quantidade de informações disponibilizadas em linguagem natural tem sido bastante recorrente. Junto ao crescimento do volume deste tipo de conteúdo, cresce também a necessidade de tornar os aplicativos mais acessíveis para os seus usuários. A apresentação de textos mais simples para o usuário, embora não esteja restrita às pessoas com algum tipo de deficiência ou dificuldade de leitura, tem-se mostrado necessária e útil para esse segmento de pessoas. Dentre essas pessoas, citam-se, por exemplo, os chamados analfabetos funcionais [Watanabe et al. 2009] e também os afásicos.

A Afasia, diferentemente, do que ocorre com os analfabetos funcionais, é uma patologia adquirida, caracterizada por problemas sérios, tanto de fala quanto de escrita. Os afásicos tanto podem apresentar problemas com relação à estruturação sintática das sentenças, quanto no que diz respeito à nomeação [Prather et al. 1997]. Estatísticas documentadas em [Carroll et al. 1998] revelam que foram registrados cerca de um milhão de casos de Afasia na América e cerca de duzentos e cinquenta mil casos no Reino Unido, apenas no ano de 1998. Leitores afásicos normalmente contraem a condição devido a sérios problemas vasculares cardíacos ou traumas cranianos, e podem encontrar uma série de diferentes dificuldades quando tentam compreender um texto.

Entre os problemas mais comuns estão o recorrente esquecimento do significado de palavras incomuns, dificuldade na compreensão de sentenças com longos períodos, e nos casos mais graves, pode vir a causar total incapacidade de comunicação, seja ela escrita ou falada. Um exemplo de sentença que impõe desafios a um leitor afásico pode ser visto no parágrafo a

seguir, que consiste em uma única frase retirada de uma notícia do jornal The New York Times [Weisman 2013]: "*Despite new calls from the White House on Wednesday to enact a combination of tax increases and cuts to postpone the so-called sequester, the House is moving forward on a legislative agenda that assumes deep and arbitrary cuts to defense and domestic programs, once considered unthinkable, will remain in place through the end of the year.*"[3].

Apenas a título de curiosidade e no que diz respeito à situação brasileira no campo da leitura e compreensão, no trabalho publicado por [Watanabe et al. 2009] são mencionadas estatísticas divulgadas pelo Instituto Brasileiro de Geografia e Estatística (IBGE), indicando que no ano de 2007, cerca de 10% de toda a população brasileira com 15 anos de idade ou mais era analfabeta. No mesmo trabalho também são mencionados dados divulgados pela UNESCO também para o ano de 2007, estimando que cerca de 27% da população brasileira é considerada analfabeta funcional, uma categoria de analfabetismo que caracteriza pessoas com 4 anos ou menos de formação escolar em nível fundamental.

2.1.2 Simplificação Textual para Aplicações de PLN

Textos simplificados podem trazer benefícios não só a usuários que têm dificuldade com leitura, mas também a sistemas que fazem uso de material representado em linguagem natural. O trabalho produzido em [Chandrasekar, Doran e Srinivas 1996] menciona cinco diferentes áreas de Processamento de Linguagem Natural que podem ser beneficiadas se associadas à tarefa de simplificação de texto:

1. **Derivação Sintática:** Produzir árvores sintáticas de sentenças muito extensas pode ser um processo custoso para os sistemas de derivação. Alguns sistemas de derivação sintática produzem, pela utilização das gramáticas livres de contexto, múltiplas árvores sintáticas candidatas para uma dada sentença, e então selecionam qual a mais apropriada por meio de métricas probabilísticas. Sentenças complexas são comumente extensas, e isto faz com que sua representação sintática possa tomar múltiplas formas possíveis. Com o crescimento no número de representações sintáticas candidatas, aumenta também a dificuldade do processo de escolha da melhor candidata, e isto tende a gerar resultados de

[3]N.T.: Apesar das novas requisições da Casa Branca na quarta-feira para fosse legitimada a combinação entre aumento de impostos e cortes de verba no objetivo de adiar o chamado sequestre, a Casa dá continuidade a uma agenda que prevê cortes profundos e arbitrários sobre os programas domésticos e de defesa, outrora considerado impensável, continuará em vigor por todo o fim do ano.

derivação imprecisos. Como visto na parte introdutória desse Capítulo, sentenças simples tendem a ser fragmentadas em subsentenças mais curtas. Sentenças mais curtas levam os sistemas de derivação a produzirem uma quantidade menor árvores sintáticas candidatas, o que facilita a decisão da melhor candidata, e consequentemente aumenta a precisão dos resultados.

2. **Tradução Automática de Texto:** Existem várias técnicas distintas de Tradução Automática de Texto, e cada tipo de sistema pode se beneficiar da simplificação de texto de uma forma diferente. Tradutores estatísticos baseados em frases (ou "*phrase-based statistical machine translation systems*" em inglês) utilizam alinhamento textual como uma das etapas de seu processo de tradução. Sentenças mais curtas tendem a produzir dados mais fiéis de alinhamento textual, o que consequentemente aumenta a qualidade das traduções produzidas por este tipo de sistema de tradução. Um exemplo de categoria de sistema de tradução que se beneficia indiretamente da simplificação de texto é a dos sistemas de tradução em nível sintático. Esta categoria de tradutor depende de sistemas de derivação sintática para produzir dados sintáticos durante o processo de tradução. Como mencionado, sistemas de derivação sintática também se beneficiam do emprego da simplificação de texto, o que consequentemente traz benefícios para sistemas que os utilizem, incluindo os de tradução automática em nível sintático.

3. **Sumarização:** O processo de confecção de uma versão resumida de um determinado documento é intuitivamente relacionado à simplificação de texto. Muitas vezes o processo de sumarização consiste na produção de uma versão compactada de um documento, composta pelas sentenças mais relevantes ao seu conteúdo. Se o documento sendo sumarizado for um jornal, por exemplo, estas sentenças podem possuir longos períodos e termos incomuns. Nestes casos, a tarefa de simplificação de texto pode remover termos complexos e facilitar a legibilidade destas sentenças, aumentando assim a qualidade do sumário produzido.

4. **Indexação de Informação:** Sistemas de indexação de informação muitas vezes retornam grandes quantidades de documentos como resultado de uma busca feita por um usuário. Estes documentos nem sempre são todos relevantes para o usuário, o que provoca

a insatisfação do mesmo. Em ferramentas de busca de páginas na internet por exemplo, usuários tendem a utilizar palavras comumente empregadas em conversas do dia-a-dia para compôr as palavras-chave da busca. Portanto, substituir termos complexos de páginas da web por suas equivalentes simples no momento da indexação de páginas, pode levar o sistema de busca a retornar páginas mais condizentes com as palavras-chave sendo buscadas pelo usuário.

5. **Clareza Textual:** Certos tipos de material devem, por natureza, serem de fácil compreensão para qualquer tipo de usuário, a exemplo de manuais de montagem, guias turísticos e tutoriais em geral. Sistemas de simplificação podem ser empregados para garantir que as sentenças de materiais como estes não possuam sentenças excessivamente longas, ambíguas ou compostas por termos incomuns que possam confundir o leitor.

2.2 Métodos de Simplificação

Existem muitas estratégias diferentes para serem aplicadas ao processo de simplificação de textos. Dentre elas, citam-se a substituição individual das palavras complexas em uma sentença por sinônimos simples, e também estratégias mais sofisticadas que fazem mudanças diretamente na estrutura sintática da sentença complexa. Algumas das principais estratégias de simplificação automática empregadas nos trabalhos publicados nos últimos anos podem ser divididas em duas categorias: as estratégias de simplificação em nível lexical, e as estratégias de simplificação em nível sintático.

2.2.1 Simplificação Automática de Texto em Nível Lexical

A estratégia de simplificação em nível lexical é assim chamada por considerar apenas o conteúdo lexical de uma sentença complexa para confeccionar sua versão simplificada, desconsiderando seus aspectos gramatical e sintático. Esta estratégia pode tomar várias formas, mas em todos os casos ela se limita a transformar uma sentença complexa apenas em seu nível lexical, sem provocar alterações diretas às suas construções gramaticais e sintáticas.

Uma das formas de empregar a simplificação em nível lexical é por substituição das palavras complexas de uma sentença por seus sinônimos mais simples. Sistemas de simplificação que empregam esta técnica comumente buscam por sinônimos para palavras complexas em bancos

de dados linguísticos, como o Wordnet, e decidem qual o sinônimo mais simples com base em estatísticas de frequência do uso de palavras. Alguns exemplos do efeito produzido pela substituição de palavras complexas são encontrados na Tabela 2.1, que ilustra sentenças com termos complexos e seus equivalentes simples em negrito.

Sentença com Termos Complexos	**Sentença Simplificada**
Rome's current **ruler** fits the profile of a **sovereign**.	Rome's current **governor** fits the profile of a **king**.
Our **lecturer** uses teaching methods which are rather **unorthodox**.	Our **teacher** uses teaching methods which are very **uncommon**.
Simplified sentences tend to be **concise**.	Simplified sentences tend to be **short**.
The lion seems to be **enraged**.	The lion seems to be **furious**.
Hermeto Pascoal is a **renowed** musician.	Hermeto Pascoal is a **famous** musician.

Tabela 2.1: Efeito de simplificação em nível lexical por substituição de termos complexos

Estratégias de simplificação em nível lexical sofrem de certas limitações por não fazerem alterações diretas às construções sintáticas das sentenças complexas. Operações de simplificação como segmentação de sentenças e transferência de voz passiva para ativa não podem ser abordadas por estratégias desta natureza.

Em suma, apesar de suas limitações, sistemas de simplificação em nível lexical são ferramentas de grande utilidade. Entre suas principais aplicações estão a confecção de sentenças destinadas a leitores com problemas cognitivos [Carroll et al. 1998], e também a otimização do desempenho de sistemas de sumarização [Blake et al. 2007].

2.2.2 Simplificação Automática de Texto em Nível Sintático

Diferente da estratégia em nível lexical, a simplificação automática em nível sintático tem por objetivo obter a estrutura sintática de uma sentença complexa e então modificá-la de forma a criar uma versão simplificada de si com significado equivalente, porém com uma estrutura sintática que facilita a compreensão de seu conteúdo. Algumas das principais operações de simplificação em nível sintático seriam:

- **Segmentação de sentenças:** Consiste em transformar uma sentença com longos períodos em múltiplas sentenças mais concisas que expressam o mesmo significado. Sentenças mais curtas e diretas tendem a ser mais facilmente compreensíveis por grande parte dos leitores, especialmente aqueles que apresentam algum tipo de patologia de linguagem, como a Afasia ou Dislexia.

- **Transformação de sentenças escritas na voz passiva para a voz ativa:** [Carroll et al. 1998] afirma que leitores com Afasia têm dificuldade em compreender sentenças que não seguem a estrutura gramatical Sujeito-Verbo-Objeto. Sentenças redigidas na voz passiva comumente seguem estruturas gramaticais do tipo Objeto-Verbo-Sujeito, que podem trazer dificuldades de compreensão para leitores afásicos. A transferência de trechos na voz passiva para a voz ativa tem o objetivo de transformar sentenças de estrutura Objeto-Verbo-Sujeito em um trecho de sentença equivalente de estrutura Sujeito-Verbo-Objeto.

- **Resolução de referências anafóricas:** A anáfora é um fenômeno linguístico que consiste na retomada no texto de uma palavra ou de uma expressão mencionadas num contexto anterior que tanto podem ser via a substiuição por pronomes ou por outras expressões que sejam equivalentes aos referentes. Ela é comumente empregada para impedir que um mesmo termo ou expressão seja mencionado repetidas vezes num mesmo trecho. Um exemplo deste fenômeno é o uso do pronome pessoal "*She*" na frase "*Marie enjoys a good book. She prefers to read at night, before bed.*". A resolução de anáforas consiste na tarefa de encontrar os objetos sendo referenciados por cada anáfora na sentença complexa, e então reconstruir a sentença de forma a tornar mais explícitas as funções de seus sujeitos e objetos.

- **Remoção de Segmentos Não-essenciais:** Consiste em retirar de uma sentença complexa certos tipos de construções sintáticas que, na maioria dos casos, representam informações não-essenciais para seu significado. Na sentença "*John, a very talented musician, does not eat anything for breakfast*", por exemplo, é possível observar que a cláusula "*a very talented musician*" apresenta uma informação complementar ao sentido da sentença, e portanto poderia ser removida da sentença sem comprometer sua gramaticalidade ou

coerência.

A Tabela 2.2 ilustra exemplos do efeito de simplificação produzido pelas operações de segmentação, transferência de voz passiva para voz ativa, resolução de referências anafóricas e remoção de segmentos não-essenciais.

Método de Simplificação	**Sentença Complexa**	**Sentença Simplificada**
Segmentação de Sentenças	**George drank** as much as he wanted **and even danced** in the party.	**George drank** as much as he wanted in the party. **George also danced** in the party.
Transferência de Voz Passiva para Ativa	**The prisoner's** last meal was cooked by **a private chef** from Italy.	**A private chef** from Italy cooked **the prisoner's** last meal.
Resolução de Anáforas	The firefighter dragged the child out of the flames. **He** then became a hero.	The firefighter dragged the child out of the flames. **The firefighter** then became a hero.
Remoção de Segmentos Não-essenciais	My father, **once a vigorous rebel party supporter**, now rests in southern Russia.	My father now rests in southern Russia.

Tabela 2.2: Efeito de simplificação em nível sintático

Estratégias de simplificação em nível sintático são comumente empregadas em associação à simplificação em nível lexical para compor sistemas completos de simplificação. A associação entre as duas estratégias permite que tanto a estrutura sintática da sentença complexa seja modificada, como por exemplo por meio da segmentação ou remoção de segmentos, quanto sua estrutura lexical, por meio da substituição de termos complexos por equivalentes simples.

Sistemas de simplificação que empregam estas estratégias podem ser desenvolvidos de múltiplas formas. Comumente, estes empregam diferentes tarefas relacionadas à área de PLN para atingir o objetivo de transformar sentenças complexas em sentenças simples equivalentes. Para que seja possível compreender as estratégias empregadas pelos trabalhos correlatos descritos no Capítulo 4, é importante que sejam descritas as principais tarefas de PLN utilizadas pelos mesmos, e também como elas são utilizadas no projeto de sistemas de simplificação.

Capítulo 3

O Processo de Simplificação de Texto

Assim como apresentado no Capítulo 2, estratégias de simplificação automática de texto podem tomar uma variedade de formas distintas. Estas estratégias comumente se submetem a múltiplas outras tarefas da área de Processamento de Linguagem Natural, como a Derivação Sintática e o Alinhamento de Palavras.

A estratégia de simplificação em nível sintático, por exemplo, faz forte uso da tarefa de Derivação Sintática. Isto acontece pois, para esta estratégia de simplificação, é fundamental que se saibam quais as estruturas sintáticas das sentenças complexas para que se encontrem formas de simplificá-las. O trabalho de [Chandrasekar e Srinivas 1997] é um exemplo de estratégia de simplificação que emprega a Derivação Sintática como uma subtarefa de simplificação de texto.

Outro exemplo são as estratégias de simplificação em nível lexical, que podem ser beneficiadas pela tarefa de alinhamento de palavras. O alinhamento de palavras pode ser utilizado como um meio para produzir dicionários com sinônimos simples de palavras complexas. Estes dicionários podem então ser empregados na substituição de termos complexos nas sentenças a serem simplificadas.O trabalho descrito em [Bott e Saggion 2011] apresenta uma forma de aplicação do alinhamento textual na extração de relações entre termos complexos e simples. Nele, é proposto um método de extração não-supervisionado que cataloga correspondências entre termos complexos e simples na língua espanhola.

Na sequência, serão tratados mais detalhamente algumas das principais tarefas de PLN que são empregadas neste processo.

3.1 A Tarefa de Alinhamento de Palavras

O Alinhamento de Palavras consiste em estabelecer relações de tradução entre as palavras de uma sentença fonte e as palavras de uma sentença alvo. Essas sentenças tanto podem se referir a uma única língua ou a duas línguas distintas. Embora nesse trabalho estejamos tratando de transformações de sentençcas escritas em inglês para o prõprio inglês, para tentar facilitar a compreensão do que seja o Alinhamento, faremos uso de duas línguas distintas cujos trechos foram submetidos ao processo tradutório. No exemplo da Figura 3.1, extraída de [Specia 2012], a língua fonte é o inglês e a língua alvo, o francês.

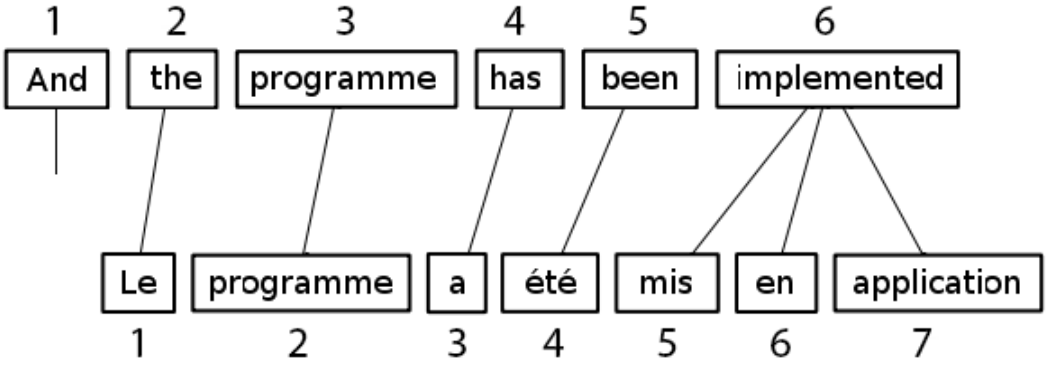


Figura 3.1: Alinhamento de palavras entre duas sentenças equivalentes, uma escrita em inglês e a outra, uma tradução para o francês

Para o Alinhamento de Palavras encontram-se disponíveis vários sistemas, dentre as quais citamos o GIZA++ e o Meteor. O GIZA++ [Och e Ney 2003] é um sistema baseado em cálculos probabilísticos que utiliza a aprendizagem de máquina não-supervisionada [Hinton e Sejnowski 1999] na implementação dos modelos IBM de 1 a 5 [Brown et al. 1993], o modelo de Hidden Markov (HMM) e também outros algoritmos de otimização. Estes modelos são caracterizados por algoritmos baseados em princípios matemáticos, e sua função é produzir correspondências entre as palavras que compõe duas línguas.

O modelo IBM 1 utiliza o algoritmo da maximização de expectativa [Osborne 2013], cuja função é encontrar correspondências entre os componentes lexicais de duas linguagens com base em um corpus paralelo de sentenças. O corpus paralelo consiste em um arquivo que alinha sentenças escritas na língua a ser traduzida a sentenças já traduzidas na língua alvo. A Tabela 3.1 ilustra um exemplo de corpus paralelo deste tipo, produzido em [Koehn 2005].

Resumption of the session.	Reprise de la session.
Please rise, then, for this minute's silence.	Je vous invite à vous lever pour cette minute de silence.
Madam President, on a point of order.	Madame la Présidente, c'est une motion de procédure.
If the House agrees, I shall do as Mr Evans has suggested.	Si l'Assemblée en est d'accord, je ferai comme M. Evans l'a suggéré.
Madam President, on a point of order.	Madame la Présidente, c'est une motion de procédure.

Tabela 3.1: Corpus paralelo entre as línguas inglesa e francesa

O algoritmo de maximização de expectativa do modelo IBM 1 encontra estas correspondências por meio de um processo iterativo que envolve cálculos probabilísticos. Primeiramente o algoritmo assume que as correspondências entre todas as palavras das duas linguagens têm a mesma probabilidade. Em seguida, o algoritmo lê cada par de sentenças do corpus paralelo e atualiza as probabilidades das correspondências: as probabilidades da correspondências entre as palavras da sentença na língua fonte e as palavras da sentença na língua alvo são incrementadas. O processo de leitura de cada par de sentenças do corpus paralelo é repetido até que as probabilidades das correspondências converjam.

Ao terminar, o model IBM 1 produz os dados de alinhamento textual para cada par de sentenças do corpus paralelo, assim como na Figura 3.1, e também um dicionário probabilístico de formato similar à Tabela 3.2, contendo as probabilidades de correspondência entre as palavras das duas linguagens do corpus paralelo.

Palavra em Inglês	**Palavra em Francês**	**Probabilidade de Correspondência**
resumption	reprise	0.0285714
the	la	0.0066225
the	du	0.0012119
silence	silence	0.0476190

Tabela 3.2: Modelo de Tabela produzida pelo modelo de tradução IBM 1

Os modelos IBM 2, 3, 4 e 5 são variações do modelo IBM 1, e empregam técnicas de otimização para aumentar a qualidade das correspondências produzidas. É importante esclarecer que cada modelo IBM é uma otimização da versão anterior, ou seja, o modelo IBM 2 acrescenta melhoras ao modelo IBM 1, o modelo IBM 3 acrescenta melhores ao modelo IBM 2, e assim sucessivamente. O modelo IBM 2, por exemplo, complementa o processo de estimação das correspondências por meio de um modelo de distorção. No modelo de distorção, a posição das palavras nas sentenças fonte e alvo são levadas em consideração quando estimadas as probabilidades entre palavras das duas sentenças, por exemplo: a terceira palavra da sentença fonte tem maior probabilidade de estar alinhada às segunda, terceira ou quarta palavras da sentença alvo do que à sua trigésima palavra.

Os incrementos dos modelos IBM 3, 4 e 5 são feitos por meio, respectivamente, de modelos de fertilidade, modelos de reordenação relativa, e técnicas de prevenção do chamado efeito de "deficiência" de alinhamento. Já o modelo de Hidden Markov (HMM), também suportado pelo sistema GIZA++, produz correspondências com base na suposição de que um alinhamento qualquer entre duas palavras é dependente do alinhamento das palavras subsequentes. Mais detalhes sobre as características dos modelos IBM e HMM, bem como sobre as diferenças na qualidade de seus resultados de alinhamento, podem ser encontrados no trabalho de [Och e Ney 2000].

O Meteor [Denkowski e Lavie 2011] é um sistema de alinhamento textual projetado para auxiliar na avaliação do desempenho de sistemas de tradução. Diferente do GIZA++, o Meteor só processa sentenças escritas numa mesma língua. Ele também não gera dicionários de probabilidades de correspondência e tampouco trabalha com aprendizagem de máquina. A estratégia empregada pelo sistema Meteor consiste na busca de dados de alinhamento entre duas sentenças, assim como na Figura 3.1, e para tanto faz uso de informações morfológicas das palavras que as compõe.

A estratégia empregada pelo sistema Meteor para produzir o alinhamento entre duas sentenças é composta por duas fases. Na primeira fase é criado um catálogo com todas as possíveis correspondências entre pares de palavras das sentenças fonte e alvo que preencham algum dos seguintes requisitos:

1. **Similaridade Morfológica:** São geradas correspondências entre todas as palavras que são idênticas nas sentenças fonte e alvo.

2. **Similaridade do Núcleo Morfológico:** Utilizando técnicas de análise morfológica [Porter 2001], o Meteor compara os termos que constitem os núcleos morfológicos de cada palavra das sentenças fonte e alvo. O termo "*run*", por exemplo, pode ser considerado o núcleo de suas variações "*running*", "*ran*", "*runs*" e etc. Caso uma palavra na sentença fonte tenha o mesmo núcleo morfológico de uma palavra na sentença alvo, a correspondência entre elas é adicionada ao catálogo.

3. **Sinonimia:** A partir de consultas ao bancos de sinônimos do WordNet [WordNet 1998], o sistema Meteor adiciona ao catálogo os alinhamentos entre palavras na sentença fonte e alvo que estejam nesse tipo de relação.

4. **Similaridade por Paráfrase:** Uma paráfrase pode ser descrita como uma pequena frase que descreve, de forma simplificada, o significado de uma outra frase ou palavra. A palavra "*expired*" na frase "*This milk has expired weeks ago.*" pode ser descrita pela paráfrase "*become not appropriate for consumption*". Substituindo a palavra "*expired*" por sua paráfrase na frase anterior obtêm-se a frase "*This milk has become not appropriate for consumption weeks ago*", que apesar de mais longa, apresenta com mais clareza o significado da palavra "*expired*". O Meteor utiliza tabelas com frases complexas e suas paráfrases simples equivalentes para encontrar relações de paráfrase nas sentenças fonte e alvo. Caso sejam encontradas estas relações, os alinhamentos entre as palavras correspondentes são adicionados ao catálogo.

Ainda na primeira fase, as correspondências entre palavras são então generalizadas para correspondências entre frases, que possuem um índice de começo e fim em ambas sentenças fonte e alvo.

Feito esse trabalho, na sequência, é iniciada a segunda fase da estratégia de alinhamento, cujo objetivo é selecionar um conjunto de correspondências para comporem o alinhamento final entre as sentenças fonte e alvo. O alinhamento final entre as sentenças é constituído pela maior quantidade de correspondências entre frases que satisfaçam os seguintes critérios, em ordem de importância:

1. Cada palavra da sentença fonte deve ser alinhada a, no máximo, uma palavra da sentença alvo, e vice-versa.

2. O conjunto de alinhamentos deve maximizar a quantidade de alinhamentos possíveis entre as palavras das sentenças fonte e alvo.

3. O conjunto de alinhamentos deve minimizar o número de *chunks* de alinhamento. Um *chunk* é caracterizado pelo alinhamento contíguo e idêntico de uma longa sequência de palavras presentes em ambas sentenças fonte e alvo.

4. O conjunto de alinhamentos deve minimizar o total da soma das distâncias absolutas entre o começo das frases alinhadas nas sentenças fonte e alvo.

A confecção do alinhamento final é feito pelo algoritmo de busca gulosa, que procura, por dentre todos os conjuntos possíveis de correspondências, pelo conjunto que melhor satisfaça os critérios mencionados.

Sistemas de alinhamento textual como o GIZA++ e o Meteor são comumente empregados em tarefas além da simplificação de textos, tais como tradução automática, construção de lexicografias bilíngues, estudos colocacionais e desambiguação de palavras [Dagan, Church e Gale 1993]. Para a tarefa de simplificação de texto, o alinhamento textual pode ser usado na elaboração de modelos de tradução monolíngue entre inglês comum e inglês simplificado [Wubben, Bosch e Krahmer 2012], no desenvolvimento de estratégias de produção de dicionários que assimilam palavras complexas a equivalentes simples [Bott e Saggion 2011], ou também no processo de transdução de árvores, que é uma das tarefas que compõe o sistema de simplificação proposto neste trabalho.

3.2 O Papel da Análise Sintática

Em [Jurafsky e Martin 2008], a análise sintática é definida como a tarefa de reconhecer uma sentença e então definir a estrutura sintática da mesma. Para isto são comumente empregadas as gramáticas livre de contexto (GLC).

Pela definição de [Aho, Sethi e Ullman 1986], as gramáticas livre de contexto são linguagens formais caracterizadas por símbolos terminais, não-terminais, um símbolo de partida e produções gramaticais:

- **Terminais:** São os símbolos básicos a partir dos quais são formadas cadeias de caracteres.
- **Não-Terminais:** São construções sintáticas que descrevem as cadeias de caracteres que definem uma linguagem.
- **Símbolo de Partida:** Não-Terminal a partir do qual são geradas todas as cadeias de caracteres que caracterizam uma linguagem.
- **Produções Gramaticais:** Elementos que especificam as formas com que os terminais e não-terminais podem ser combinados a fim de constituir cadeias de caracteres válidas de uma linguagem.

A Figura 3.2 ilustra um exemplo de como pode ser representada a estrutura sintática da sentença "*The man is tall and strong.*" a partir da gramática livre de contexto ilustrada na Tabela 3.3.

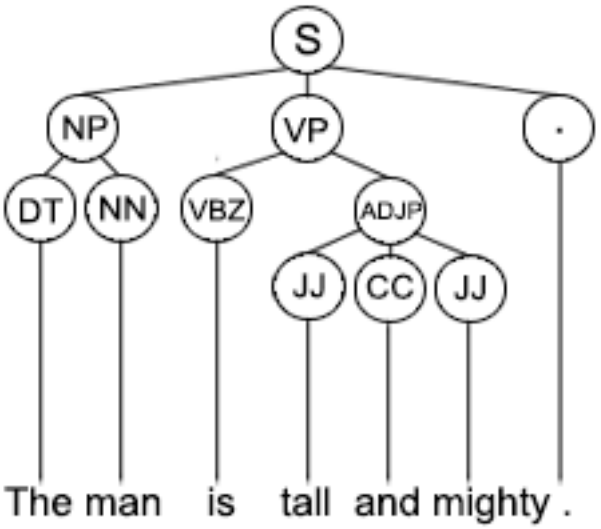


Figura 3.2: Árvore sintática de uma sentença em inglês

Construções:	Descrições:
S → S S	Sentença Composta
S → NP VP	Sentença Nominal
NP → DT NN	sujeito Nominal
VP → VBZ ADJP	Predicado Verbal com Adjetivo
ADJP → JJ CC JJ	Adjetivo Composto com Conector
ADJP → JJ	Adjetivo Simples

Instanciações:	Descrições:
JJ → tall	Adjetivo
JJ → mighty	Adjetivo
JJ → strong	Adjetivo
DT → The	Artigo
NN → man	Substantivo
CC → and	Conector
VBZ → is	Verbo
. → .	Pontuação

Figura 3.3: Gramática livre de contexto

Os dados produzidos no processo de análise sintática são úteis para muitas tarefas no âmbito de Processamento de Linguagem Natural. Alguns exemplos de sistemas que empregam informações sintáticas são:

- **Simplificação de Texto:** A análise é fortemente utilizada por sistemas de simplificação de texto em nível sintático. Alguns destes simplificadores utilizam regras sintáticas produzidas automática ou manualmente para aumentar a redigibilidade de sentenças complexas. Nesse caso, os sistemas de análise sintática são empregados na etapa de confecção das regras e também na análise das sentenças complexas, no objetivo de descobrir quais regras que devem ser utilizadas na sua simplificação.

- **Verificação Gramatical:** São utilizados sistemas de análise sintática para descobrir se uma dada sentença contém erros gramaticais ou não. Caso a sentença não possa ser derivada pelo sistema, isto significa que ela possui erros gramaticais e deve ser corrigida.

- **Tradução Automática:** Sistemas de tradução podem utilizar árvores sintáticas nas etapas de treinamento para auxiliar na tradução entre certas linguagens. Traduzir sentenças do inglês para o alemão, por exemplo, é um grande desafio. Muitas vezes, apenas uma palavra em alemão pode representar uma sentença inteira em inglês, e nesses casos, sistemas

sintáticos de tradução tendem a produzir melhores resultados do que os produzidos por sistemas que usam apenas transformações em nível léxico.

- **Análise Semântica:** Algumas aplicações que fazem análise semântica de sentenças utilizam a análise sintática como uma etapa intermediária na produção de seus resultados. As árvores sintáticas produzidas pelo sistema de análise servem como referência para a confecção de árvores semânticas. Alguns compiladores, por exemplo, constróem a árvore semântica do código com base na árvore sintática construída na etapa anterior de análise sintática.

- **Resolução de Perguntas:** Em alguns sistemas de busca online, como Google e Bing, são comumente embutidas funções para responder perguntas que são digitadas pelo usuário nos campos de busca. Neste contexto, as informações sintáticas são um meio de compreender que tipo de resposta deve ser retornada. Para a pergunta "Quem matou o presidente Kennedy?", por exemplo, a resposta esperada são os nome e dados da pessoa que o fez, enquanto para a pergunta "Quais os livros escritos por Paulo Coelho?" a resposta esperada é uma lista de obras.

Um dos principais sistemas de análise sintática conhecidos para a língua inglesa é o Stanford Parser [Klein e Manning 2003]. Sua estratégia de análise consiste em empregar uma gramática livre de contexto probabilística (ou em inglês *Probabilistic Context-Free Grammar*) para construir a árvore sintática de uma sentença qualquer da língua inglesa.

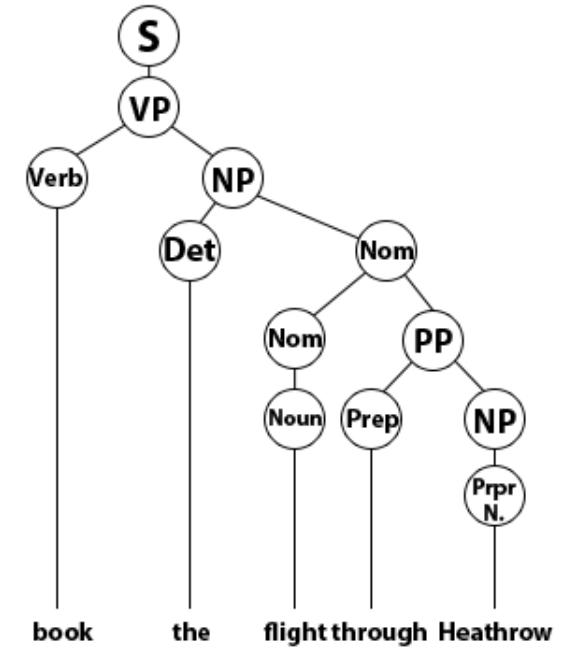
Uma gramática livre de contexto probabilística (PCFG) é uma variação das gramáticas livres de contexto comuns (CFG). A diferença entre as PCFG e as CFG está na probabilidade que é assinalada a cada construção das gramáticas PCFG [Jurafsky e Martin 2008]. Assim como na gramática de exemplo da Figura 3.4, extraída de [Mooney 2013], as probabilidades das construções derivadas de cada não-terminal devem ter soma igual a 1(um).

Construções:	Probabilidades:		Dicionário:	Probabilidades:	
S → NP VP	0.8		Det → the	0.6	
S → Aux NP VP	0.1	1	Det → a	0.2	1
S → VP	0.1		Det → that	0.2	
NP → Pronoun	0.2		Noun → book	0.2	
NP → Proper-Noun	0.2	1	Noun → meal	0.5	1
NP → Det Nominal	0.6		Noun → flight	0.3	
Nominal → Noun	0.3		Verb →book	0.5	
Nominal → Nominal Noun	0.2	1	Verb → include	0.2	1
Nominal → Nominal PP	0.5		Verb → prefer	0.3	
VP → Verb	0.2		Pronoun → I	0.6	
VP → Verb NP	0.5	1	Pronoun → he	0.2	1
VP → VP PP	0.3		Pronoun → she	0.2	
PP → Prep NP	1.0		Proper-Noun → Houston	0.8	1
			Proper-Noun → NWA	0.2	
			Aux → does	1.0	
			Prep → from	0.5	
			Prep → to	0.25	1
			Prep → on	0.25	

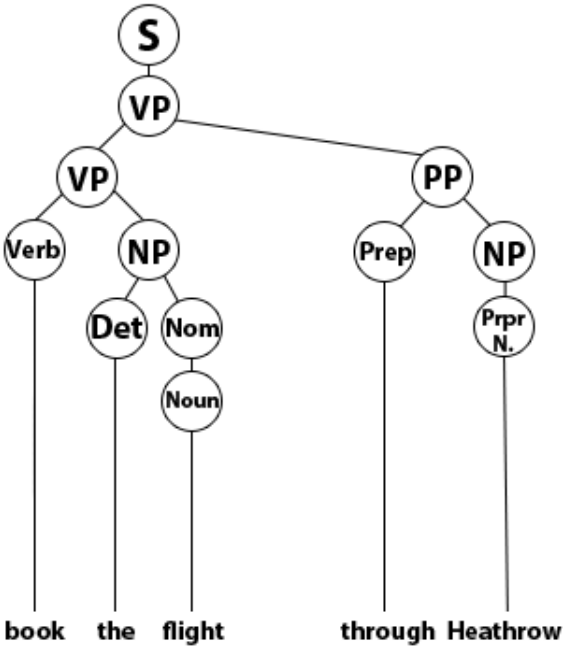
Figura 3.4: Gramática livre de contexto probabilística

A atribuição desses percentuais às produções das gramáticas é feita pelo treinamento não-supervisionado, que avalia grandes corpus de estruturas sintáticas manualmente corrigidas e então extrai as estatísticas de ocorrência de cada construção gramatical encontrada. A especificação da gramática que descreve a língua inglesa e também o corpus de treinamento que são utilizados pelo sistema Stanford Parser são descritos em [Marcus, Marcinkiewicz e Santorini 1993].

A adição das probabilidades às construções da gramática livre de contexto permite que seja feito o processo de desambiguação sintática no caso de uma sentença possuir mais de uma estrutura sintática possível. Considere por exemplo a análise sintática da sentença "*book the flight through Heathrow*" obtida pela aplicação da gramática da Figura 3.4. Os dois resultados possíveis para esta análise sintática junto às suas probabilidades são ilustrados nas Figuras 3.5(a) e 3.5(b), extraídas de [Mooney 2013].



(a) Primeira possibilidade de estrutura sintática



(b) Segunda possibilidade de estrutura sintática

Figura 3.5: Possíveis estruturas sintáticas para uma sentença em inglês

Caso a gramática da Figura 3.4 não oferecesse as probabilidades das construções gramaticais, a escolha entre as duas possíveis árvores sintáticas para a sentença "*book the flight through Heathrow*" teria que ser feita de forma arbitrária. As probabilidades das construções grama-

ticais permitem que se dê preferência a certas estruturas sintáticas sobre outras, levando em consideração as estruturas sintáticas ideais presentes no corpus de treinamento.

Há dois tipos de PCFG, a lexicalizada e a não-lexicalizada. A diferença entre elas está na forma como são estimadas as probabilidades de suas construções gramaticais. Diferente das gramáticas não-lexicalizadas, o treinamento de PCFG's lexicalizadas também estima as probabilidades das construções que levam aos terminais da gramática. Considere o exemplo de gramática lexicalizada da Figura 3.4, onde as construções de seu dicionário possuem probabilidades: caso esta gramática fosse uma PCFG não-lexicalizada, não existiriam probabilidades assinaladas a cada construção de seu dicionário. PCFG's não-lexicalizadas tendem a produzir resultados de análise menos precisos que PCFG's lexicalizadas, porém, agilizam o processo de análise sintática, requerem menos espaço de armazenamento, e suas probabilidades são mais fáceis de serem estimadas.

A gramática utilizada pelo sistema Stanford Parser é da categoria das PCFG's não-lexicalizadas. Para construir sua PCFG, o sistema Stanford Parser utiliza técnicas de aprendizado de máquina sobre um grande corpus que contém sentenças da língua inglesa junto a suas respectivas árvores sintáticas. Os algoritmos de aprendizagem de máquina empregados pelo Stanford Parser têm o objetivo de, a partir das árvores sintáticas de exemplo do corpus, inferir quais as construções gramaticais da língua inglesa. Mais detalhes sobre o corpus utilizado na construção da PCFG não-lexicalizada empregada pelo Stanford Parser podem ser encontrados no trabalho de [Marcus, Marcinkiewicz e Santorini 1993].

3.3 A Tarefa de Tradução Automática de Texto

Uma das tarefas mais desafiadoras da área de Processamento de Linguagem Natural que se relacionam à simplificação automática diz respeito à tradução automática de texto. Seu objetivo é o de empregar métodos computacionais para traduzir textos de uma linguagem para outra, mantendo seu significado, bem como sua redigibilidade. A dificuldade da tarefa se dá pelas muitas diferenças que se encontram entre as linguagens, sejam estas nas palavras em seus dicionários ou na organização das sentenças de acordo com as diferenças estruturais que podem existir entre as línguas utilizadas na composição de sentenças. Alguns exemplos conhecidos por serem de difícil tradução para a língua inglesa são o chinês, o japonês e o alemão.

A tradução automática de texto pode ser empregada no desenvolvimento de estratégias de simplificação automática. Isto significa que, utilizando as mesmas técnicas empregadas no desenvolvimento de sistemas de tradução automática, é possível elaborar sistemas de simplificação que traduzem sentenças do inglês complexo ao inglês simples, transformando assim a simplificação em um processo de tradução monolíngue.

Existem várias maneiras de se produzir um sistema capaz de traduzir uma sentença de uma língua para outra. Os três principais tipos de sistemas de tradução automática são:

- **Sistemas baseados em regras:** Aplicam regras em diferentes níveis de informação para transformar uma sentença de uma linguagem em uma sentença de outra.

- **Sistemas baseados em exemplos:** Baseiam-se no princípio de reuso de traduções já feitas. Tipicamente necessitam de grandes bancos de dados de traduções ou etapas elaboradas de treinamento com corpus paralelos bilíngues.

- **Sistemas estatísticos:** Utilizam modelos probabilísticos de tradução, como o IBM de 1 a 5 ou Hidden Markov Model, descritos na Seção 3.1, para encontrar equivalências entre as palavras que compõe duas linguagens. Estas equivalências são então utilizadas na tradução entre palavras e termos de uma linguagem fonte a uma linguagem alvo.

Além destas categorias, sistemas de tradução podem ser classificados com base no tipo de informação trabalhada durante o processo de tradução. A pirâmide mostrada na Figura 3.6 mostra quais são tais níveis de informação.

De forma geral, quanto mais próximo da base da pirâmide, mais superficial é o nível de informação utilizado pelo tradutor. O método direto localizado na base da pirâmide, é comumente empregado por tradutores em nível lexical, que são tipicamente mais fáceis de serem desenvolvidos porém tendem a possuir grandes limitações em seu potencial. Essas limitações se dão pela incapacidade dos sistemas de tradução direta de lidarem com problemas como referências de longa distância e ambiguidade de palavras. Já tradutores em nível puramente sintático são os que empregam o método de transferência, localizado no meio da pirâmide.

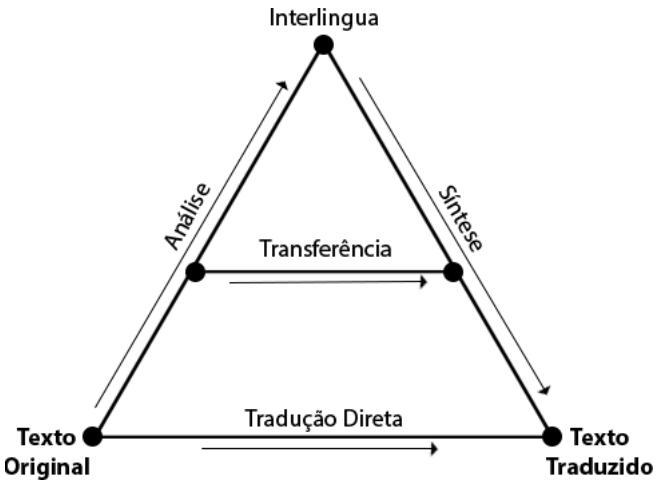


Figura 3.6: Hierarquia dos diferentes níveis de informação considerados no processo de tradução automática

Na transferência são utilizadas representações intermediárias das sentenças a serem traduzidas, normalmente representadas em forma de árvore dos dados sintáticos. O processo de tradução por transferência é composto por três etapas:

1. **Análise:** Primeira etapa, onde são empregados sistemas de análise no intuito de produzir uma representação intermediária da sentença, composta comumente pelos dados sintáticos da mesma armazenados em algum tipo de estrutura de dados.

2. **Transferência:** Etapa seguinte à de análise, que transforma a representação intermediária da sentença em sua linguagem original para uma representação intermediária equivalente na linguagem alvo.

3. **Síntese:** Terceira e última etapa, onde acontece a produção da sentença traduzida a partir da representação intermediária na linguagem alvo.

Mais ao topo da pirâmide estão os sistemas que trabalham em nível semântico, que utilizam as chamadas interlinguas para traduzir sentenças. A interlingua é uma linguagem de abstração capaz de representar o significado de uma sentença independente da linguagem em que ela é escrita. O desenvolvimento de tal representação é um grande desafio, devido às grandes diferenças no conjunto de conceitos que podem ser representados por cada linguagem conhecida na atualidade. Na língua japonesa, por exemplo, existem duas palavras distintas para representar

"irmão mais velho" e "irmão mais novo", palavras estas não encontradas nos dicionários de linguagens como o inglês e o português.

O processo de tradução de sistemas que utilizam interlinguas é dividido em duas etapas:

1. **Análise:** Processo que produz a versão em interlingua do significado de uma sentença na linguagem base. Este processo é normalmente executado em etapas, onde a sentença parte de seu estado inicial, sendo transformada em múltiplas representações intermediárias antes de chegar em nível final de interlingua.

2. **Síntese:** Utiliza a representação de sentido em interlingua gerada na etapa de análise para produzir a sentença traduzida na linguagem alvo. De maneira inversa à análise, a representação em interlingua é transformada, por grande parte dos sistemas desenvolvidos neste modelo, em múltiplas representações intermediárias até chegar em sua forma final que é a sentença traduzida.

3.4 A Tarefa de Transdução de Árvores

No livro *Speech and Language Processing* [Jurafsky e Martin 2008], define-se a transdução como uma tarefa responsável por "mapear uma representação a outra". Na transdução de árvores são estudadas técnicas computacionais para mapear uma árvore fonte em outra árvore modificada alvo. Em tarefas relacionadas ao Processamento de Linguagem Natural, a transdução de árvores é comumente empregada na transformação de árvores sintáticas de sentenças.

A função da transdução de árvores no sistema proposto neste trabalho é confeccionar regras de simplificação em nível sintático e lexical, e então aplicá-las na confecção de versões simplificadas de sentenças complexas. Esta estratégia foi formulada com base nos trabalhos correlatos de [Woodsend e Lapata 2011] e [Cohn e Lapata 2008], que empregam a transdução de árvores nas tarefas de simplificação automática e compressão de textos, respectivamente.

O trabalho de [Woodsend e Lapata 2011] emprega a transdução de árvores no processo de simplificação de textos em inglês. Neste trabalho, as regras de transdução de árvores são representadas por uma gramática quasi-síncrona[1] que descreve, em forma de produções gramaticais,

[1] O conceito de gramática quasi-síncrona é descrito em [Smith e Eisner 2006] como sendo uma modelagem para produções gramaticais que utiliza como base de conhecimento relações entre sentenças fonte e alvo representadas em forma de grafo. As relações que caracterizam o grafo de uma gramática quasi-síncrona compreendem conexões de intra e interdependência entre as palavras de sentenças fonte e alvo.

as regras de transdução que transformam a árvore sintática de uma sentença complexa em uma árvore sintática equivalente simples.

Em [Woodsend e Lapata 2011], a confecção das produções da gramática quasi-síncrona é feita pelo aprendizado de máquina, que utiliza como base de conhecimento um corpus paralelo com sentenças complexas alinhadas a suas equivalentes simples [Yatskar et al. 2010]. Para cada par de sentenças no corpus paralelo é produzido um grafo onde estão determinadas as relações existentes entre elas. Nestes grafos, as relações de interdependência entre as sentenças fonte e alvo são representadas por dados de alinhamento textual, enquanto as relações de intradependência são representadas pelas construções sintáticas que as caracterizam. Na Figura 3.7, extraída de [Cohn e Lapata 2009], é ilustrada uma forma da representação das relações de intra e interdependência da gramática quasi-síncrona empregada em [Woodsend e Lapata 2011], e a Equação 3.1 mostra um exemplo de produção gramatical extraída destes dados.

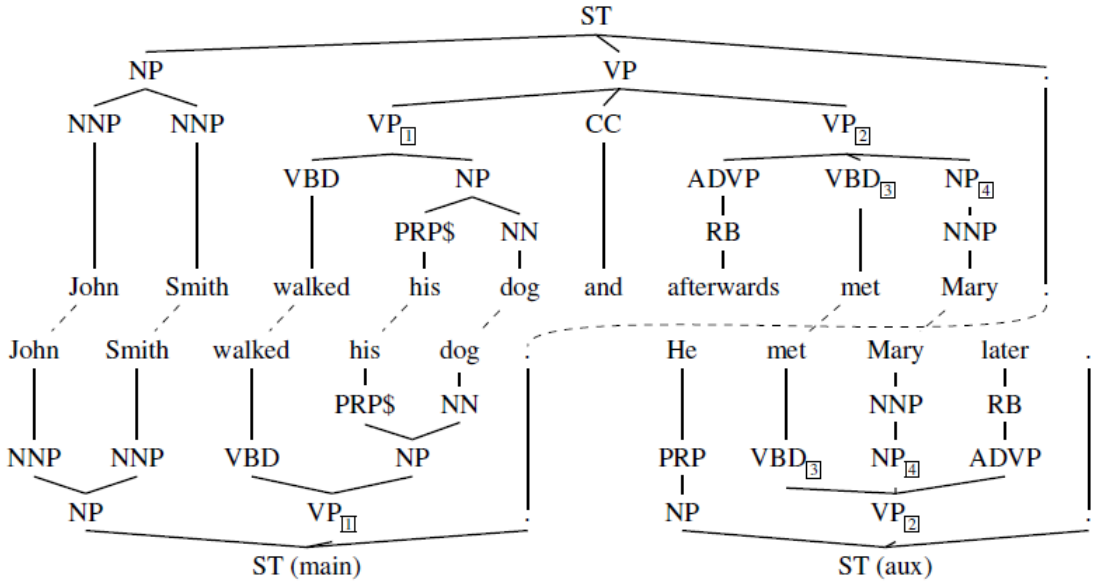


Figura 3.7: Relações de intra e interdependência entre sentenças

$$\langle \text{VP};\ \text{VP},\ \text{ST} \rangle \rightarrow \langle [\text{VP}_1\ and\ \text{VP}_2]\ ;\ [\text{VP}_1]\ ,\ [\text{NP}\,[\text{PRP}\ He]\,\text{VP}_2] \rangle \tag{3.1}$$

Cada produção da gramática quasi-síncrona caracteriza uma regra de transdução de árvores, que é composta por uma árvore fonte alinhada a uma árvore alvo. Na produção gramatical da Equação 3.1, o componente à esquerda ($\langle \text{VP};\ \text{VP},\ \text{ST} \rangle$) descreve os nós raiz das árvores fonte e alvo, enquanto o componente à direita ($\langle [\text{VP}_1\ and\ \text{VP}_2]\ ;\ [\text{VP}_1]\ ,\ [\text{NP}\ [\text{PRP}\ He]\ \text{VP}_2] \rangle$) descreve a forma dos nós adjacentes às raizes das mesmas. Observa-se que a produção gramatical em questão contém três elementos em ambos seus componentes à esquerda e à direita, significando que esta regra de transdução divide a árvore fonte de raiz VP em duas árvores correspondentes de raizes VP e ST.

A confecção das produções gramaticais é realizada pelo mapeamento de padrões da árvore sintática da sentença complexa a padrões equivalentes da árvore sintática da sentença simples. Observa-se que a produção gramatical da Equação 3.1, por exemplo, mapeia o padrão complexo $[\text{VP}\ [\text{VP}_1\ and\ \text{VP}_2]]$ da árvore sintática ao topo da Figura 3.7 aos padrões simples equivalentes $[\text{VP}_1]$ e $[\text{ST}\ [\text{NP}\ [\text{PRP}\ He]\ \text{VP}_2]]$ da árvore sintática mais abaixo.

A aplicação de uma regra de transdução a uma sentença complexa é feita por meio de dois passos: busca seguida de tradução. Primeiramente é realizada a busca pelo padrão complexo da regra na árvore sintática da sentença complexa. Caso o padrão não seja encontrado, a regra de transdução não é aplicável. Se encontrado, os dados do padrão complexo são traduzidos ao padrão simples da regra, e desta forma a árvore sintática da sentença complexa é modificada. Considere por exemplo a árvore sintática da frase "*Henry cried and then ran away*", e o padrão complexo da regra de transdução da Equação 3.1. A Figura 3.8 revela onde o padrão complexo pode ser encontrado nesta sentença, e a Figura 3.9 mostra como é realizada a tradução para o padrão simples.

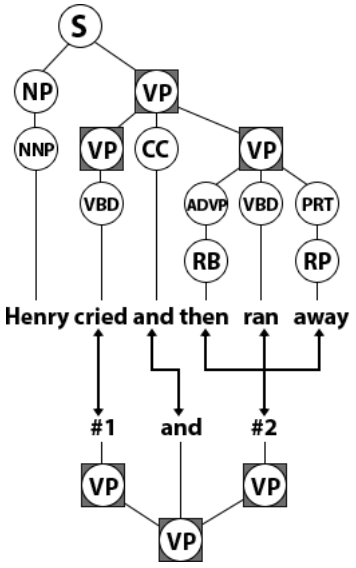


Figura 3.8: Relações entre duas estruturas de árvore compatíveis

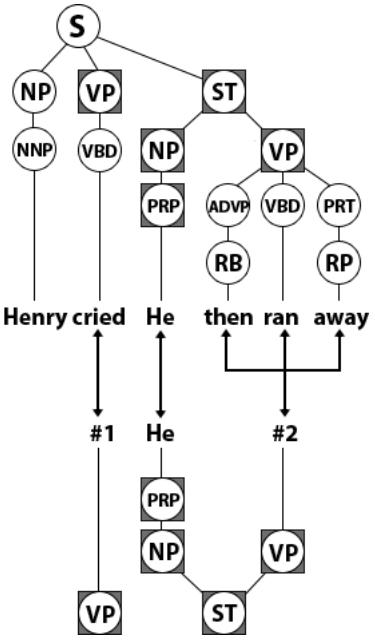


Figura 3.9: Processo de transdução da árvore sintática de uma sentença de acordo com as árvores alvo de uma regra de transdução

As regras produzidas pela estratégia de [Woodsend e Lapata 2011] são lexicalizadas, ou seja, os nós folha das árvores fonte e alvo da regra de transdução podem conter não só variáveis, mas também palavras. A lexicalização das regras utilizadas em [Woodsend e Lapata 2011] permite que uma determinada regra aplique ambos processos de simplificação em nível sintático

e lexical. Entretanto, são raros os casos onde o algoritmo de inferência de regras é capaz de lexicalizar todas as palavras complexas de uma determinada entrada do corpus paralelo e ao mesmo tempo confeccionar uma regra genérica o suficiente para que seja aplicável a um grande arranjo de sentenças de estrutura sintática similar. Um dos principais objetivos do sistema proposto neste trabalho é endereçar esta limitação, utilizando algoritmos distintos para a seleção de regras sintáticas e lexicais, e então realizando a simplificação em nível sintático e lexical em duas etapas consecutivas.

O trabalho de [Cohn e Lapata 2008] apresenta uma estratégia para a aplicação da transdução de árvores como uma solução para a compressão de textos. Neste trabalho, as regras de transdução de árvores são extraídas por aprendizado de máquina e modeladas em forma de uma gramática similar à utilizada em [Woodsend e Lapata 2011].

As diferenças principais entre as gramáticas empregadas em [Cohn e Lapata 2008] e [Woodsend e Lapata 2011] está na forma do corpus paralelo utilizado nas etapas de treinamento, e também nas técnicas de extração de produções gramaticais. Em [Cohn e Lapata 2008], o corpus paralelo é composto por sentenças originais alinhadas a sentenças equivalentes mais curtas, e na extração são priorizadas produções gramaticais que transduzam a árvore sintática fonte em uma árvore sintática alvo com menos nós folha. Já em [Woodsend e Lapata 2011], o corpus paralelo possui sentenças complexas alinhadas a sentenças equivalentes simples, e no processo de extração são priorizadas produções gramaticais que representem operações de simplificação, como segmentação de sentenças e substituição de palavras complexas.

Existem também sistemas de transdução de árvore flexíveis, que são dedicados a produzir e aplicar regras de transdução de árvores para qualquer propósito, seja ele a compressão de texto, simplificação de texto, ou qualquer outra tarefa. O *Tree Transducer Toolkit* (T3) [Cohn e Lapata 2009] é um exemplo de sistema deste tipo. Dependendo do tipo de corpus paralelo utilizado em seu treinamento, o T3 pode ser empregado em qualquer tarefa que possa se beneficiar da transdução de árvores. O T3 possui várias funções, dentre elas estão a função de produção de regras de transdução de árvores, e também a função de aplicação das mesmas. A função de produção de regras de transdução (ou função de "*harvesting*") do T3 é um componente do sistema de simplificação proposto neste trabalho, e a descrição detalhada de seu funcionamento é documentada na Seção 5.1.2. Mais detalhes sobre o funcionamento do sistema

T3 podem ser encontrados em [Cohn e Lapata 2009].

3.5 Modelos Estatísticos de Linguagem

A principal finalidade dos modelos estatísticos de linguagem é representar os padrões das construções sentenciais de uma língua por meio de distribuições probabilísticas.

As distribuições probabilísticas dos modelos estatísticos de linguagem são comumente estimadas em uma etapa de treinamento, que utiliza como base de conhecimento grandes corpus contendo exemplos de sentenças de uma determinada língua. Neste processo são catalogadas as frequências de ocorrência dos chamados N-gramas, que são todos os segmentos compostos por N palavras encontrados nas sentenças do corpus de treinamento. As frequências dos N-gramas permitem que seja estimada a frequência com que uma sentença qualquer tende a ocorrer no uso de uma determinada língua com base em suas distribuições probabilísticas. O cálculo da probabilidade de sentenças pode ser aplicado de diferentes formas:

- **Reconhecimento de fala:** Considere, por exemplo, a tarefa de descobrir qual a sentença que é falada por um locutor em um trecho de áudio ruidoso. Muitas vezes, sistemas de reconhecimento de fala ficam indecisos com relação a algumas palavras onde o áudio tem mais ruídos, e nestes casos é necessário decidir qual palavra foi dita pelo locutor dentre todas as possibilidades. Um modelo de linguagem pode ser utilizado nesta decisão, uma vez que suas distribuições probabilísticas permitem estimar qual palavra melhor se encaixa no contexto em que está inserida. Suponha que um sistema de reconhecimento de fala esteja indeciso quanto à ultima palavra da sentença "*I went to the*", e que existam duas possibilidades: "*beach*" e "*peach*". Se apropriadamente estimado, um modelo de linguagem da língua inglesa evidenciará que "*I went to the beach*" tem uma probabilidade muito maior do que a alternativa "*I went to the peach*", pois a primeira sentença ocorre com mais frequência do que a segunda.

- **Tecnologias assistivas para indivíduos com deficiências cognitivas:** De forma similar com que são aplicados no reconhecimento de fala, modelos estatísticos de linguagem podem auxiliar pessoas com dificuldades na pronúncia de palavras. Muitas vezes pessoas com problemas de pronúncia têm dificuldades em se comunicar, pois encontram muitas

dificuldades em entender o que está sendo dito. Modelos estatísticos de linguagem podem ser utilizados no reconhecimento da sentença sendo falada pelo usuário com deficiência, para que esta então seja direcionada a um módulo de áudio que sintetiza e repronuncia a sentença para o interlocutor.

- **Correção gramatical:** pelos modelos estatísticos de linguagem é possível que sejam encontrados alguns tipos de erros gramaticais em sentenças. Considere por exemplo a sentença "*In thirty minuets we will reach the shore*". Neste caso, o modelo de linguagem é capaz de indicar que a palavra "*minuets*" tem baixa probabilidade de ocorrência no contexto onde está inserida, e pode sugerir que a mesma seja substituída pela alternativa "*minutes*", que tem grande probabilidade de ocorrência neste contexto.

- **Preenchimento automático em dispositivos móveis:** Devido a seu reduzido tamanho, dispositivos móveis como celulares e tablets possuem recursos de digitação limitados que impedem que o usuário digite de forma ágil. Modelos estatísticos de linguagem podem ser utilizados na construção de sistemas que analisam a sentença sendo digitada, e então fazem sugestões de preenchimento automático para que o usuário não tenha que digitar toda a sentença.

As probabilidades dos N-gramas de um modelo de linguagem são estimadas a partir da contagem de suas ocorrências no corpus de treinamento. Considere por exemplo um corpus de treinamento com |*V*| palavras distintas. A probabilidade de um segmento composto por apenas uma palavra w do corpus (N-grama de tamanho 1, ou unigrama) pode ser estimada a partir da Equação 3.2.

$$P(w) = \frac{\text{count}(w)}{|V|} \tag{3.2}$$

Para N-gramas com N de tamanho maior que 1 o cálculo é feito de outra forma. No cálculo de probabilidade de um N-grama de tamanho 2 (ou bigrama), por exemplo, é estimada a probabilidade da segunda palavra w_n do segmento ser precedida pela primeira palavra w_{n-1}. A Equação 3.3 descreve o cálculo da probabilidade de um bigrama.

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)} \tag{3.3}$$

Onde w é uma palavra qualquer do corpus e $C(w_{n-1}w_n)$ é o número de vezes com que o bigrama composto pelas palavras w_{n-1} e w_n aparece no corpus de treinamento. A Figura 3.10, extraída de [Stevenson 2012], ilustra um exemplo de Tabela contendo algumas das probabilidades dos bigramas de um modelo de linguagem.

Segunda palavra do bigrama

	He	poured	out	a	glassful	and	drank	it	greedily
He	0	.000274	0	.000579	0	.003720	.000609	.000091	0
poured	0	0	.311828	.021505	0	0	0	.053763	0
out	.000393	0	0	.017424	0	.014673	0	.000131	0
a	.000017	0	0	0	.000103	.000035	0	0	0
glassful	0	0	0	0	0	.066667	0	0	0
and	.015837	.000146	.001412	.0241428	0	0	.000392	.009381	.000022
drank	0	0	.032258	.088710	0	.008064	0	.120968	0
it	.000468	.000027	.008364	.008832	0	.005668	0	.000027	.000027
greedily	0	0	0	0	0	0	0	0	0

Primeira palavra do bigrama

Figura 3.10: Probabilidades de bigramas de um modelo de linguagem

O cálculo de N-gramas pode também ser generalizado para qualquer tamanho de N, resultando na Equação 3.4.

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{\sum_w C(w_{n-N+1}^{n-1})} \tag{3.4}$$

A probabilidade de uma sentença pode ser calculada a partir das probabilidades dos N-gramas do modelo de linguagem. A Equação 3.5 descreve a probabilidade de uma sentença composta por n palavras quando calculada a partir das probabilidades dos bigramas de um modelo de linguagem.

$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k|w_{k-1}) \tag{3.5}$$

Como exemplo, a Figura 3.11 mostra como é calculada a probabilidade da sentença *He poured out a glassful and drank it greedily* com relação às probabilidades da Figura 3.10.

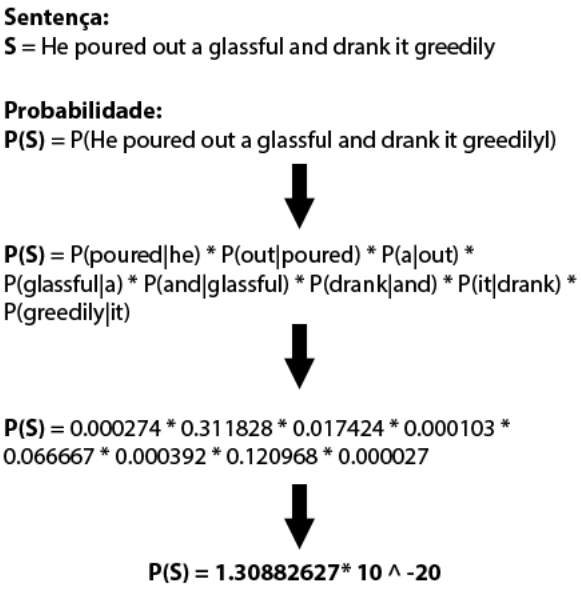


Figura 3.11: Cálculo da probabilidade de uma sentença por um modelo de linguagem

Um exemplo de sistema que constroí modelos estatísticos de linguagem é o SRILM [Stolcke 2002]. O SRILM dá suporte à estimativa de N-gramas de qualquer tamanho, e também utiliza técnicas de normalização de probabilidades de N-gramas. A normalização consiste no processo de atribuir coeficientes maiores que zero a N-gramas sem nenhuma ocorrência no corpus de treinamento. Este processo é importante, pois um N-grama de probabilidade zero leva a probabilidade de qualquer sentença que o possua a zero também, o que acaba prejudicando a qualidade da estimativa. O trabalho de [Chen e Goodman 1996] apresenta um estudo comparativo entre diversas técnicas de normalização comumente utilizadas em modelos estatísticos de linguagem.

Capítulo 4

Trabalhos Correlatos

Assim como previamente mencionado no Capítulo 1, a estratégia de simplificação proposta neste trabalho é de caráter sintático-lexical. Nesse sentido, são empregadas regras que modificam sentenças complexas tanto em seu nível sintático quanto lexical no objetivo de simplificá-la. Isto implica que a estratégia descrita neste trabalho foi estruturada de forma a associar algumas das principais funcionalidades de estratégias de simplificação em nível sintático e lexical.

Um dos trabalhos da área de simplificação de textos em nível sintático é o apresentado em [Siddharthan 2004]. Neste trabalho, o processo de simplificação em nível sintático é dividido em três etapas: análise, transformação e regeneração. Na análise são empregados algoritmos de extração de informações morfossintáticas da sentença complexa. Em seguida são aplicadas regras de simplificação confeccionadas por humanos sobre a estrutura sintática da sentença complexa. Finalmente, no processo de regeneração, o sistema avalia se a sentença simplificada construída é coerente e se não possui erros gramaticais.

Em [Gasperin et al. 2009] é descrita outra arquitetura de sistema de simplificação. A proposta é estruturada em duas camadas. A primeira camada tem o objetivo de identificar que tipo de simplificação uma determinada sentença de entrada necessita (como segmentação, remoção de segmentos não-essenciais, etc). Para isto é gerado um classificador, criado por meio da aplicação de algoritmos de aprendizado não-supervisionado sobre um corpus paralelo de sentenças complexas e simples. O classificador analisa as estruturas sintática e lexical da sentença, e então determina, com base no treinamento recebido, de que forma a sentença deve ser simplificada. A segunda camada utiliza regras sintáticas de simplificação para reestruturar a sentença complexa de entrada.

A tarefa de simplificação automática da língua inglesa pode ser interpretada também como

sendo uma tarefa de tradução monolíngue entre o inglês complexo e o inglês simples. Os trabalhos de [Zhu, Bernhard e Gurevych 2010] e [Wubben, Bosch e Krahmer 2012] propõem estratégias de simplificação em nível sintático que utilizam técnicas de tradução automática probabilística[1] no objetivo de traduzir sentenças do inglês complexo ao inglês simples. De forma análoga, no trabalho de [Specia 2010] é apresentada uma estratégia para tradução entre o português brasileiro complexo e o português brasileiro simples. Estes trabalhos fazem o treinamento de modelos de tradução entre inglês complexo e simples por meio de corpus paralelos de sentenças complexas e simples, e então os aplicam na simplificação de sentenças complexas quaisquer.

Em [Bach et al. 2011] é apresentada uma estratégia de simplificação em nível sintático que utiliza modelos matemáticos probabilísticos e algoritmos de decodificação para reestruturar sentenças complexas. A estratégia de [Bach et al. 2011] tem o objetivo de simplificar sentenças em inglês que se encaixem no modelo "Subject-Verb-Object" (ou "Sujeito-Verbo-Objeto" em português) por meio da reestruturação da árvore sintática destas sentenças. Entretanto, é bastante comum que sentenças complexas da língua inglesa possuam períodos muito longos, referências a múltiplos objetos e também o uso constante de referências anafóricas, o que dificulta, ou até mesmo impossibilita a identificação de seu sujeito e objeto principais. Sentenças deste tipo não são simplificáveis pela estratégia de [Bach et al. 2011], restringindo assim sua aplicabilidade a sentenças complexas curtas, que possuam um número reduzido de componentes sintáticos.

Estratégias de simplificação como as apresentadas em [Keskisarkka 2012], especializam-se apenas em reduzir a complexidade em nível lexical de sentenças. A função principal da estratégia de simplificação apresentada neste trabalho é substituir termos complexos por termos equivalentes simples em sentenças complexas da língua sueca. Neste trabalho, a busca por sinônimos para termos complexos é feita por meio de uma consulta ao banco de dados linguístico do Wordnet [WordNet 1998], e a escolha de qual sinônimo é o mais simples dentre todos os disponíveis é realizada por meio de uma consulta ao Oxford Psycholinguistics Database [Dunbar 1994], que possui estatísticas de frequência de uso das palavras da língua sueca. Em [Lal e Ruger 2002] é apresentado um sistema de simplificação bastante similar, porém di-

[1] A tradução automática probabilística é um técnica de tradução automática cuja principal característica é o uso de modelos matemáticos e cálculos estatísticos na tradução de uma sentença de uma língua fonte para uma língua alvo.

recionado à língua inglesa. Seu sistema também faz simplificação em nível lexical utilizando a substituição de palavras complexas por sinônimos extraídos do Wordnet. A decisão de qual sinônimo de um certo termo complexo é o mais simples, é feita com base em um modelo de frequência de uso de palavras.

O trabalho de [Carroll et al. 1998] é uma das primeiras publicações que apresenta uma estratégia que simplifica sentenças complexas em nível sintático e lexical. Nesta estratégia, as simplificações sintática e lexical são executadas em sequência, ou seja: a sentença complexa é primeiramente submetida à simplificação em nível sintático, e em seguida à simplificação em nível lexical. Em sua etapa de simplificação sintática, o sistema transforma sentenças escritas na voz passiva para a voz ativa e remove segmentos desimportantes, enquanto na etapa de simplificação lexical, o mesmo substitui termos complexos por sinônimos encontrados no banco de dados Wordnet. Por suportar apenas dois tipos de transformação em nível sintático, a estratégia de [Carroll et al. 1998] não é capaz de lidar com sentenças complexas que carecem da aplicação de outros processos de simplificação, como a segmentação de sentenças.

O sistema de simplificação Facilita, cuja arquitetura é descrita em [Watanabe et al. 2009], é mais um exemplo de estratégia de simplificação híbrida, destinada a simplificar páginas da internet. Diferente da estratégia empregada pelo trabalho de [Carroll et al. 1998], o sistema Facilita simplifica textos complexos utilizando não apenas regras de simplificação em nível sintático e lexical, mas também técnicas de sumarização e de elaboração de texto. A simplificação no sistema Facilita é realizada de forma iterativa:

- **Primeira iteração:** As características da página da internet são avaliadas, e então é feita a decisão de quais as formas mais adequadas de simplificação para as sentenças de sua estrutura.

- **Segunda iteração:** Utilizam-se as técnicas de simplificação escolhidas em um processo de prototipação evolutiva de simplificação da página da internet. A página é inicialmente submetida a uma das técnicas de simplificação escolhidas, criando assim um protótipo primário simplificado da mesma. O protótipo primário é submetido a uma segunda técnica de simplificação, e assim por diante até que sejam empregadas todas as técnicas de simplificação escolhidas na primeira iteração.

Outro trabalho que apresenta uma estratégia de simplificação híbrida é o de [Kandula, Curtis e Zeng-Treitler 2010]. Sua estratégia tem o objetivo de simplificar documentos que tratam de assuntos da área da saúde por meio de técnicas de simplificação em nível sintático e semântico. Nesta estratégia, assume-se que qualquer sentença com mais de 10 palavras necessita de simplificação em nível sintático, e portanto é submetida a uma série de módulos de pré-processamento e simplificação. Dentre os módulos estão o módulo de segmentação de períodos, tokenização, categorização gramatical, simplificação gramatical e validação. Após a etapa de simplificação em nível sintático, a sentença é submetida à etapa de análise semântica, que substitui palavras e termos complexos da medicina por sinônimos simples ou então paráfrases construídas manualmente pelos autores.

Analisando os trabalhos mencionados, é possível determinar quais os méritos dos mesmos quanto a aos incrementos de desempenho explorados na tarefa de simplificação automática, e também identificar quais as limitações impostas pelas estratégias empregadas. A confecção manual de regras de simplificação em nível sintático, técnica utilizada nos trabalhos de [Siddharthan 2004] e [Gasperin et al. 2009], é um processo bastante custoso que requer a contratação de profissionais qualificados a identificar problemas de complexidade em sentenças, e também formalizar métodos para modificar ou reconstruir sentenças com estes problemas. Uma língua natural, como por exemplo o inglês, permite que sejam construídas sentenças complexas com infinitas estruturas sintáticas distintas, e portanto torna-se inviável confeccionar manualmente um conjunto de regras grande o suficiente para tratar de todos os possíveis problemas de complexidade em nível sintático.

Para solucionar este problema, é necessário que sejam empregadas técnicas de produção automática de regras de simplificação. Para realizar esta tarefa, o sistema proposto neste trabalho emprega a tarefa de transdução de árvores. Uma das razões da escolha pela transdução de árvores nesta tarefa são os resultados satisfatórios obtidos por [Woodsend e Lapata 2011], que também emprega a transdução de árvores na simplificação de textos, assim como relatado na Seção 3.4.

Foi constatado também que a aplicação de apenas um tipo de simplificação, como realizado em [Siddharthan 2004] e [Keskisarkka 2012], tende a limitar a capacidade da estratégia de simplificação. A simplificação em nível sintático, por exemplo, não contempla técnicas para que

sejam substituídos ou removidos termos técnicos e palavras incomuns de sentenças complexas, enquanto a simplificação em nível lexical não é capaz de modificar ou reconstruir a árvore sintática de uma sentença que carece deste tipo de simplificação, como por exemplo no caso de sentenças muito longas ou que fazem uso constante da voz passiva.

Em contrapartida, trabalhos que realizam a associação entre múltiplas formas de simplificação apresentam resultados promissores, especialmente no uso da simplificação automática como ferramenta assistiva para portadores de patologias da linguagem ou semi-analfabetismo. Ambos trabalhos de [Carroll et al. 1998] e [Watanabe et al. 2009] visam auxiliar usuários com dificuldade na leitura, e as discussões apresentadas sugerem que, para que usuários como os portadores da Afasia possam compreender conteúdo de caráter formal ou técnico, as sentenças complexas em seu conteúdo devem ser reestruturadas sintaticamente, filtradas de palavras complexas ou termos técnicos, e também sumarizadas.

Considerando a discussão apresentada, é possível sugerir três dos principais aspectos que fundamentam os processos de concepcção e desenvolvimento do sistema de simplificação automática proposto neste trabalho. São eles:

1. O desempenho do sistema não deve ser dependente do trabalho de especialistas humanos, e, portanto, devem ser evitadas técnicas como o uso de regras confeccionadas manualmente.

2. O sistema deve ser capaz de simplificar o maior conjunto de sentenças complexas possível, evitando se restringir a sentenças com apenas um tipo de estrutura sintática.

3. Sentenças complexas comumente apresentam complexidade em ambas estruturas sintática e lexical, devendo ser simplificadas em diferentes níveis de informação para que a qualidade da versão simplificada não seja comprometida.

Capítulo 5

O Sistema Proposto

Não apenas com um interesse científico investigativo, mas também buscando a implementação de um de sistema computacional capaz de empregar a estratégia de simplificação sintático-lexical por meio da transdução de árvores, nesse trabalho apresentamos um protótipo desenvolvido. Vale mencionar que esse trabalho corresponde aos avanços obtidos durante o estágio de graduação do autor, proporcionado pelo Programa Ciências sem Fronteiras, do qual tomaram parte pesquisadores dos Curso de Ciências da Computação da UNIOESTE/campus Cascavel e da Universidade de Sheffield (Reino Unido).

Esse sistema vem sendo implementado nas linguagens de programação Python e Java, interagindo ainda com os sistemas externos T3, de transdução de árvores [Cohn e Lapata 2009], o sistema de análise sintática Stanford Parser [Klein e Manning 2003], o sistema de alinhamento de palavras Meteor Aligner [Denkowski e Lavie 2011], e também o modelo de linguagem SRILM [Stolcke et al. 2011]. A função de cada um dos sistemas externos mencionados são apropriadamente especificadas ao longo desta monografia. A opção pelas linguagens Python e Java se deveu, não só mas principalmente, à facilidade que essas linguagens apresentam entre si para o desenvolvimento de uma programação híbrida, e de fácil interação com sistemas terceiros. Outro fator que influenciou nesta escolha foi a portabilidade de ambas as linguagens, permitindo assim que o sistema desenvolvido rode tanto em Unix (Linux) e Windows.

Em linhas gerais, a função do sistema proposto é produzir uma sentença simplificada em inglês de saída a partir de uma sentença complexa em inglês apresentada como entrada. Sua estrutura é composta por três Módulos: Treinamento, Simplificação e Ranqueamento. A Figura 5.1 ilustra o fluxograma do sistema. Para facilitar a visualização, os Módulos acima referidos estão identificados na Figura 5.1 por M1, M2 e M3, respectivamente.

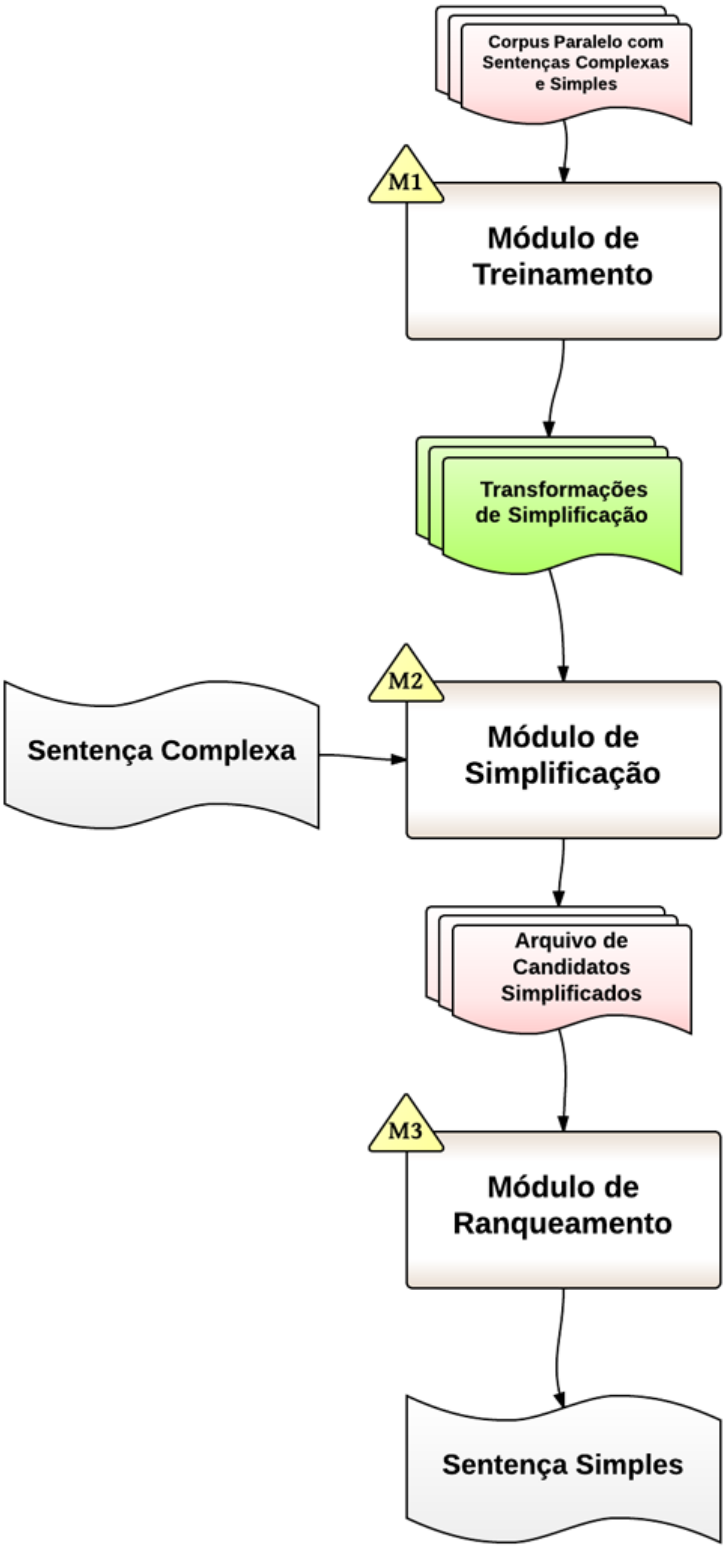


Figura 5.1: Fluxograma da ordem de execução do sistema de simplificação proposto

As breves descrições da função de cada um dos Módulos que compõe o sistema são:

- **M1:** Módulo de Treinamento, responsável por produzir regras de simplificação em nível lexical e sintático.
- **M2:** Módulo de Simplificação, responsável por aplicar as regras de simplificação produzidas pelo Módulo de Treinamento na produção de múltiplas versões simplificadas de uma sentença complexa passada como entrada.
- **M3:** Módulo de Ranqueamento, cuja função é avaliar as sentenças simplificadas produzidas pelo Módulo de Simplificação, classificá-las de acordo com métricas de pontuação, e então selecionar a candidata de maior pontuação como sendo a versão simplificada definitiva da sentença complexa original.

Os Módulos, embora complementares, são executados de forma independente. As saídas geradas de uns servem de entrada para os outros. No intuito de esclarecer a funcionalidade de cada um dos Módulos, nas seções seguintes serão descritos os aspectos computacionais principais levados em consideração na programação de cada um.

5.1 Função, Especificação e Programação do Módulo de Treinamento (M1)

O Módulo M1 de Treinamento é o responsável por produzir regras de simplificação de texto, tanto no nível lexical quanto sintático. Para tanto ele recebe como entrada um corpus paralelo com sentenças complexas alinhadas as suas correspondentes mais simples. São dois os arquivos gerados por M1 como saída: um arquivo de texto, que armazena regras de simplificação em nível lexical, e outro arquivo de texto que armazena regras de simplificação em nível sintático.

A execução de M1 se dá por meio do *script* reproduzido no Algoritmo 1. O *script* divide o processamento em 3 etapas distintas, onde faz várias chamadas a funções em Python e aplicações em Java desenvolvidas especificamente para este trabalho, e também a sistemas externos Stanford Parser, Meteor Aligner e T3. A Figura 5.2 ilustra o fluxo de execução do Módulo de Treinamento do sistema proposto.

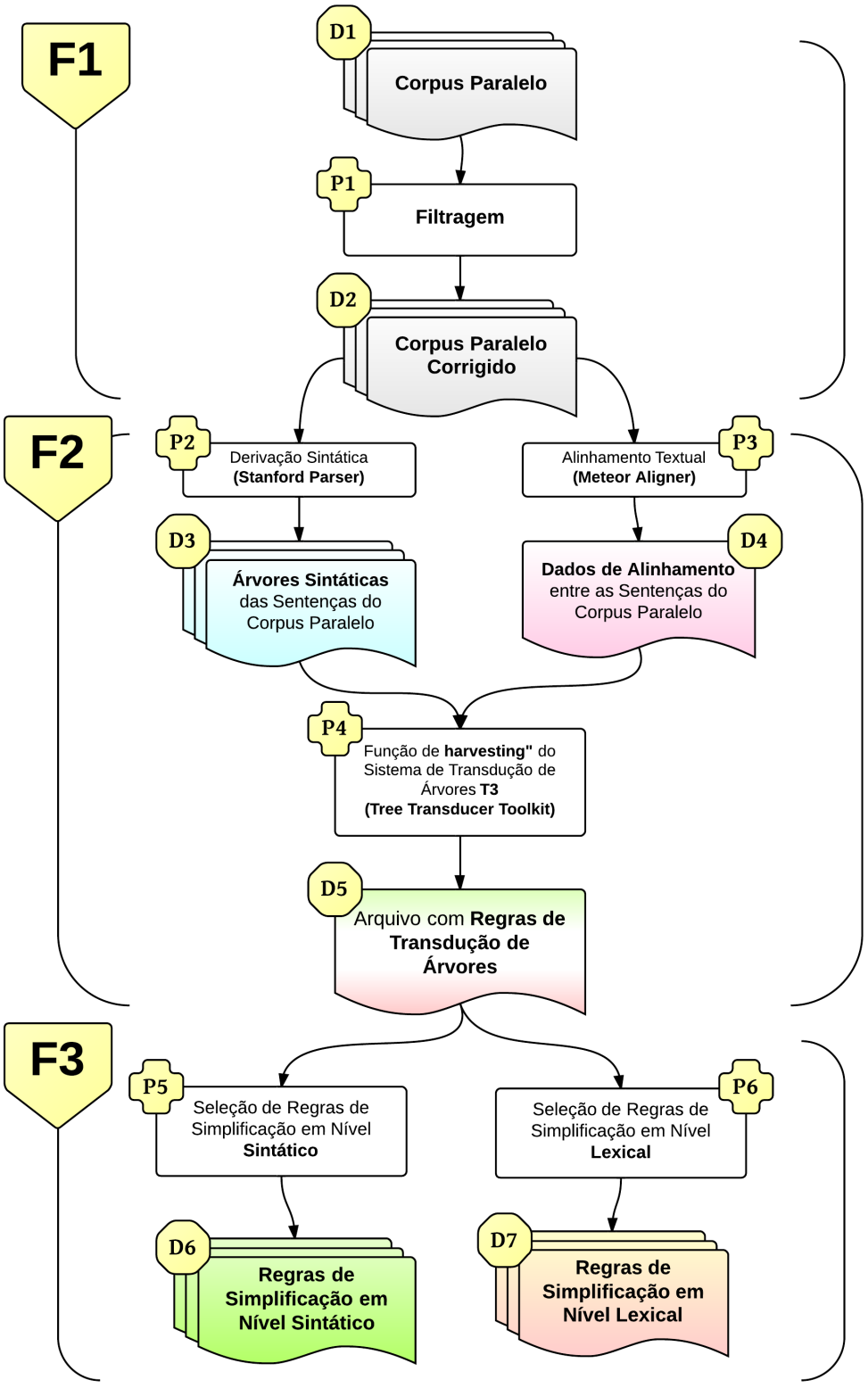


Figura 5.2: Fluxograma de execução do Módulo de Treinamento

Algoritmo 1 *Script* do Módulo de Treinamento (M1)

Receba corpus paralelo de entrada $D1$
$D2$ = Filtre o corpus paralelo $D1$
$D3$ = Produza as árvores sintáticas de $D2$ pelo $Stanford\ Parser$
$D4$ = Produza os dados de alinhamento de $D2$ pelo $Meteor\ Aligner$
$D5$ = Produza regras de transdução de árvores a partir dos dados de $D3$ e $D4$
$D6$ = Extraia regras de simplificação em nível sintático de $D5$
$D7$ = Extraia regras de simplificação em nível lexical de $D5$
Apresente como saída os arquivos $D6$ e $D7$

Na Figura 5.2 são encontrados os seguintes componentes:

- **D1:** Corpus paralelo de entrada do Módulo M1, composto por 2 tipos de informação. De um lado, exemplos de sentenças originais escritas em inglês e de outro, essas mesmas sentenças, porém reescritas de forma mais simples, tanto em termos lexicais quanto estruturais. A organização desse arquivo será explicada com mais detalhes na Seção 5.1.1.

- **P1:** Aplicação em Java desenvolvida especificamente para este projeto. Sua função é "limpar" o conteúdo do arquivo D1, via a remoção de entradas duplicadas e caracteres inválidos do corpus paralelo D1, e assim produzir sua versão corrigida D2.

- **D2:** Versão corrigida do corpus paralelo de entrada, produzida ao final da execução da aplicação P1.

- **P2:** Processo responsável pela produção das árvores sintáticas das sentenças dos dois lados do corpus paralelo D2. Esse processo o é realizado por meio de uma chamada ao sistema externo Stanford Parser.

- **P3:** Processo responsável pelo alinhamento de palavras entre as sentenças dos dois lados do corpus paralelo D2. Esse trabalho é realizado por meio de uma chamada a um sistema externo: o Meteor Aligner.

- **D3:** Arquivo produzido ao final de P2, contendo as árvores sintáticas de cada sentença presente no corpus paralelo D2.

- **D4:** Arquivo produzido ao final de P3, contendo os alinhamentos de cada par de sentenças complexa e simples presente no corpus paralelo D2.

- **P4:** Processo responsável pela produção de regras de transdução de árvores. A produção destas regras é feita pela função de "*harvesting*" do sistema de transdução de árvores T3, que recebe os arquivos D3 e D4 como entrada e produz as regras de transdução de árvores como saída.

- **D5:** Arquivo produzido ao final de P4, contendo as regras de transdução produzidas pela função de "*harvesting*" do sistema de transdução T3.

- **P5:** Aplicação em Python desenvolvida especificamente para este projeto, cuja função é extrair regras de simplificação em nível sintático do arquivo D5.

- **P6:** Aplicação em Python desenvolvida especificamente para este projeto, cuja função é extrair regras de simplificação em nível lexical do arquivo D5.

- **D6:** Arquivos de saída do Módulo M1, contendo regras de simplificação em nível sintático.

- **D7:** Arquivos de saída do Módulo M1, contendo regras de simplificação em nível lexical.

- **F1:** Primeira fase de execução do Módulo M1.

- **F2:** Segunda fase de execução do Módulo M1.

- **F3:** Terceira e última fase de execução do Módulo M1.

Como já dito, o treinamento é implementado em 3 fases, a saber: Pré-processamento (F1), Produção de Regras de Transdução (F2) e Seleção de Regras de Simplificação (F3), cujas funções seguem abaixo descritas.

1. **Pré-processamento (F1):** Nessa fase, o corpus paralelo (D1) passa por um processo de filtragem (P1), que corrige entradas problemáticas de sua estrutura e então produz o corpus paralelo corrigido (D2).

2. **Produção de Regras de Transdução (F2):** De posse do arquivo D2 como entrada, nessa fase é chamada a função de "harvesting" do sistema T3, que gera um arquivo com regras de transdução de árvores (D5).

3. **Seleção de Regras de Simplificação (F3):** Fase onde são aplicadas rotinas de seleção (P5 e P6) ao arquivo de regras de transdução de árvores (D5), que ao seu final produzem arquivos com regras de simplificação em nível sintático (D6) e em nível lexical (D7).

5.1.1 A Fase de Pré-processamento (F1)

Nessa fase, o Módulo de Treinamento recebe e prepara seus arquivos de entrada para a fase F2 seguinte. Esta preparação tem o objetivo de corrigir os dados de entrada pela remoção de elementos problemáticos que possam prejudicar o desempenho do restante do sistema de simplificação.

O documento de índice D1 presente na Figura 5.2 representa a única entrada do Módulo M1. Nele, encontra-se o corpus de sentenças paralelas. O corpus paralelo é composto por dois documentos: um arquivo contendo sentenças complexas, e outro contendo versões simples das sentenças do primeiro arquivo. Estes dois documentos devem ambos conter uma sentença por linha, e devem também ser alinhados em nível de sentença. O alinhamento em nível de sentença entre os dois documentos implica que, para cada sentença contida no documento de sentenças complexas, deve existir uma sentença simples equivalente na mesma linha do documento de sentenças simples. A Tabela 5.1 contém um trecho do corpus paralelo produzido em [Coster e Kauchak 2011], que alinha sentenças complexas extraídas da Wikipedia à sentenças simples equivalentes extraídas da Simple Wikipedia. Este corpus é o mesmo utilizado no treinamento do sistema proposto na realização dos experimentos documentados no Capítulo 6.

Sentença Complexa	**Sentença Simples**
Grand View is a city in Owyhee County , Idaho , United States .	Grand View is a city of Idaho in the United States .
The population was 144 at the 2000 census .	The population was 144 .
Grand View is a city in Owyhee County , Idaho , United States .	Grand View is a city of Idaho in the United States .
Thomas Eastoe Abbott was an English poet , born in East Dereham , Norfolk .	Thomas Eastoe Abbott was an English poet .

Tabela 5.1: Corpus paralelo fornecido como entrada para o Módulo de Treinamento

Muitos trabalhos relacionados à área de PLN fazem uso de corpus paralelos muito grandes. Em [Specia 2010], por exemplo, é utilizado um corpus com 4.483 sentenças complexas em português alinhadas a suas equivalentes simples na construção de um sistema de simplificação automática de texto. Construir corpus paralelos manualmente permite maior controle sobre a qualidade de seu conteúdo, porém é um processo bastante custoso e demorado. Existem trabalhos como [Bott e Saggion 2011] e [Coster e Kauchak 2011] que apresentam técnicas para a construção automática de corpus paralelos. Técnicas como estas permitem que corpus paralelos sejam construídos de maneira menos custosa e mais ágil, porém corpus paralelos produzidos desta forma comumente contêm entradas com erros gramaticais, ou também entradas cujas sentenças fonte e alvo não compartilhem do mesmo significado.

O corpus ilustrado na Tabela 5.1 foi confeccionado automaticamente pela técnica apresentada em [Coster e Kauchak 2011], e possui entradas que ilustram alguns destes problemas. Observe por exemplo a segunda entrada da Tabela 5.1, onde as sentenças complexa e simples não compartilham inteiramente do mesmo significado.

Antes de concluir a fase F1 é necessário que o corpus paralelo de entrada se submeta a um processo de correção. Este processo tem o objetivo de remover ou substituir caracteres que, em determinados contextos, podem impactar negativamente na qualidade qualidade das regras de simplificação que serão produzidas posteriormente. Alguns exemplos de caracteres espúrios são parênteses debalanceados, aspas simples e aspas duplas desbalanceadas. Outros exemplos de problemas comumente encontrados em corpus paralelos são sentenças que não terminam em ponto final, que terminam em vírgula ou múltiplos pontos em seguida, e também entradas onde as duas sentenças são idênticas dos dois lados.

Uma sentença com caracteres espúrios no corpus paralelo pode fazer com que sistemas de análise sintática não consigam encontrar produções gramaticais para representar sua estrutura sintática. Para contornar esta dificuldade, os sistemas podem ser forçados a remover segmentos desta sentença para que esta se encaixe em alguma das produções gramaticais disponíveis. Neste processo de remoção, é possível que sejam eliminados não apenas os caracteres espúrios, mas também elementos lexicais importantes, como preposições e substantivos. Remover elementos lexicais deste tipo pode prejudicar a qualidade das regras de simplificação produzidas posteriormente na fase F3 de produção de regras de transdução.

Algoritmo 2 Algoritmo de Filtragem de Corpus Paralelo

$Complexas$ = Abrir sentenças complexas de $D1$
$Simples$ = Abrir sentenças simples de $D1$
$ComplexasD2$ = Abrir novo arquivo
$SimplesD2$ = Abrir novo arquivo
enquanto fim dos arquivos $Complexas$ e $Simples$ não atingido **faça**
  C = Leia linha de $Complexas$
  S = Leia linha de $Simples$
  se C diferente de S **então**
    Substitua todos “ . ” por “ .” em C e S
    Substitua todos “. .” por “.” em C e S
    Remova todos “” desbalanceados em C e S
    Remova todos “”” desbalanceados em C e S
    Remova todos “‘” desbalanceados em C e S
    Remova todos “‘‘” desbalanceados em C e S
    Remova todos “(” desbalanceados em C e S
    Remova todos “)” desbalanceados em C e S
    Remova espaços aos começo e final de C e S
    Remova “’,” ao final de C e S
    Remova “;” ao final de C e S
    se C não termina em “ .’ **então**
      se C termina em “’.’ **então**
        Remova “’.” ao final de C
      fim se
      Adicione “ .” ao final de C
    fim se
    se S não termina em “ .’ **então**
      se S termina em “’.’ **então**
        Remova “’.” ao final de S
      fim se
      Adicione “ .” ao final de S
    fim se
    Substitua todos “ “ por “ ’ em C e S
    Salve C em $ComplexasD2$
    Salve S em $SimplesD2$
  fim se
fim enquanto

O componente P1 da Figura 5.2 representa uma aplicação desenvolvida em Java, cuja função é contornar o problema dos corpus paralelos mencionado. Esta aplicação é acionada pelo *script* em Python principal do Módulo de Treinamento para aplicar medidas de filtragem ao corpus paralelo. A aplicação P1 tem como objetivo evitar que o sistema Stanford Parser, chamado pelo processo de índice P2 da Figura 5.2, produza dados incorretos de análise sintática. É importante

ressaltar que a aplicação P1 não trata de erros gramaticais, ortográficos ou de ordem semântica presentes no corpus paralelo, mas sim remove alguns elementos lexicais problemáticos que comumente afetam negativamente o processo de análise sintática.

A aplicação P1 avalia todas as linhas de ambos arquivos do corpus paralelo D1, e utiliza expressões regulares para remover caracteres espúrios e também correspondências compostas por duas sentenças idênticas. O Algoritmo 2 descreve o funcionamento da aplicação em Java P1.

Em sua conclusão, a aplicação P1 gera uma versão corrigida do corpus paralelo, identificada pelo índice D2 na Figura 5.2. O corpus paralelo corrigido D2 é então encaminhado como entrada à fase de produção de regras de transdução do Módulo de Treinamento.

5.1.2 A Fase da Produção de Regras de Transdução (F2)

A fase F2 recebe como entrada o arquivo D2 gerado na fase anterior, e utiliza-o na confecção de regras de transdução de árvores. O sistema de transdução de árvores utilizado para produzir estas regras é o T3 (*Tree Transducer Toolkit*) [Cohn e Lapata 2009]. Os dados de entrada necessários para que o sistema T3 possa produzir regras de transdução precisam ser preparados previamente. Essa preparação é feita pelos sistemas externos chamados nos processos de índice P2 e P3 da Figura 5.2. P2 e P3 correspondem, respectivamente, ao analisador sintático (Stanford Parser [Klein e Manning 2003]) e o alinhador de palavras (Meteor Aligner [Denkowski e Lavie 2011]).

A saída gerada por P2 corresponde ao arquivo de índice D3, que contém as árvores sintáticas montadas pelo Stanford Parser tanto das sentenças complexas, quanto das sentenças simples equivalentes. Cada entrada do arquivo D3, assim como nos exemplos da Tabela 5.2, é composta por um par de linhas, que contém a árvore sintática de uma sentença complexa do corpus D2, chamada de árvore fonte, seguida da árvore sintática de sua equivalente simples, chamadas de árvore alvo.

Descrição	Árvore Sintática
Árvore sintática da sentença complexa (Árvore Fonte):	(S (NP (DT The)(NN population))(VP (VBD was)(NP (NP (CD 144))(PP (IN at)(NP (DT the)(CD 2000)(NN census)))))(. .))
Árvore sintática da sentença simples equivalente (Árvore Alvo):	(S (NP (DT The)(NN population))(VP (VBD was)(NP (CD 144)))(. .))
Árvore sintática da sentença complexa (Árvore Fonte):	(S (NP (NNP Hamilton))(VP (VBD was)(NP (NP (DT the)(NN county)(NN town))(PP (IN of)(NP (NNP Lanarkshire)))))(. .))
Árvore sintática da sentença simples equivalente (Árvore Alvo):	(S (NP (NNP Hamilton))(VP (VBZ is)(NP (NP (DT a)(NN town))(PP (IN in)(NP (NP (NNP South)(NNP Lanarkshire))(, ,)(NP (NNP Scotland))))))(. .))
Árvore sintática da sentença complexa (Árvore Fonte):	(S (NP (NNP Christopher)(NNP Eccleston))(VP (VBZ is)(NP (DT an)(JJ English)(NN stage)(, ,)(NN film)(CC and)(NN television)(NN actor)))(. .))
Árvore sintática da sentença simples equivalente (Árvore Alvo):	(S (NP (NNP Christopher)(NNP Eccleston))(VP (VBZ is)(NP (DT an)(JJ English)(NN actor)))(. .))
Árvore sintática da sentença complexa (Árvore Fonte):	(S (NP (NNP Romania))(VP (VBZ is)(NP (DT a)(JJ secular)(NN state))(, ,)(S (VP (RB thus)(VBG having)(NP (DT no)(JJ national)(NN religion)))))(. .))
Árvore sintática da sentença simples equivalente (Árvore Alvo):	(S (S (NP (NNP Romania))(VP (VBZ is)(NP (DT a)(JJ secular)(NN state)))(. .))(S (NP (DT This))(VP (VBZ means)(SBAR (S (NP (NNP Romania))(VP (VBZ has)(NP (DT no)(JJ national)(NN religion))))))(. .)))
Árvore sintática da sentença complexa (Árvore Fonte):	(S (S (WHNP (RB exactly)(WP what))(S (NP (NNS records))(VP (VBD made)(NP (PRP it)))))(S (WHNP (WP which))(S (NP (NNS ones))(VP (VBP are)(VP (VBN involved))))))
Árvore sintática da sentença simples equivalente (Árvore Alvo):	(S (WHNP (WP what))(S (NP (NNS recorsd))(VP (VBP are)(VP (VBN involved)))))

Tabela 5.2: Entradas que compõem o arquivo D3 produzido pelo sistema Stanford Parser

Os rótulos que identificam cada nó das árvores fonte e alvo representam construções e classes gramaticais da língua inglesa. Alguns dos rótulos presentes nas entradas da Tabela 5.2 estão abaixo identificados.

- S: Representa a construção gramatical "Simple declarative clause", ou "Oração simples declarativa" em português.
- NP: Representa a construção gramatical "Noun phrase", ou "Frase nominal" em português.
- VP: Representa a construção gramatical "Verb phrase", ou "Frase verbal" em português.
- PRP: Representa a classe gramatical "Personal pronoun", ou "Pronome pessoal" em português.
- NNS: Representa a classe gramatical "Plural noun", ou "Substantivo no plural" em português.
- RB: Representa a classe gramatical "Adverb", ou "Advérbio" em português.

O significado de todos os rótulos utilizados pelo sistema Stanford Parser na construção das árvores gramaticais da Tabela 5.2 são apresentadas no trabalho de [Marcus, Marcinkiewicz e Santorini 1993].

O arquivo D4 constitui a saída da chamada P3 ao sistema Meteor Aligner. Este arquivo contém os dados de alinhamento textual entre cada par de sentenças complexa e simples que compõe o corpus paralelo corrigido D2. O Meteor Aligner, descrito com mais detalhes na Seção 3.1, é um sistema de alinhamento textual que utiliza uma estratégia não-probabilística para confeccionar resultados de alinhamento de palavras entre duas sentenças. O trabalho apresentado em [Denkowski e Lavie 2011] revela que o sistema Meteor produz alinhamentos muito similares aos confeccionados por linguistas, e por isso foi escolhido para compor esta etapa do sistema.

A Tabela 5.3 ilustra o formato do arquivo de índice D4. É importante ressaltar que os arquivos D3 e D4 devem estar alinhados ao arquivo D2. Isto significa que para cada entrada do arquivo D2 devem haver entradas correspondentes de dados sintáticos e alinhamento textual nos arquivos D3 e D4, respectivamente.

Dados de Alinhamento Textual
10-0 11-1 14-2 15-9 18-19 0-20 3-21 1-25 5-28 6-29 7-30 8-31 9-32 17-36
0-0 1-1 2-2 3-3 4-4 5-5 6-6 7-7 8-8 9-9 10-10 11-11 12-12 13-13 14-14 15-15 16-16 17-17
0-0 1-1 2-3 3-4 4-5 5-6 6-7 7-8 8-9 9-10
1-4 2-6 9-7 21-9 19-10 27-19 3-27 3-28 3-29 4-30 7-31

Tabela 5.3: Entradas que compõe o arquivo D4 produzido pelo sistema Meteor Aligner

A formatação dos dados de saída gerados por P3 é ilustrada na Tabela 5.3. Neste formato, cada linha representa os dados de alinhamento textual entre as sentenças das árvores fonte e alvo de uma entrada do arquivo D3. Cada linha é composta por uma sequência de elementos constituídos por dois números inteiros separados por um hífen. O número à esquerda do hífen representa o índice de uma palavra da sentença na árvore fonte, enquanto o número à direita do hífen representa o índice da palavra na sentença da árvore alvo a cuja palavra na árvore fonte está alinhada.

Uma vez concluídos os processos P2 e P3, a chamada P4 ativa a função de "*harvesting*" do sistema de transdução T3, função esta cujo objetivo é inferir regras de transdução de árvores a partir de dados sintáticos e de alinhamento de palavras de um corpus paralelo. Os dados sintáticos e de alinhamento passados como entrada na chamada P4 contendo os dados sintáticos e de alinhamento são os arquivos D3 e D4, respectivamente.

A função de "*harvesting*", primeiramente recebe os arquivos D3 e D4, e então associa as árvores sintáticas aos dados de alinhamento de palavras na confecção de correspondências entre frases. No contexto de transdução de árvores, uma frase pode ser descrita como o segmento de uma sentença que constitui uma das construções gramaticais que compõem sua estrutura sintática. A Figura 5.3, extraída de [Cohn e Lapata 2009], ilustra um exemplo de correspondências entre frases confeccionadas pela função de "*harvesting*" a partir dos dados sintáticos e de alinhamento textual de um par de sentenças.

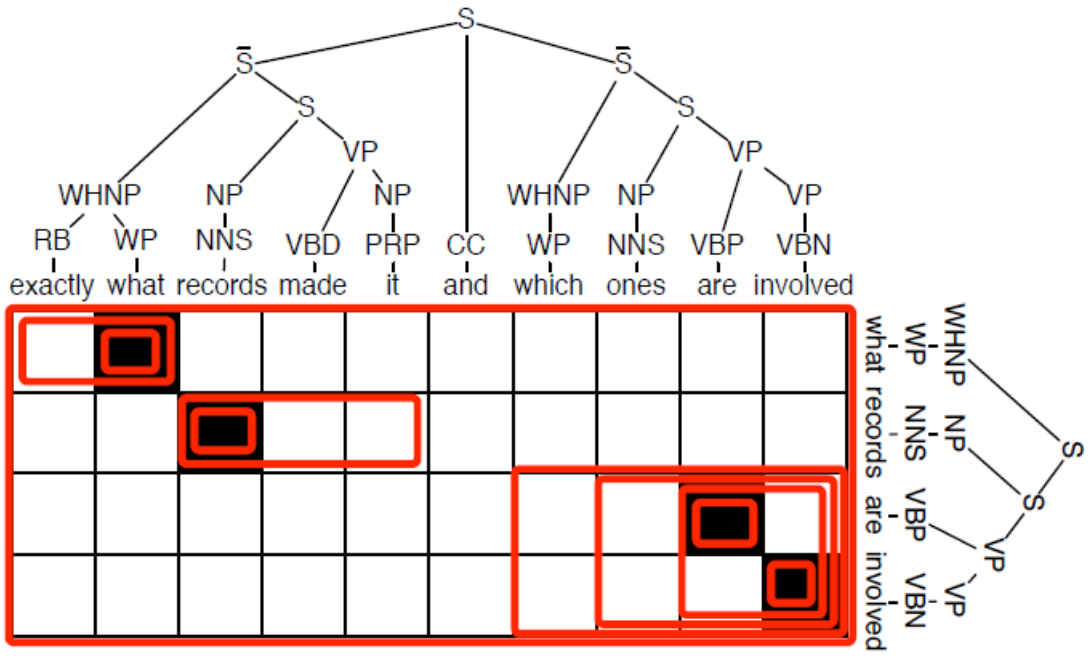


Figura 5.3: Associação dos dados sintáticos e de alinhamento textual feita pela função de "*harvesting*" do sistema T3

Os elementos da Figura 5.3 podem ser descritos da seguinte forma:

- **Árvore ao topo da Figura:** Árvore sintática da sentença complexa (árvore fonte).
- **Árvore à direita da Figura:** Árvore sintática da sentença simples equivalente (árvore alvo).
- **Quadrados de fundo preto:** Indicam que existe alinhamento entre uma palavra da sentença complexa e uma palavra da sentença simples equivalente. Observa-se que na Figura 5.3 existe alinhamento entre as palavras "*what*", "*records*", "*are*" e "*involved*", que estão presentes em ambas sentenças complexa e simples.
- **Quadrados de fundo branco:** Indicam que não existe alinhamento entre uma palavra da sentença complexa e uma palavra da sentença simples equivalente.
- **Retângulos de borda curva:** Indicam que existe alinhamento entre uma frase da sentença complexa e uma frase da sentença simples equivalente. Na Figura 5.3 existe alinhamento entre a frase "*exactly what*" na sentença complexa e a frase "*what*" na sentença simples,

entre a frase "*records made it*" e a frase "*records*" na sentença simples, e também vários outros casos.

Na notação usada pelo sistema T3 em sua função de "*harvesting*", uma regra de transdução de árvores é constituída por uma árvore fonte não-nula alinhada a um de dois possíveis tipos de árvore alvo: árvore alvo não-nula, o que caracteriza uma regra de transdução comum, ou árvore alvo nula, que caracteriza uma regra de remoção [Cohn e Lapata 2009].

Após produzir as correspondências entre frases, a função de "*harvesting*" utiliza-as para confeccionar regras de transdução de árvores. Primeiramente são produzidas as regras de remoção, identificadas por árvores de frases base sem correspondência com qualquer árvore de frase alvo. Em seguida são confeccionadas as regras de transdução comum, caracterizadas por árvores de frases base com correspondência a alguma árvore de frase alvo.

Uma vez confeccionadas, cada regra de transdução comum passa então pelo processo de generalização. Na generalização são produzidas versões modificadas da regra original, que contém variáveis nos lugares dos nós presentes em diferentes profundidades das árvores fonte e alvo. A Figura 5.4 mostra um exemplo de regra de transdução comum junto a algumas de suas variações generalizadas, onde as variáveis são os nós folha cujo rótulo começa com o caractere "#". Um nó folha de variável representa uma subárvore sintática de forma qualquer. Uma variável pode representar tanto uma palavra, quanto a estrutura sintática completa de uma construção sentencial.

A Tabela 5.4 ilustra o formato do arquivo D5, produzido ao final da execução da chamada P4. Cada entrada do arquivo D5 é composta por uma árvore fonte e uma árvore alvo separadas por tabulação.

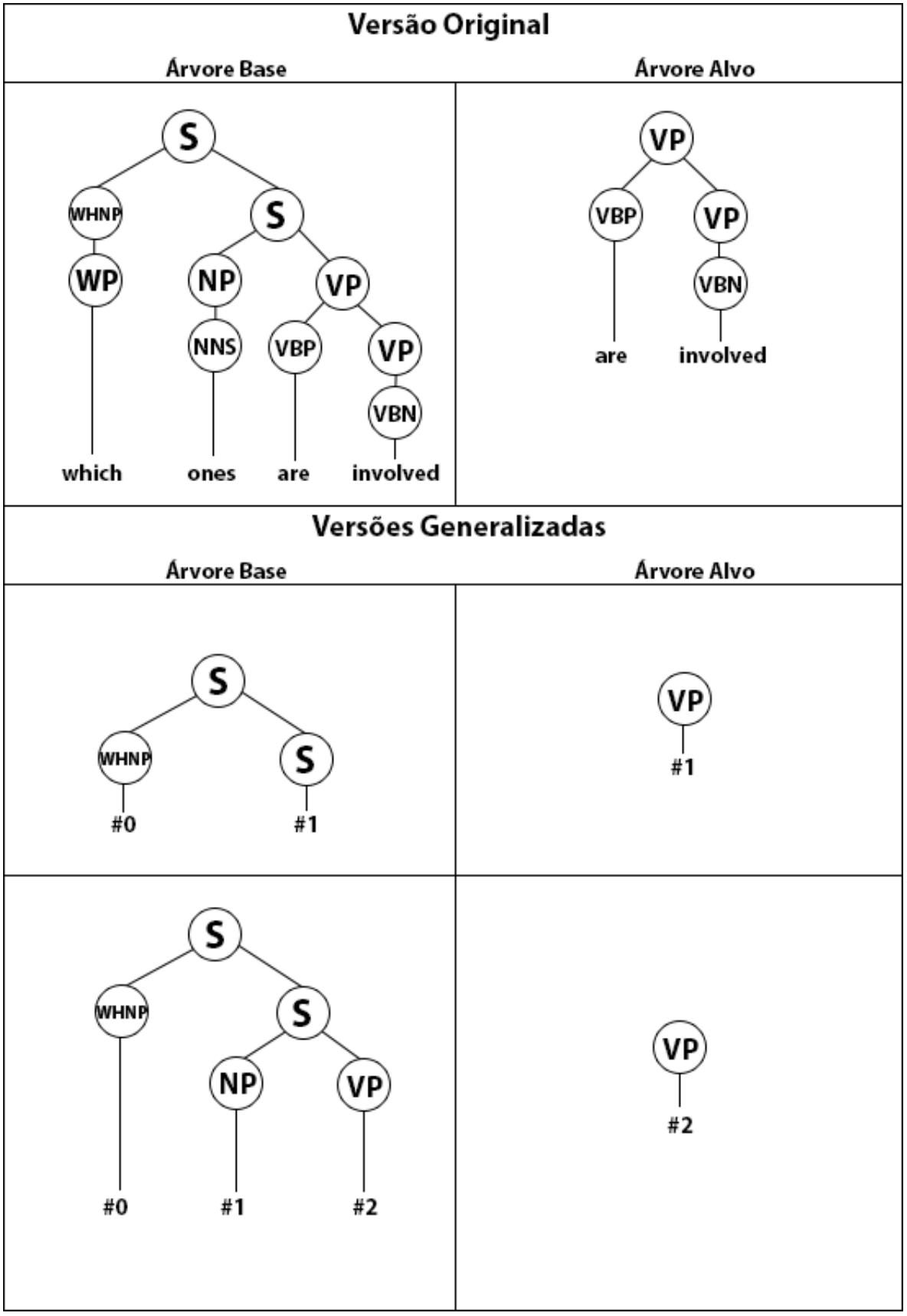


Figura 5.4: Regras de transdução produzidas a partir do processo de generalização

árvore fonte	**Árvore Alvo**
(NP (DT #0) (ADJP #1) (NN #2) (NN #3))	(NP (ADJP #1) (NN #2) (NN #3))
(S (NP (NP (NNS #0)) (VP #1)) (VP #2) (. #3))	(S (NP (NNP #0)) (VP #1) (. #3))
(S (PP #0) (NP #1) (VP (VBD #2) (ADJP #3) (PP #4)) (. #5))	(S (PP #0) (NP #1) (VP (VBD #2) (ADJP #3)) (. #5))
(S (PP #0) (PP #1) (, #2) (NP #3) (VP #4))	NULL
(S (PP #0) (S #1) (CC #2) (S #3))	NULL

Tabela 5.4: Regras de transdução produzidas pela função de "*harvesting*" do sistema T3

O arquivo D5 contém apenas parte das regras generalizadas que podem ser inferidas pela função de "*harvesting*" a partir dos dados dos arquivos D3 e D4. A razão pela qual o sistema T3 armazena apenas uma parte das regras de transdução, é o grande volume de possíveis regras que podem ser inferidas a partir de cada par de sentenças. No caso das sentenças da Figura 5.3 por exemplo, apesar de serem sentenças relativamente curtas (a maior delas contendo 10 palavras) e terem apenas 4 palavras alinhadas entre si, a função de "*harvesting*" é capaz de inferir um total de 45 regras de transdução a partir de seus dados. Considerando que as sentenças podem atingir dezenas palavras cada e o corpus paralelo pode conter centenas de milhares de pares destas sentenças, o sistema T3 seleciona apenas as regras de transdução mais genéricas para compor o arquivo final de saída D5.

A conclusão do processo P4 finaliza a etapa de produção de regras e dá início à fase final F3 de seleção de regras de simplificação do Módulo M1 de Treinamento.

5.1.3 A Fase de Seleção de Regras de Simplificação (F3)

O arquivo gerado na etapa anterior (D5), na sequência, é direcionado à fase F3 de seleção de regras de simplificação. Em D5 existem regras de transdução, que podem ou não serem consideradas regras de simplificação em nível sintático ou lexical. Na fase F3, o Módulo de Treinamento tem como objetivo buscar por estas regras de simplificação no arquivo D5, e então produzir dois arquivos: um contendo regras de simplificação em nível sintático, e outro contendo regras de simplificação em nível lexical. Estes são os dois arquivos de saída do Módulo

de Treinamento, representados pelos índices D6 e D7 da Figura 5.2. Para produzir os arquivos D6 e D7 são invocadas as aplicações de índice P5 e P6, respectivamente.

P5 é uma aplicação de seleção de regras de simplificação em Python, cujo fluxo de execução é descrito no Algoritmo 3.

Algoritmo 3 Algoritmo de seleção de regras de simplificação em nível sintático

```
RegrasT3 = Abrir arquivo D5
D6 = Abrir novo arquivo
para cada regra R de RegrasT3 faça
  Arvores = Compile a expressão regular "([^\t]+)\t([^\n]+)" sobre R
  ArvEsquerda = Arvores[0]
  ArvDireita = Arvores[1]
  FolhasEsq = Compile a expressão regular "\\(([^\s]+) [\^\\(\\)]+\\)" sobre ArvEsquerda
  VariaveisEsq = Compile a expressão regular "\\([^\s]+ #[^\\(\\)]+[^\\)]*\\)" sobre ArvEsquerda
  VariaveisDir = Compile a expressão regular "\\([^\s]+ #[^\\(\\)]+[^\\)]*\\)" sobre ArvDireita
  PontosArvEsquerda = Compile a expressão regular "\\(\\. \\.\\)" sobre ArvEsquerda
  PontosArvDireita = Compile a expressão regular "\\(\\. \\.\\)" sobre ArvDireita
  Conjuncoes = Itens de classe CC em FolhasEsq
  se ArvEsquerda diferente de ArvDireita e ArvDireita diferente de NULL então
    se ordem de VariaveisEsq diferente da ordem de VariaveisDir então
      Escreva R em D6
    senão
      se Tamanho de Conjuncoes maior que 0 então
        Escreva R em D6
      senão
        se PontosArvEsquerda menor que PontosArvDireita então
          se Tamanho de PontosArvEsquerda > 0 então
            Escreva R em D6
          fim se
        fim se
      fim se
    fim se
  fim se
fim para
```

A aplicação de seleção P5 percorre todo o arquivo D5 em busca de certos tipos de regras de transdução que caracterizam regras de simplificação em nível sintático. São 3 os tipos de regras buscadas por P5:

- **Regras de Segmentação de Sentenças Longas:** São caracterizadas por transformar uma sentença complexa em duas ou mais sentenças mais curtas de significado equivalente. Sua detecção pela aplicação em P5 é feita pela contagem de nós folhas de classe gramatical "." (ponto final) presentes nas árvores fonte e alvo da regra de transdução. Se a árvore fonte possuir menos nós folha de classe gramatical "." do que a árvore alvo, então a regra de transdução é uma transformação de segmentação e deve ser adicionada ao arquivo D6.

- **Regras de Resolução de Sentenças com Conectores:** Estas regras de simplificação são caracterizadas por re-estruturar sentenças complexas que possuem segmentos interligados por conectores da língua inglesa, como "*and*" e "*or*". A aplicação P5 identifica estas regras avaliando se a árvore fonte da regra possui nós folha cujo nó pai é de valor "CC". O valor "CC" representa a classe gramatical das conjunções conectoras na representação sintática descrita em [Marcus, Marcinkiewicz e Santorini 1993].

- **Regras de Reordenação de Segmentos da Sentença:** Este tipo de regra de simplificação é caracterizado por reordenar segmentos que compõe a sentença. Regras deste tipo podem, em alguns casos, caracterizar operações de transferência de voz passiva para voz ativa. Estas regras são identificadas por P5 pela comparação entre a ordem de ocorrência das variáveis que compõe os nós folha das árvore fonte e alvo da regra. A busca por este tipo de regra acusa resultado positivo apenas se a ordem de ocorrência entre as variáveis da árvore fonte e da árvore alvo não forem idênticas.

Empregando uma estratégia similar à vista em P5, a aplicação em Java P6, descrita no Algoritmo 4, tem como função extrair e armazenar no arquivo D7 somente as regras de simplificação em nível lexical encontradas no arquivo D5.

Algoritmo 4 Algoritmo de seleção de regras de simplificação em nível lexical

$RegrasT3$ = Abrir arquivo $D5$
$D7$ = Abrir novo arquivo

para cada regra R de $RegrasT3$ **faça**
$Arvores$ = Compile a expressão regular "([^\t]+)\t([^\n]+)" sobre R
$ArvEsquerda$ = $Arvores[0]$
$ArvDireita$ = $Arvores[1]$
$FolhasEsquerda$ = Compile a expressão regular "\\(([\^\s]+) [\^\\(\\)]+\\)" sobre $ArvEsquerda$
$FolhasDireita$ = Compile a expressão regular "\\(([\^\s]+) [\^\\(\\)]+\\)" sobre $ArvDireita$
$FolhasSemVariaveis$ = Compile a expressão regular "\\([^\s]+ [^#\\(\\)]+[^\\)]*\\)" sobre R

se Tamanho de $FolhasSemVariaveis$ igual a 0 e $ArvDireita$ diferente de $NULL$ **então**
$FraseComplexa$ = Extraia e concatene as palavras de $FolhasEsquerda$
$FraseSimples$ = Extraia e concatene as palavras de $FolhasDireita$
$NucleoComplexa$ = Extraia núcleo morfológico de $FraseComplexa$
$NucleoSimples$ = Extraia núcleo morfológico de $FraseComplexa$
$RaizEsquerda$ = Extraia a raiz de $ArvEsquerda$
$RaizDireita$ = Extraia a raiz de $ArvDireita$

se $RaizEsquerda$ igual a $RaizDireita$ **então**
se $FraseComplexa$ não faz parte de $FraseSimples$ e vice-versa **então**
se $NucleoComplexa$ diferente de $NucleoSimples$ **então**
se $FraseComplexa$ e $FraseSimples$ não contém $Numeral$ ou $NomePrprio$ **então**
Escreva R em $D7$
fim se
fim se
fim se
fim se
fim se
fim para

No sistema proposto, as regras de simplificação em nível lexical são empregadas na tarefa de substituição de termos complexos por equivalentes simples. A estratégia de substituição de termos complexos é empregada nos trabalhos de [Carroll et al. 1998], [Keskisarkka 2012] e [Kandula, Curtis e Zeng-Treitler 2010], onde provou produzir bons resultados de simplificação em nível lexical. Na execução da aplicação P6 é percorrida toda a extensão do arquivo D5 em busca de regras de transdução que preencham os seguintes requisitos:

1. **Os nós de folha de ambas árvores complexa e simples devem ser compostos apenas por palavras:** Regras de transdução com árvores fonte ou alvo que possuam variáveis em seus nós folha tendem a modificar grandes porções do texto. Este tipo de regra de transdução comumente caracteriza uma regra de simplificação em nível sintático.

2. **A árvore fonte não pode estar alinhada a uma árvore alvo nula:** Regras desta natureza caracterizam a operação de remoção de segmento não-essencial, que é uma operação de simplificação em nível sintático não abordada neste trabalho.

3. **As classes gramaticais dos nós raiz das árvores alvo e base devem ser idênticas:** Caso as árvores fonte e alvo não sejam da mesma classe gramatical, a substituição de uma árvore pela outra em uma sentença complexa pode causar resultados de simplificação estranhos, como a substituição de um verbo por um substantivo ou vice-versa.

4. **A árvore alvo não pode ser subárvore da árvore fonte:** A frase constituída pelos nós folha da estrutura da árvore alvo não podem ser um segmento da frase constituída pelos nós folha da árvore fonte. Regras que não atendem a este requisito devem ser descartadas pois quando aplicadas a uma sentença complexa, sua versão simplificada tem grandes chances de continuar com termos complexos em sua estrutura.

5. **A árvore fonte não pode ser subárvore da árvore alvo:** A frase constituída pelos nós folha da estrutura da árvore fonte não pode ser um segmento da frase da árvore alvo. Este tipo de regra de transdução implica que a totalidade das palavras complexas nas folhas da árvore fonte estarão contidas também nas folhas da árvore alvo. Quando aplicada a uma sentença complexa, uma regra de simplificação desse tipo não substitui os termos complexos contidos na mesma.

6. **As árvores da regra não podem conter nós folha com numerais, nomes próprios ou siglas:** Regras de transdução com estas palavras em nós folha são aplicáveis apenas em contextos isolados. Estas regras de transdução caracterizam regras de simplificação pouco genéricas, e portanto não devem ser adicionadas ao arquivo D7.

7. **As palavras da árvore fonte não podem ter o mesmo núcleo morfológico que as palavras da árvore alvo:** O núcleo morfológico de uma palavra é o termo que representa

sua forma não flexionada. O termo "*run*", por exemplo, é o núcleo morfológico de suas formas flexionadas "*ran*" e "*running*". Regras que não atendem a este requisito podem causar problemas similares aos causados pelas regras onde a árvore fonte é uma subárvore da árvore alvo ou vice-versa, pois quando aplicadas tendem a manter termos complexos nas versões simplificadas. Este tipo de regra pode também comprometer a gramaticalidade da versão simplificada: considere por exemplo substituir a palavra "*running*" na frase "*The boy is running from the dog*" pela palavra "*ran*", que tem o mesmo núcleo morfológico. A frase resultante desta transformação é "*The boy is ran from the dog*", que possui um erro de concordância no tempo verbal de suas palavras. Na aplicação P6, o sistema utilizado na extração do núcleo morfológico das palavras das árvores fonte e alvo é o Stanford CoreNLP.

A Figura 5.5 ilustra exemplos da forma assumida pelas regras de simplificação em nível sintático e lexical encontradas nos arquivos D6 e D7, respectivamente.

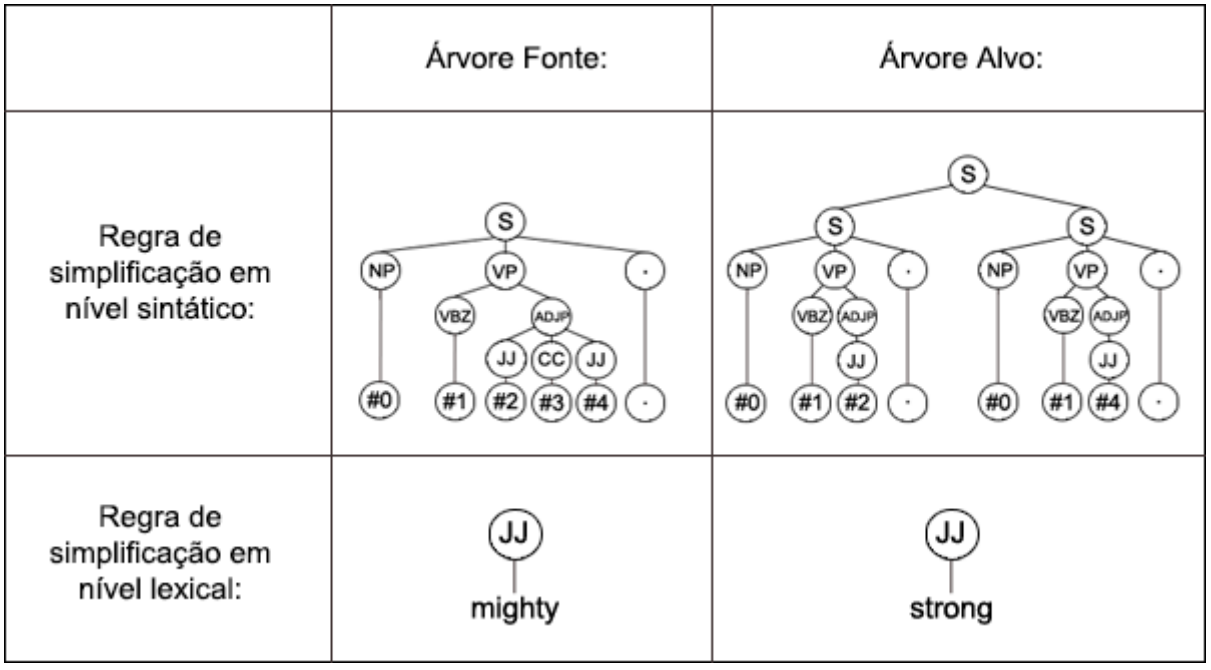


Figura 5.5: Regras de simplificação em nível sintático e lexical

Observa-se que o exemplo de regra de simplificação em nível sintático da Figura 5.5 possui árvores fonte e alvo de grande profundidade com muitas variáveis. Já o exemplo de regra de simplificação em nível lexical da mesma Figura possui árvores fonte e alvo de baixa profundidade, contendo apenas palavras em seus nós folha. Estas diferenças entre os dois tipos de regra se dá pela distinção no resultado de simplificação que é esperado na aplicação das mesmas.

A finalização das aplicações em Java ativadas pelos processos P5 e P6 marcam o fim da etapa de treinamento do sistema. Se não acusados erros durante nenhum de seus processos, o Módulo M1 de Treinamento produz os documentos D6 e D7 de regras de simplificação em nível sintático e lexical, respectivamente. Estes arquivos fazem parte dos dados de entrada do Módulo M2 de Simplificação, e são utilizados pelo mesmo para confeccionar as versões simples de uma dada sentença complexa.

5.2 Função, Especificação e Programação do Módulo de Simplificação (M2)

O Módulo M2 de Simplificação tem a função de, por meio do uso das regras de simplificação produzidas pelo Módulo M1, produzir múltiplas versões simplificadas de uma sentença complexa passada como entrada. Este Módulo é, assim como o M1, constituído por um *script* em Python cuja execução é dividida em 3 etapas: Pré-Processamento, Simplificação em Nível Sintático e Simplificação em Nível Lexical. A Figura 5.6 e o Algoritmo 5 ilustram o fluxo de execução dos componentes do Módulo M2 de Simplificação.

Algoritmo 5 *Script* do Módulo de Simplificação (M2)

Receba a Sentença Complexa e os arquivos $D6$ e $D7$ de entrada
$D8$ = Produza a árvore sintática da Sentença Complexa pelo $Stanford\ Parser$
$D9$ = Utilize as regras de simplificação em $D6$ para produzir versões simplificadas em nível sintático da árvore sintática em $D2$ pela unidade $U1$
$D10$ = Utilize as regras de simplificação em $D7$ para produzir versões simplificadas em nível lexical das árvores sintáticas em $D9$ pela unidade $U2$
Apresente como saída o arquivo $D10$

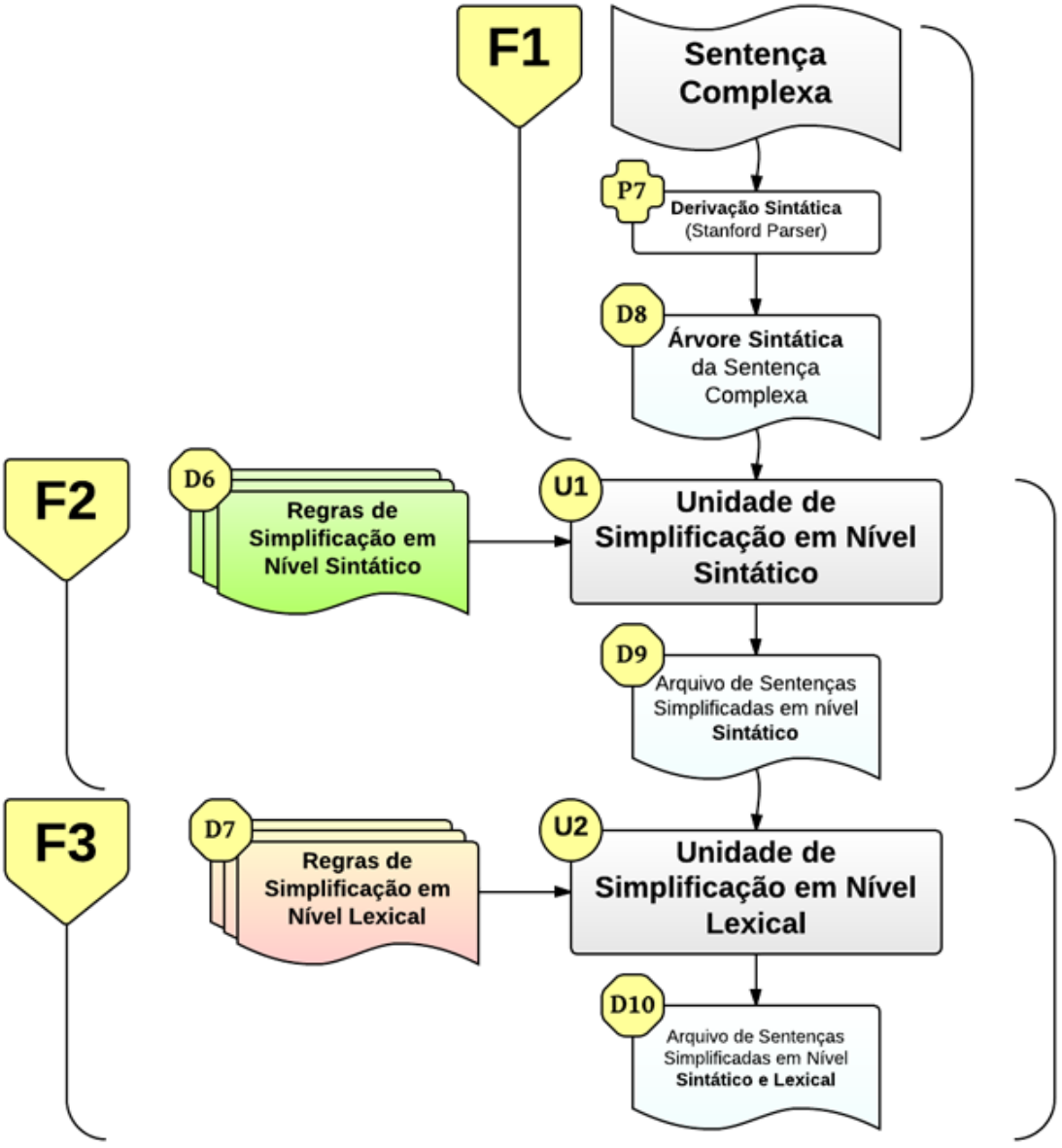


Figura 5.6: Fluxograma de execução do Módulo de Simplificação (M2)

Como entrada, o Módulo recebe uma sentença complexa e os arquivos com regras de simplificação em nível sintático e lexical. Como saída, é produzido um arquivo com diferentes versões simplificadas em nível sintático e lexical da sentença complexa.

Na Figura 5.6 são encontrados os seguintes componentes:

- **Sentença Complexa:** Sentença complexa escrita na língua inglesa a ser simplificada.
- **P7:** Chamada ao sistema de análise sintática, cujo objetivo é confeccionar a árvore sintática da sentença complexa de entrada.
- **D8:** Arquivo produzido ao final de P7, que contém apenas a árvore sintática da sentença complexa de entrada.
- **D6:** Arquivos de regras de simplificação em nível sintático produzidas pelo Módulo M1 de Treinamento.
- **U1:** Aplicação em Java desenvolvida especificamente para este projeto. Sua função é confeccionar, pela transdução de árvores, versões simplificadas em nível sintático da sentença complexa de entrada.
- **D9:** Arquivo que contém os candidatos simplificados em nível sintático produzidos ao final da execução de U1.
- **D7:** Arquivos de regras de simplificação em nível lexical produzidas pelo Módulo M1 de Treinamento.
- **U2:** Aplicação em Python desenvolvida especificamente para este projeto. Sua função é confeccionar, pela transdução de árvores, novas versões simplificadas em nível lexical das sentenças candidatas simplificadas em D9.
- **D10:** Arquivo que contém os candidatos simplificados em nível sintático e lexical produzidos ao final da execução de U1.

Os componentes da Figura 5.6 identificados por U1 e U2 representam as duas unidades de simplificação do Módulo: a unidade de simplificação em nível sintático e a unidade de simplificação em nível lexical. Estes são os dois principais componentes do Módulo de Simplificação.

U1 e U2 transduzem a árvore sintática de uma sentença complexa em múltiplas árvores sintáticas mais simples.

As três etapas do Módulo M2, que são executadas em sequência, são:

1. **Pré-processamento (F1):** Nessa fase é recebida uma sentença complexa, localizada ao topo da Figura 5.6, que é então direcionada a um sistema de análise sintática (P7), que produz como saída um arquivo contendo a árvore sintática da árvore complexa (D8).

2. **Simplificação em Nível Sintático (F2):** Fase onde são aplicadas regras de simplificação em nível sintático à sentença complexa de entrada. Esta fase utiliza as regras de simplificação do documento D6 para produzir como saída um arquivo com múltiplos candidatos a versão simplificada da sentença complexa de entrada (D9).

3. **Simplificação em Nível Lexical (F3):** Fase onde são aplicadas regras de simplificação em nível lexical às sentenças simplificadas do arquivo D9. Esta fase utiliza as regras de simplificação do documento D7 para produzir como saída novos candidatos, agora simplificados em ambos níveis sintático e lexical (D10).

5.2.1 A Fase de Pré-processamento (F1)

Nesta fase, o Módulo de Simplificação é encarregado de confeccionar a árvore sintática da sentença complexa de entrada. Como ilustrado na Tabela 5.4, as regras de simplificação sintáticas e lexicais do sistema são representadas por uma árvore sintática base complexa alinhada a uma árvore sintática alvo simples. Para que as regras de simplificação possam ser empregadas, é necessário que se confeccione a árvore sintática da sentença complexa de entrada.

Esta necessidade surge pois, nas fases seguintes do Módulo M2, as unidades de simplificação de índice U1 e U2 da Figura 5.6 buscam regras cuja árvore sintática base é compatível com a árvore sintática da sentença complexa, e então transformam a árvore sintática da sentença complexa utilizando a árvore sintática alvo da regra. Sem a árvore sintática da sentença complexa seria impossível para as unidades de simplificação U1 e U2 verificar quais regras de simplificação têm a árvore fonte compatível a mesma, impedindo que o sistema aplique qualquer tipo de simplificação à sentença complexa.

Para confeccionar a árvore sintática da sentença complexa, o sistema de análise sintática Stanford Parser é empregado. O processo P7 faz uma chamada ao sistema Stanford Parser, passando a sentença complexa a ser simplificada como entrada. Ao seu final, o sistema de análise sintática produz o arquivo D8, que contém a árvore sintática da sentença complexa. O arquivo D8 é então direcionado à fase de simplificação em nível sintático.

5.2.2 A Fase de Simplificação em Nível Sintático (F2)

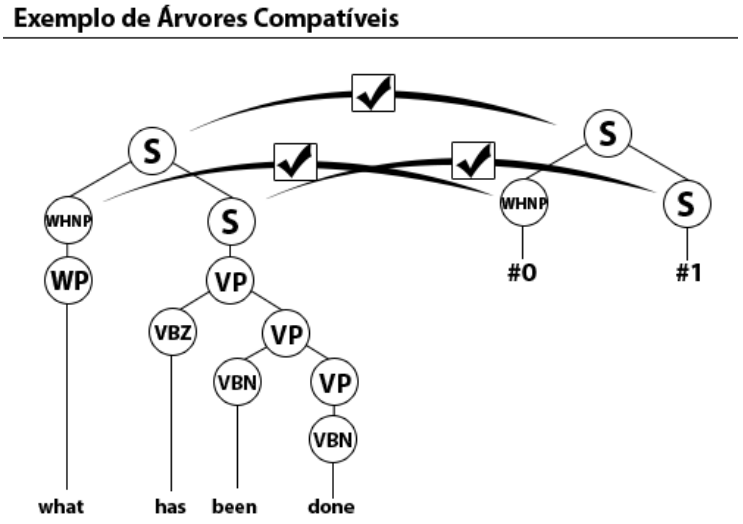
O objetivo da fase F2 de Simplificação em Nível Sintático do Módulo M2 é, pelas regras de simplificação do arquivo D6, confeccionar um arquivo com múltiplas versões distintas simplificadas em nível sintático da sentença complexa de entrada.

Esta fase recebe como entrada as regras de simplificação em nível sintático (D6), e a árvore sintática da sentença complexa (D8). Seu principal componente, é a unidade U1. O Algoritmo 6 descreve seu processamento.

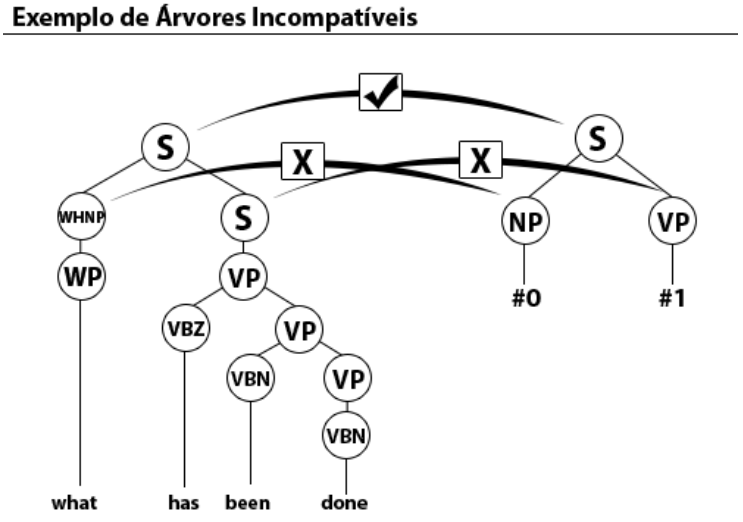
Algoritmo 6 Algoritmo de Transdução de Árvores da Unidade U1

$Regras$ = Abrir arquivo $D6$
$DadosSintaticos$ = Abrir arquivo $D8$
$D9$ = Abrir novo arquivo
$Sentenca$ = Ler árvore sintática de $DadosSintaticos$
$ArvSentenca$ = Transforme $Sentenca$ em estrutura de dados
para cada regra R em $Regras$ **faça**
  $Arvores$ = Compile a expressão regular "([^\t]+)\t([^\n]+)" sobre R
  $ArvFonteEmTexto$ = $Arvores[0]$
  $ArvAlvoEmTexto$ = $Arvores[1]$
  $ArvFonte$ = Transforme $ArvFonteEmTexto$ em estrutura de dados
  $ArvAlvo$ = Transforme $ArvAlvoEmTexto$ em estrutura de dados
  se $ArvFonte$ é compatível com $ArvSentenca$ em algum nó N **então**
    $Dicionario$ = Encontre subárvores de $ArvSentenca$ correspondentes a cada variável de $ArvFonte$
    Substitua variáveis de $ArvAlvo$ pelas subárvores de $Dicionario$
    $ArvSimplificada$ = Encaixe $ArvAlvo$ no nó N de $ArvSentenca$
    $ArvSimplificadaEmTexto$ = Transforme $ArvSimplificada$ em formato textual
    Adicione $ArvSimplificadaEmTexto$ a $D9$
  fim se
fim para

O primeiro passo executado pela unidade U1 é a transformação da árvore sintática em formato textual da sentença complexa em uma estrutura de dados de árvore. Em seguida, para cada regra de simplificação em D6, a unidade U1 transforma também as representações textuais de suas árvores fonte e alvo em duas estruturas de dados de árvore. A unidade U1 então verifica se a árvore fonte da regra pode ser encontrada em alguma das subárvores da estrutura sintática da sentença complexa. As Figuras 5.7(a) e 5.7(b) ilustram casos de falha e sucesso no processo de busca de compatibilidade entre a árvore fonte de uma regra e a árvore sintática de uma sentença complexa.



(a) Árvores sintáticas compatíveis



(b) Árvores sintáticas incompatíveis

Figura 5.7: Casos de compatibilidade e incompatibilidade entre pares de árvores sintáticas

Se encontrada a árvore fonte na árvore da sentença complexa, a unidade produz então um dicionário chave-valor, cujas chaves são as variáveis presentes na árvore fonte da regra, e os valores são as referências de memória às subárvores da sentença complexa correspondentes às variáveis da árvore fonte. A Figura 5.8 ilustra um exemplo de dicionário produzido a partir das árvores compatíveis da Figura 5.7(a).

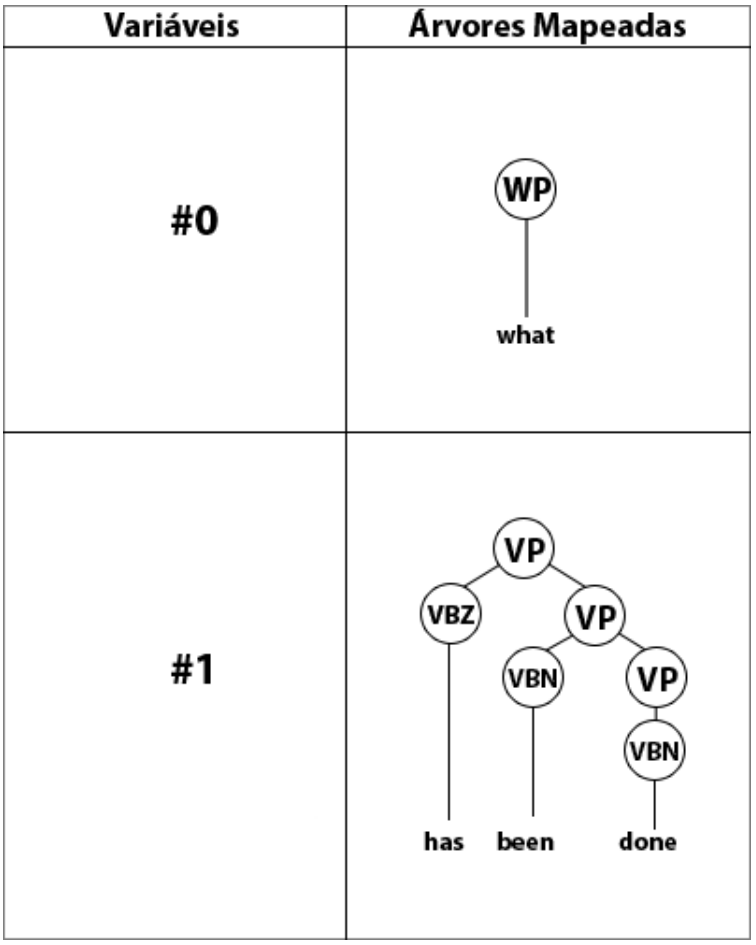


Figura 5.8: Dicionário de correspondências entre variáveis e subárvores

A unidade então utiliza o dicionário para substituir as variáveis da árvore alvo da regra pelas subárvores da sentença complexa correspondentes, assim como ilustrado na Figura 5.9. Em seguida, a árvore alvo livre de variáveis é transferida à árvore sintática da sentença complexa na mesma posição onde foi encontrada a árvore fonte no processo de verificação. Desta forma, a unidade U1 modifica a estrutura sintática da sentença complexa, criando uma árvore sintática simplificada.

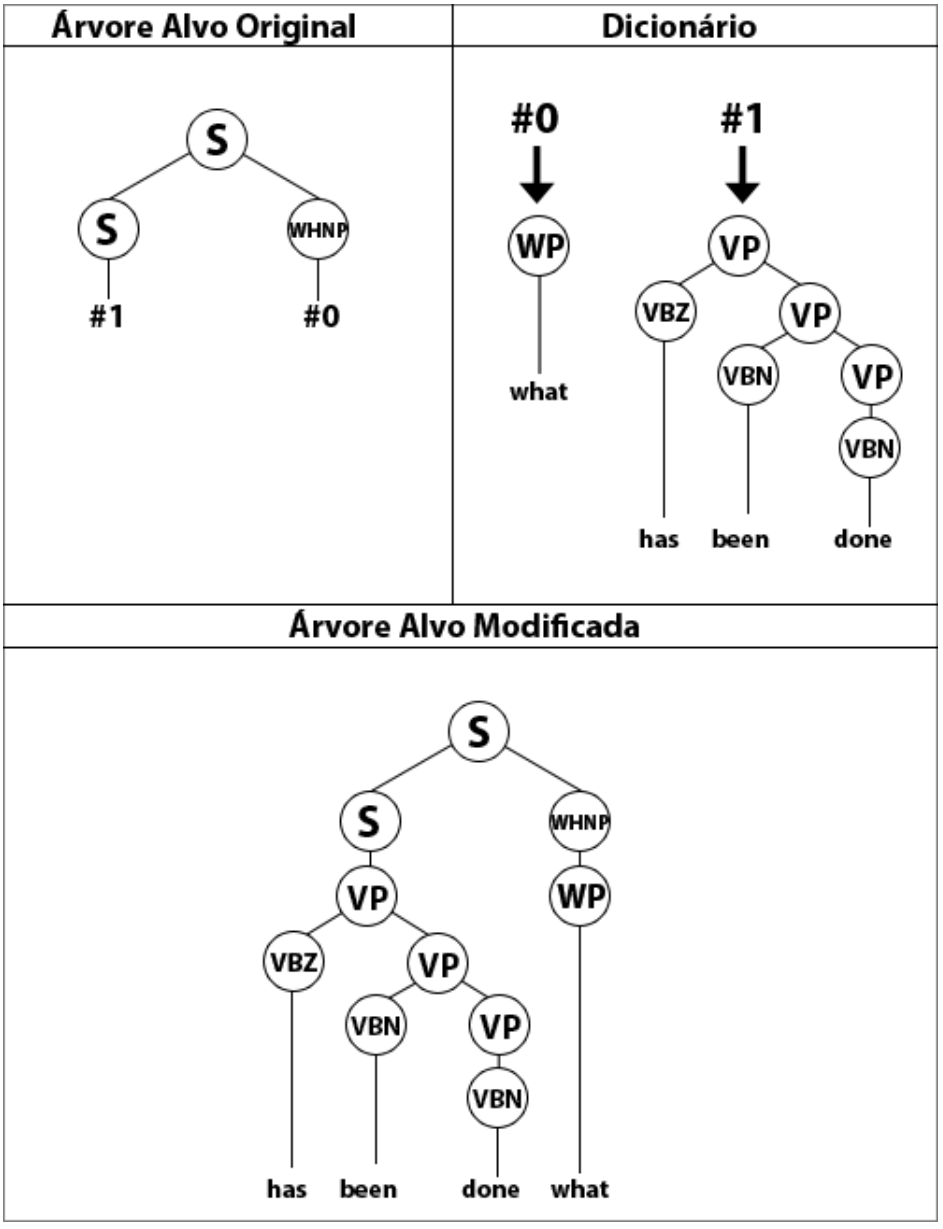


Figura 5.9: Processo de substituição de variáveis em uma árvore alvo de uma regra

Finalmente, a unidade U1 transforma a estrutura de dados da árvore simplificada resultante em formato textual, e a armazena no arquivo D9, que possuirá as árvores sintáticas simplificadas produzidas por cada uma das regras do arquivo D6. O formato do arquivo D9 é ilustrado na Tabela 5.5.

Árvore Sintática Simplificada
(S (NP (PRP He)) (VP (VBD was) (NP (DT a) (ADJP reckless) (NN sovereign))) (. .))
(S (NP (PRP He)) (VP (VBD was) (NP (DT a) (ADJP reckless))) (. .))
(S (NP (PRP He)) (VP (VBD was) (ADJP reckless)) (. .))
(S (NP (PRP He)) (VP (ADJP reckless) (VBD was)) (. .))

Tabela 5.5: Formato do arquivo D9 produzido ao final da execução da unidade U1

5.2.3 A Fase de Simplificação em Nível lexical (F3)

A fase F3 do Módulo M2 é responsável pelo processo de simplificação em nível lexical das versões simplificadas produzidas na fase anterior.

Na fase F3 são recebidos os arquivos D7, contendo regras de simplificação em nível lexical produzidas pelo Módulo M1 de Treinamento, e D9, que contém sentenças simplificadas produzidas na fase F2 de simplificação em nível sintático. Estes dois arquivos são então direcionados ao principal componente da fase F3, que é a unidade U2 de simplificação em nível lexical. A unidade U2 é uma aplicação desenvolvida em Python. Seu processamento é descrito no Algoritmo 7.

Primeiramente, a unidade U2 lê o arquivo D7 e o transforma em uma estrutura de dados de dicionário. Este dicionário mapeia cada uma das árvores sintáticas de termos complexos do arquivo D7 às árvores sintáticas de seus termos simples equivalentes. Este processo é ilustrado na Figura 5.10.

Algoritmo 7 Algoritmo de Transdução de Árvores da Unidade U2

$Regras$ = Abrir arquivo $D7$
$Candidatos$ = Abrir arquivo $D9$
$D10$ = Abrir novo arquivo
$Dicionario$ = Criar novo dicionario

para cada regra R de $Regras$ **faça**
  $ArvBase$ = Extraia árvore fonte de R
  $ArvAlvo$ = Extraia árvore alvo de R
  Adicione $ArvAlvo$ à lista $Dicionario$[$ArvBase$]
fim para

para cada candidato C em $Candidatos$ **faça**
  $Pilha$ = Nova pilha
  $ListaDeControle$ = Nova lista
  Adicione C a $Pilha$, $ListaDeControle$ e $D10$

  enquanto $Pilha$ não estiver vazia **faça**
    $CandidatoAtual$ = Retire elemento do topo de $Pilha$
    $Subarvores$ = Identifique subárvores de $CandidatoAtual$

    para cada subárvore S em $Subarvores$ **faça**
      se existe $Dicionario$[S] **então**
        para cada árvore simplificada $ArvSimples$ de $Dicionario$[S] **faça**
          $NovaCandidata$ = Substitua S por $ArvSimples$ em $CandidatoAtual$
          se $NovaCandidata$ não estiver em $ListaDeControle$ **então**
            Adicione $NovaCandidata$ a $D10$
            Adicione $NovaCandidata$ a $Pilha$
            Adicione $NovaCandidata$ a $ListaDeControle$
          fim se
        fim para
      fim se
    fim para
  fim enquanto
fim para

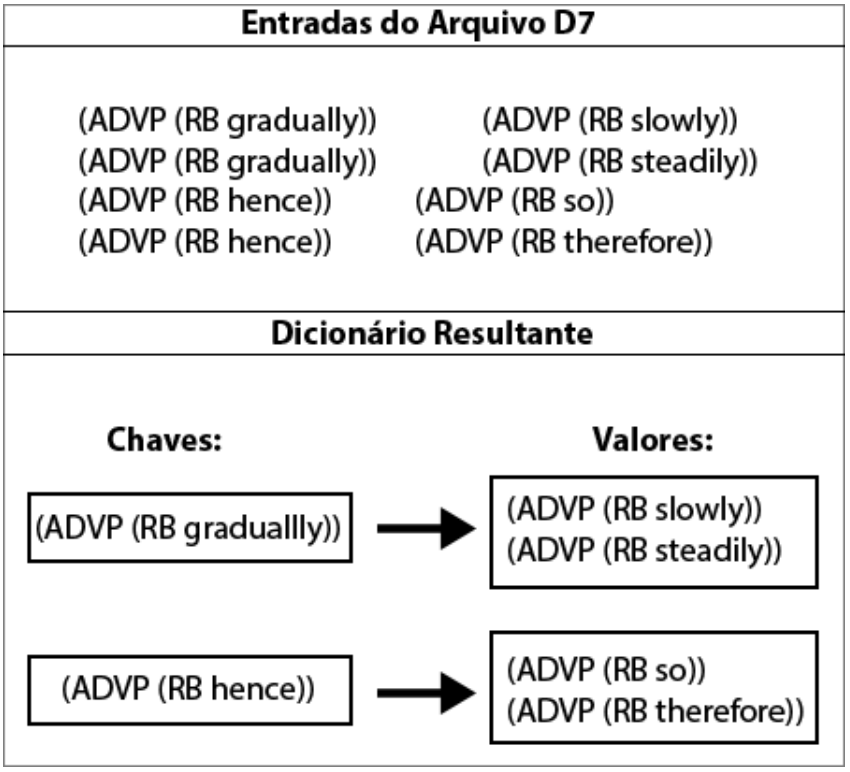


Figura 5.10: Entradas do arquivo D7 e sua estrutura de dicionário equivalente

São então produzidas novas versões simplificadas para cada uma das sentenças em D9 a partir dos dados do dicionário confeccionado anteriormente. Para cada árvore sintática em D9, a unidade U2 primeiramente identifica todas as subárvores de sua estrutura e as armazena em uma lista, assim como mostrado na Figura 5.11. É importante ressaltar que as árvores sintáticas do arquivo D9 não são transformadas em estruturas de dados de árvore, e portanto todo o processamento da unidade U2 é feito sobre árvores ainda em sua forma textual.

Uma vez identificadas as subárvores da estrutura sintática de uma dada sentença, a unidade U2 as compara às entradas do dicionário, e então produz novas versões simplificadas resultantes de todas as combinações da aplicação, ou não, das regras de simplificação do dicionário cuja árvore sintática do termo complexo é idêntica a alguma das subárvores identificadas. São então extraídos as palavras dos nós folha de cada árvore sintática, e assim confeccionadas as novas sentenças simplificadas. As sentenças são então armazenadas no arquivo D10, cujo formato é ilustrado na Tabela 5.6.

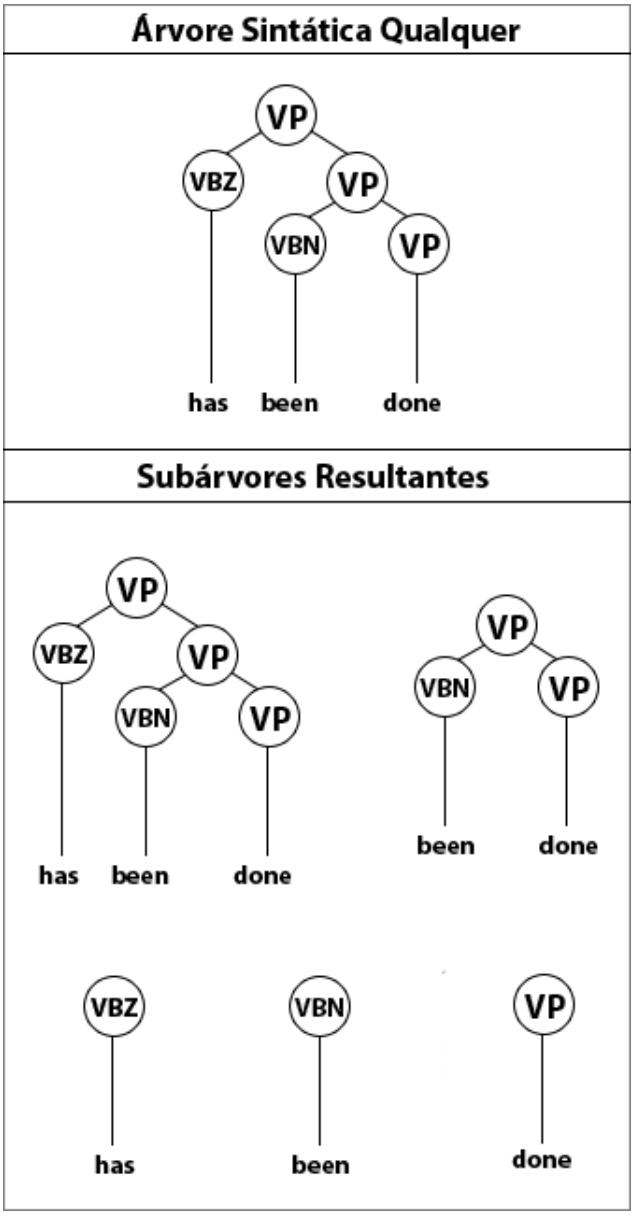


Figura 5.11: Árvore sintática e suas subárvores

Sentença Simplificada
He was a reckless king .
He was a reckless sovereign .
He was a reckless monarch .
He was a vicious king .
He was a vicious sovereign .
He was a vicious monarch .

Tabela 5.6: Versões simplificadas de uma sentença complexa

Finalmente, o arquivo D10 é direcionado ao Módulo M3 de Ranqueamento para que o mesmo produza a versão simplificada final da sentença complexa de entrada do Módulo M2.

5.3 Função, Especificação e Programação do Módulo de Ranqueamento (M3)

O ranqueamento de sentenças é a última etapa do sistema proposto. Sua função é pontuar um conjunto de sentenças simplificadas. Em cada caso, a sentença que obtiver a maior pontuação dentre as outras candidatas é selecionada. O Módulo M3 recebe como entrada o arquivo de sentenças simplificadas em nível sintático e lexical, e produz como saída uma única sentença, que representa a sentença com maior qualidade de simplificação presente no arquivo.

A codificação desse Módulo, assim como em outros Módulos, foi desenvolvida na forma de *script* na linguagem Python, a partir do qual são chamadas não só funções prontas da linguagem, mas também o sistema SRILM, que constrói modelos de linguagem. O Algoritmo 8 descreve o *script* em Python do Módulo M3, ilustrado no fluxograma da Figura 5.12.

Algoritmo 8 *Script* do Módulo de Ranqueamento (M3)

Receba o arquivo com sentenças simplificadas em nível sintático e lexical $D10$
$D11$ = Produza as medidas de perplexidade das sentenças em $D10$ pelo $SRILM$
$D12$ = Calcule a pontuação das sentenças em $D10$ a partir das medidas em $D11$
Ordene as sentenças de $D12$ em ordem decrescente de pontuação
Apresente como saída a sentença de maior pontuação de $D12$

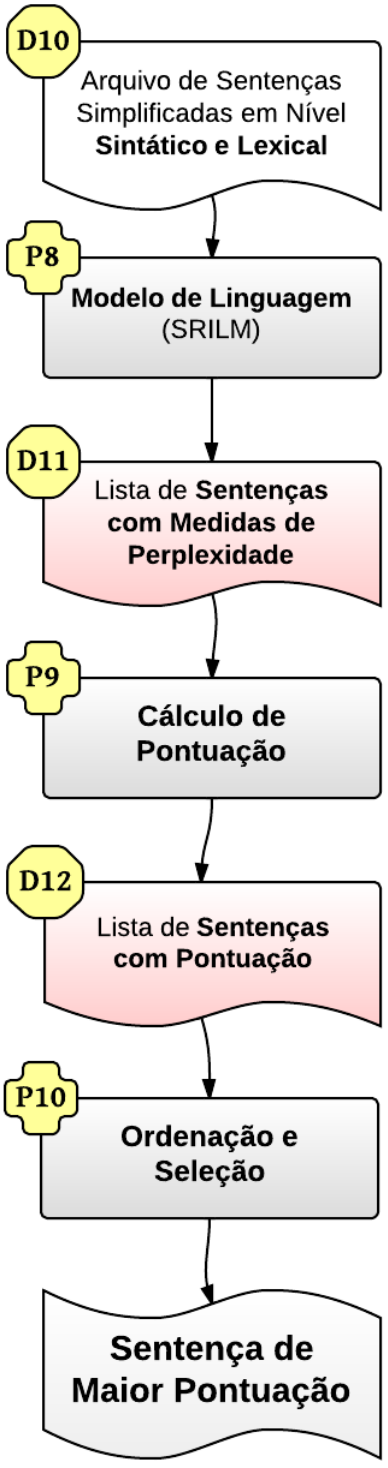


Figura 5.12: Fluxo de execução do Módulo M3 de Ranqueamento

Como ilustrado na Figura 5.12, o fluxo de execução do Módulo M3 de Ranqueamento é constituído por 7 (sete) componentes:

- **D10:** Arquivo contendo as sentenças simplificadas em nível sintático e lexical a serem ranqueadas.
- **P8:** Chamada ao sistema externo SRILM, cujo objetivo é atribuir medidas de perplexidade a cada sentença do arquivo D10.
- **D11:** Arquivo produzido ao final de P8, contendo as sentenças simplificadas do arquivo D10, porém com suas respectivas medidas de perplexidade.
- **P9:** Função em Python desenvolvida especificamente para este trabalho, cujo objetivo é calcular a medida de pontuação referente a cada sentença do arquivo D11.
- **D12:** Arquivo produzido ao final de P10, contendo as mesmas sentenças do arquivo D11, porém com suas respectivas pontuações.
- **P10:** Função em Python desenvolvida especificamente para este trabalho, cujo objetivo é ordenar as sentenças do arquivo D12 em ordem decrescente de pontuação, e então selecionar a sentença de maior pontuação dentre todas.
- **Sentença Simplificada:** Sentença de maior pontuação selecionada pelo processo P10. Esta sentença é a versão simplificada final da sentença complexa passada como entrada ao Módulo M2 de Simplificação.

O Módulo M3 primeiramente recebe o arquivo de entrada D10. Este arquivo deve ser composto por uma ou mais sentenças armazenadas uma em cada linha. O arquivo D10 é então passado como entrada a P8, que chama o sistema SRILM, cuja função é gerenciar um modelo de linguagem da língua inglesa.

O sistema SRILM, descrito na Seção 3.5, gerencia um modelo de linguagem treinado a partir de um corpus com sentenças simples da língua inglesa. Este modelo de linguagem é utilizado para estimar as medidas de perplexidade para cada uma das sentenças do arquivo D10. A medida de perplexidade representa uma aproximação de quão frequentemente uma sentença

é utilizada na língua em que está escrita. A Equação 5.1 descreve o cálculo da medida de perplexidade de uma sentença S.

$$Perplexidade(S) = 10^{(-logprob(S)/(Palavras-PD+CountSentencas))} \tag{5.1}$$

Os elementos da Equação de perplexidade 5.1 são descritos por [Stolcke 2007] da seguinte forma:

- $-logprob(S)$**:** Probabilidade logarítmica negativa da sentença S com respeito às sentenças simples utilizadas na etapa de treinamento. Primeiramente é calculada a probabilidade $P(S)$ pelos N-gramas do modelo de linguagem, assim como descrito na Seção 3.5. Em seguida, é calculado o valor $-log(P(S))$, que caracteriza a medida final de $-logprob(S)$.
- $Palavras$**:** Soma da quantidade de palavras presentes nas sentenças do arquivo D10.
- PD**:** Número de palavras das sentenças do arquivo D10 não encontradas em nenhuma das 711 mil sentenças simples do corpus de treinamento.
- $CountSentencas$**:** Número de sentenças presentes no arquivo D10.

Ao final da chamada P8 é produzida a lista de índice D11, cujo formato é ilustrado na Tabela 5.7. A lista D11 contém cada sentença do arquivo D10 com sua respectiva medida de perplexidade calculada pelo modelo de linguagem SRILM.

Sentença	**Medida de Perplexidade**
He was a reckless king .	214.085
He was a reckless sovereign .	715.327
He was a reckless monarch .	622.201
He was a vicious king .	110.087
He was a vicious sovereign .	271.192
He was a vicious monarch .	389.133

Tabela 5.7: Formato da lista D11 produzido ao final da chamada P8

Finalizada a chamada P8, a lista D11 é então direcionada à função em Python de índice P9. A função P9 é responsável por ler os dados da lista e calcular a pontuação final de cada

sentença, armazenando os resultados obtidos em uma nova lista de índice D12. A Equação 5.2 descreve o cálculo da pontuação final de cada sentença.

$$Pontuacao(S) = \frac{1}{Perplexidade(S)} \tag{5.2}$$

O valor de pontuação é inversamente proporcional à medida de perplexidade da sentença. A pontuação é calculada desta forma pois quanto menor sua medida de perplexidade, maior é a familiaridade da sentença em relação às 711 mil sentenças simples utilizadas no treinamento do modelo de linguagem. Em outras palavras, quanto menor a perplexidade, maior a simplicidade da sentença. A Tabela 5.8 exibe a lista produzida pela função P9 a partir dos dados apresentados na Tabela 5.7.

Sentença	**Pontuação Final**
He was a reckless king .	0.004671
He was a reckless sovereign .	0.001397
He was a reckless monarch .	0.001607
He was a vicious king .	0.009083
He was a vicious sovereign .	0.003687
He was a vicious monarch .	0.002569

Tabela 5.8: Formato da lista D12 produzido ao final da chamada P9

Uma vez finalizada a execução da função P9, a lista D12 é então direcionada à função em Python P10 de ordenação e seleção. Esta função lê as sentenças da lista D12, organiza-as em ordem decrescente de pontuação, e então exibe a sentença no primeiro índice da lista como saída do Módulo.

A conclusão da execução do Módulo de Ranqueamento produz uma versão simplificada final da sentença complexa passada como entrada ao Módulo de Simplificação. Estas sentenças simplificadas podem então ser submetidas a processos de avaliação, os quais permitem que sejam construídas estatísticas sobre o desempenho do sistema proposto.

Capítulo 6

Experimentos

Dada a proposta desse trabalho, de apresentar uma ferramenta de alinhamento e simplificação de textos, testes foram realizados e os resultados obtidos pela aplicação do sistema proposto em três diferentes experimentos são apresentados.

- **Avaliação de Desempenho Geral:** Neste experimento de caráter quantitativo e qualitativo, são utilizados métodos automáticos de avaliação de sistemas de tradução sobre os resultados produzidos pelo sistema de simplificação proposto. O sistema de simplificação é empregado na simplificação de sentenças complexas em duas configurações: uma que aplica a etapa de simplificação em nível sintático após a etapa de simplificação em nível lexical, e outra que aplica a simplificação em nível lexical após a simplificação em nível sintático. O objetivo deste experimento é avaliar não só quão eficiente é o sistema proposto na simplificação de sentenças complexas, mas também descobrir se existe alguma diferença significativa de desempenho entre as duas possíveis configurações. Detalhes com respeito aos métodos automáticos de avaliação escolhidos são descritos na Seção 6.1.

- **Avaliação de Desempenho de Componentes:** Experimento de caráter quantitativo e qualitativo, cujo objetivo é avaliar, de forma individual, a eficiência dos componentes que realizam as etapas de simplificação em nível sintático e lexical no sistema proposto. Neste experimento, o sistema proposto é empregado na simplificação de múltiplas sentenças complexas em nível sintático e lexical separadamente, para que a qualidade das versões simplificadas possa ser avaliada com respeito à sua simplicidade, gramaticalidade e coerência. Os resultados obtidos poderão ser utilizados como parâmetro em trabalhos futuros

de incrementação e/ou reestruturação das técnicas e métodos utilizados pelo sistema.

- **Avaliação de Métricas Alternativas de Ranqueamento:** Experimento também de caráter quantitativo e qualitativo, cujo objetivo é avaliar a eficiência de possíveis incrementos ao Módulo de Ranqueamento do sistema proposto. Neste experimento são aplicadas algumas das métricas apresentadas em [Graesser et al. 2004] no ranqueamento de sentenças complexas, para que seu desempenho possa então ser comparado a ranqueamentos ótimos confeccionados manualmente. Espera-se que os resultados obtidos neste experimento auxiliem na decisão de quais as métricas mais adequadas na incrementação da métrica de pontuação aplicada pelo Módulo de Ranqueamento do sistema.

A configuração de alguns aspectos do sistema de simplificação adotada para a condução dos três experimentos é disposta da seguinte forma:

- **Corpus paralelo:** Assim como relatado na Seção 5.1, o Módulo de Treinamento recebe como entrada um corpus paralelo de sentenças complexas alinhadas a sentenças simples equivalentes. O corpus paralelo utilizado nos experimentos foi confeccionado automaticamente pela técnica apresentada em [Coster e Kauchak 2011]. É composto por 137362 sentenças complexas extraídas da Wikipedia, alinhadas a 137362 sentenças equivalentes simples extraídas da Simple Wikipedia.

- **Modelo de linguagem:** O Módulo de Ranqueamento, assim como descrito na Seção 5.3, utiliza o sistema SRILM, que aplica um modelo estatístico de linguagem no cálculo medidas de perplexidade para sentenças a serem ranqueadas. Para os experimentos deste trabalho, o modelo de linguagem empregado pelo sistema SRILM é treinado com base em um superconjunto do corpus paralelo de entrada do Módulo de Treinamento, composto por 710000 sentenças simples extraídas do Simple Wikipedia. O modelo é constituído pelas estatísticas dos unigramas, bigramas e trigramas presentes nas 710000 sentenças do corpus de treinamento. No modelo não são incluídas as estatísticas dos quadrigramas e pentagramas do corpus pois, assim como apontado pelo estudo de [Chen e Goodman 1996], as estatísticas de modelos de linguagem de ordem mais alta tendem a ser esparsas, e portanto pouco informativas caso o corpus de treinamento seja ruidoso ou caso o corpus não

seja grande extenso o suficiente para representar toda ou grande parte da estrutura da língua.

6.1 Avaliação de Desempenho Geral

No âmbito de PLN, os métodos de avaliação automática são recursos frequentemente utilizados na avaliação de desempenho de sistemas de Tradução Automática. Em geral, estes métodos de avaliação calculam a similaridade entre traduções confeccionadas pelo sistema, com traduções ótimas produzidas por linguistas e/ou falantes nativos, referidas como "*Gold Standards*" na literatura.

Neste experimento, utilizamos algumas das principais métricas de avaliação automática da literatura na avaliação de desempenho do sistema de simplificação proposto. São dois os objetivos deste experimento: estimar a qualidade das versões simplificadas produzidas pelo sistema proposto, e descobrir se a ordem de execução das etapas de simplificação em nível sintático e lexical impacta na qualidade das simplificações.

Para realizar esta avaliação, foram escolhidas 100 sentenças complexas, extraídas de forma arbitrária de uma porção reservada do corpus paralelo utilizado pelo Módulo de Treinamento. As sentenças possuem versões simples diferentes da versão complexa original, e variam entre 60 e 180 caracteres em comprimento. Estas sentenças foram simplificadas pelo sistema proposto sob duas configurações:

- **Configuração Regular**: Nesta configuração, o Módulo de Simplificação do sistema proposto submete a sentença complexa primeiramente à etapa de simplificação em nível sintático, e em seguida à etapa de simplificação em nível lexical.

- **Configuração Inversa**: Nesta configuração, o Módulo de Simplificação do sistema proposto submete a sentença complexa primeiramente à etapa de simplificação em nível lexical, e em seguida à etapa de simplificação em nível sintático.

As métricas de avaliação automática escolhidas para este experimento são BLEU, NIST e Meteor. A métrica BLEU [Papineni et al. 2002] determina a similaridade entre uma versão simplificada produzida pelo sistema e uma versão simplificada de referência com base na similaridade entre seus N-gramas. A Equação 6.1 descreve a forma como esta métrica é calculada.

$$\text{Bleu} = \text{BP} \cdot \left(\sum_{n=1}^{N} w_n \log(p_n) \right) \tag{6.1}$$

Na Equação 6.1, o valor BP define uma penalidade calculada com base na brevidade de uma determinada sentença. Os possíveis valores assumidos pelo valor BP para uma sentença produzida pelo sistema com c caracteres de comprimento dentre um conjunto de r sentenças, é calculado por meio da Equação 6.2.

$$\text{BP} = 1 \quad \text{se} \quad c > r, \quad \text{ou} \quad e^{(1-r/c)} \quad \text{se} \quad c \leqslant r \tag{6.2}$$

A variável w_n da Equação 6.1 define um peso para os N-gramas de tamanho n no cálculo da média ponderada, que é constante para todos N-gramas e é calculado com base no número total de N-gramas presentes em uma sentença. E a variável p_n define a proporção de N-gramas da versão simplificada produzida pelo sistema que também estão presentes na versão simplificada de referência. O valor da métrica BLEU varia entre 0, nos casos onde não existe similaridade entre as duas versões, e 1, nos casos onde as duas sentenças são idênticas.

A métrica NIST [Doddington 2002] foi desenvolvida com base na métrica BLEU, porém possui alguns incrementos. Ao invés de atribuir probabilidades idênticas a cada N-grama da sentença, assim como feito no cálculo da métrica BLEU, a métrica NIST atribui um peso diferente a cada N-grama das sentenças com base no nível de informação do mesmo. Isto significa que, quanto mais raro é o N-grama na língua em que está representado, mais importante é o N-grama no cálculo da similaridade entre uma versão do sistema e uma versão de referência. O cálculo da métrica NIST é realizado por meio da Equação 6.3.

$$\text{NIST} = \sum_{n=1}^{N} \left\{ \sum_{w_n \, \epsilon \, Overlap} Info(w_n) / \sum_{w_n \, \epsilon \, Sys} (1) \right\} \cdot exp \left\{ \beta \log^2 \left[min \left(\frac{L_{Sys}}{L_{Ref}}, 1 \right) \right] \right\} \tag{6.3}$$

O significado das variáveis, conjuntos e funções presentes na Equação 6.3 são:

- w_n: N-grama de tamanho n.
- $Overlap$: Conjunto de N-gramas da versão produzida pelo sistema que também podem ser encontrados na versão de referência.

- Sys: Conjunto de N-gramas da versão produzida pelo sistema.
- β: Penalidade para sentenças muito curtas, com valor 0.5.
- L_{Sys}: Número de palavras na versão produzida pelo sistema.
- L_{Ref}: Número de palavras na versão de referência.
- $Info(w_n)$: Nível de informação referente a um N-grama w_n. A função $Info$ é calculada pela Equação 6.4.

$$Info(w_n) = \log_2 \left(\frac{quantidade(w_1 ... w_{n-1})}{quantidade(w_1 ... w_n)} \right) \tag{6.4}$$

Na Seção 3.1, o Meteor é descrito como um sistema de alinhamento textual projetado para auxiliar na avaliação do desempenho de sistemas de tradução. Diferente das métricas BLEU e NIST, o Meteor avalia a qualidade de sentenças produzidas por sistemas de tradução alinhando-as a versões de referência, e então realizando um cálculo de similaridade sobre os dados de alinhamento obtidos. O cálculo de similaridade realizado é descrito pela Equação 6.5.

$$\text{Meteor} = (1 - Pen) \cdot F_{mean} \tag{6.5}$$

O coeficiente Pen da Equação 6.5 representa uma penalidade pela esparsidade entre os alinhamentos de palavras, e F_{mean} representa a média harmônica entre as medidas de Precisão e *Recall* referentes aos dados de alinhamento. A medida de Precisão se refere à proporção de alinhamentos ótimos dentre os alinhamentos de palavra obtidos, enquanto a medida de *Recall* se refere à razão entre o número de alinhamentos ótimos obtidos e o total de alinhamentos ótimos existentes. Neste contexto, os alinhamentos ótimos são todos aqueles que seriam obtidos no caso da sentença de referência ser idêntica à sentença produzida pelo sistema sendo avaliado. Mais detalhes sobre como são calculados os componentes do sistema Meteor podem ser encontrados em [Denkowski e Lavie 2011].

6.1.1 Resultados

A Tabela 6.1 descreve os resultados obtidos por meio da aplicação das métricas BLEU, NIST e METEOR sobre as 100 versões simplificadas produzidas pelo sistema proposto em ambas

configurações Regular (simplificação lexical após sintática) e Inversa (simplificação sintática após lexical).

	BLEU	**NIST**	**Meteor**
Configuração Regular	0.3354	4.9895	0.2787
Configuração Inversa	0.3136	4.9540	0.2778

Tabela 6.1: Resultados obtidos na Avaliação de Desempenho Geral

Os resultados obtidos nas tarefas de avaliação de sistemas de Tradução Automática documentados em [Callison-Burch et al. 2010], [Callison-Burch et al. 2011] e [Callison-Burch et al. 2012] permitem estimar quais os níveis de pontuação necessários para que um determinado sistema de tradução seja considerado preciso. De acordo com os resultados obtidos por esses estudos, as pontuações das métricas BLEU e Meteor, para alguns dos sistemas de tradução mais eficientes, variam entre 0.7 e 0.9, enquanto a pontuação calculada pela métrica NIST oscila entre 7 e 8.

Analisando os dados da Tabela 6.1, nota-se que o sistema proposto não atingiu a pontuação adequada em nenhuma das métricas de avaliação automática. A grande similaridade entre os valores das métricas BLEU, NIST e Meteor, para ambas configurações, indica que não existe uma diferença substancial entre a qualidade das versões produzidas pelas duas configurações do sistema. As baixas pontuações obtidas revelam também que as sentenças produzidas pelo sistema são bastante distintas das sentenças simples de referência. Os exemplos da Tabela 6.2 ilustram este fenômeno.

Sentença Complexa:	The original flag was adopted on April 4, 1959, when Mali joined the Mali Federation.
Referência Simples:	The original flag became official on April 4, 1959, when Mali joined the Mali Federation.
Conf. Regular:	The original flag was created on April 4, 1959, when the Mali Federation.
Conf. Inversa:	The original flag was on April 4, 1959.
Sentença Complexa:	He remained in office until his death in 1642; he was succeeded by Jules Cardinal Mazarin, whose career he fostered.
Referência Simples:	He remained in office until his death in 1642; then Jules Cardinal Mazarin became chief minister.
Conf. Regular:	He remained in office until his death in 1642.
Conf. Inversa:	He remained in branch until his death in 1642.
Sentença Complexa:	Cage won and retained the title, but Abyss attacked him and stole the belt.
Referência Simples:	Christian won but Abyss took the title belt.
Conf. Regular:	Cage won and kept the title, but attacked him.
Conf. Inversa:	Abyss, but attacked him and stole the belt.
Sentença Complexa:	Kirilenko reached #18, her career-high singles ranking, on the WTA tour in July 2008.
Referência Simples:	Kirilenko reached #19, her career-high singles ranking, on the WTA tour in June 2008.
Conf. Regular:	Kirilenko reached her career-high singles ranking in July 2008.
Conf. Inversa:	#18, her career-high singles ranking, on the WTA tour in July 2008.

Tabela 6.2: Sentenças simplificadas produzidas pelo sistema proposto

Analisando os exemplos da Tabela 6.2, observa-se que, apesar da notável diferença entre as versões produzidas pelo sistema e as versões de referência, nem todas as versões produzidas pelo sistema são de qualidade inferior à versão de referência. As sentenças produzidas pela Configuração Regular da segunda e quarta entradas da Tabela 6.2, por exemplo, sintetizam o conteúdo principal da sentença complexa original de forma mais eficiente do que a versão simplificada de referência. Este fenômeno evidencia que o processo de simplificação de uma

sentença complexa pode ser realizado de múltiplas formas. As métricas de avaliação automática escolhidas não são capazes de considerar esta característica da simplificação na pontuação de sentenças simplificadas, podendo fazer com que as pontuações de versões simplificadas de boa qualidade seja reduzida.

Nota-se também que em grande parte dos exemplos da Tabela 6.2, as versões simplificadas de ambas configurações foram produzidas pelos processos de remoção de segmentos da sentença complexa original e substituição de termos complexos por equivalentes simples. A segunda entrada da Tabela 6.2 para a Configuração Regular ilustra um exemplo onde o processo de remoção de segmentos foi capaz de simplificar a sentença complexa sem comprometer sua gramaticalidade ou significado. Entretanto, nos exemplos de ambas Configurações da primeira e terceira entrada da Tabela 6.2, a remoção de segmentos não produziu o mesmo efeito, comprometendo ambas gramaticalidade e coerência das versões simplificadas.

A grande quantidade de versões simplificadas sem segmentos essenciais à compreensão da sentença complexa original, incoerentes ou com erros gramaticais, leva a acreditar que o sistema proposto não é capaz de produzir regras de simplificação com a qualidade necessária para que a estrutura sintática de uma sentença complexa possa ser simplificada de forma adequada. Uma das possíveis causas para a baixa qualidade das regras de simplificação produzidas é a grande quantidade de entradas incoerentes no corpus paralelo de treinamento, como por exemplo no caso da quarta entrada da Tabela 6.2, onde as sentenças complexa e simples não possuem o mesmo significado. Outra possível causa para este problema é a técnica utlizada pelo algoritmo de inferência de regras do sistema T3, que favorece regras as quais removem grandes porções de sentenças complexas.

6.2 Avaliação de Desempenho de Componentes

A Avaliação de Desempenho de Componentes é um experimento cujo intuito é descobrir se existe diferença na eficiência com que o sistema de simplificação proposto realiza as etapas de simplificação em nível sintático e lexical. A avaliação da qualidade individual dos processos de simplificação permite que sejam identificados com mais precisão, caso existam, os fatores que levam à má qualidade das versões simplificadas produzidas. Caso exista grande diferença na qualidade das versões simplificadas em nível sintático e lexical, é possível que sejam estu-

dadas formas adequadas de se modificar ou incrementar o sistema de simplificação de modo a aumentar a qualidade das sentenças simples produzidas.

Neste experimento, a arquitetura do Módulo de Simplificação do sistema proposto é ligeiramente modificada para que este possa realizar as etapas de simplificação em níveis sintático e lexical individualmente, e não em sequência, assim como ilustrado na Figura 5.6 da Seção 5.2. A Figura 6.1 ilustra o fluxo de execução da versão modificada do Módulo de Simplificação.

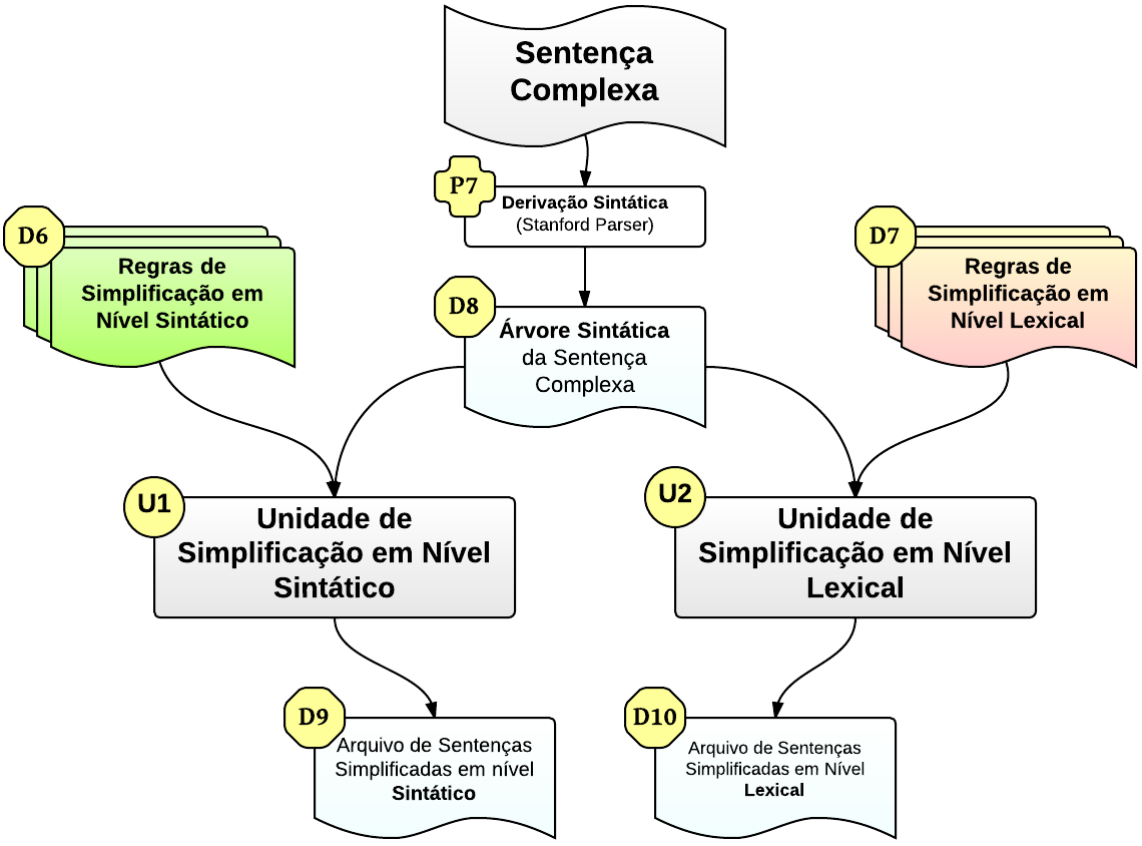


Figura 6.1: Fluxograma de execução do Módulo de Simplificação modificado

Observa-se que, ao invés de produzir apenas uma lista de versões simplificadas em ambos níveis sintático e lexical, o Módulo de Simplificação agora produz duas listas de versões simplificadas distintas. Ao final da execução do Módulo de Simplificação, as duas listas de versões simplificadas são direcionadas ao Módulo de Ranqueamento. O Módulo de Ranqueamento é então executado duas vezes, para que pontue cada lista individualmente e forneça como saída tanto a melhor versão simplificada em nível sintático, quanto em nível lexical.

Uma vez modificado, o sistema foi submetido na simplificação de 130 sentenças complexas

retiradas de forma arbitrária de uma porção reservada do corpus paralelo utilizado pelo Módulo de Treinamento. As sentenças variam entre 50 e 240 caracteres em comprimento, e cada uma possui uma versão simplificada distinta da versão complexa no corpus paralelo. Como a versão modificada do sistema produz duas versões simplificadas para cada sentença complexa, foi produzido um total de 260 versões simplificadas. As técnicas empregadas na availação da qualidade das versões simplificadas são:

1. **Avaliação automática:** As versões simplificadas das 130 sentenças complexas foram avaliadas pela métrica BLEU [Papineni et al. 2002]. Esta métrica, descrita na Seção 6.1, compara as versões simplificadas produzidas pelo sistema com as versões simples de referência encontradas no corpus paralelo, e assim calcula um coeficiente de similaridade entre elas.

2. **Avaliação humana:** Para complementar a avaliação automática, foram contratados 5 falantes não-nativos da língua inglesa para avaliar manualmente a qualidade das 260 versões simplificadas. A cada avaliador foi designado um folheto de avaliação contendo um conjunto de 100 versões simplificadas: 40 versões simplificadas individuais ao avaliador, e 60 comuns aos 5 avaliadores. Aos avaliadores foi requisitado que respondessem as seguintes perguntas com respeito a cada uma das 100 sentenças:

   - A versão simplificada é mais simples do que a versão complexa?
   - A versão simplificada é livre de erros gramaticais?
   - A versão simplificada possui o mesmo significado da versão complexa?

   As possíveis respostas para cada pergunta são “Sim”, “Não”, ou também “Não aplicável”, destinada a casos onde, por exemplo, a sentença tem erros gramaticais demais para ter sua simplicidade julgada. Para evitar que os avaliadores se confundam no processo de avaliação das sentenças, o folheto de avaliação possui um pequeno tutorial ensinando como a avaliação deve ser feita.

6.2.1 Resultados

Na etapa de avaliação automática, tanto as versões simplificadas em nível sintático quanto em nível lexical atingiram uma pontuação de 0.342 pela métrica BLEU. Isto indica que, em teoria, não existe diferença de qualidade entre as versões simplificadas em nível sintático e lexical. Porém, devido à natureza do problema da simplificação de textos, não é prudente que sejam tiradas conclusões a respeito do desempenho do sistema de simplificação com base apenas nestas pontuações, uma vez que, assim como constatado no experimento da Seção 6.1, métricas de avaliação automática não representam com eficiência o potencial de sistemas de simplificação automática.

Na etapa de avaliação humana, foi utilizada a métrica de Kappa-Cohen [Carletta 1996] de concordância entre avaliadores no objetivo de verificar quão similares foram as avaliações realizadas. O valor da métrica de Kappa-Cohen mede a concordância entre um par de avaliadores, e varia entre 0, indicando que não existe qualquer concordância entre os avaliadores, e 1, indicando que os avaliadores concordaram em todos os julgamentos feitos. Os valores de concordância entre cada combinação de pares de avaliadores variou entre 0.32 (baixo) e 0.68 (substancial), indicando que, em geral, houve considerável discordância entre os julgamentos feitos pelos mesmos. A Tabela 6.3 lista as estatísticas coletadas da combinação dos julgamentos dos cinco avaliadores com respeito à simplicidade, corretude gramatical e coerência das 260 versões simplificadas.

Nível de Simplificação	Sintático	Lexical
Simplicidade	26.12%	42.15%
Gramaticalidade	38.28%	83.40%
Coerência	34.68%	56.50%

Tabela 6.3: Resultados obtidos na etapa de avaliação humana

No processo de simplificação das sentenças, o Módulo de Treinamento foi capaz de produzir 562033 regras de simplificação em nível sintático, e 4809 regras de simplificação em nível lexical. O grande número de regras de simplificação em nível sintático garantiu que cada uma das sentenças complexas pôde ser simplificada de ao menos uma forma. Entretanto, nota-se que apenas cerca de 26% das versões simplificadas em nível sintático são de fato mais simples

que suas versões originais, e em cerca de 65% dos casos a versão simplificada possui erros gramaticais e/ou é incoerente.

Por meio de uma inspeção manual das regras de simplificação em nível sintático, foi possível observar que grande parte delas são ruidosas. Isto significa que grande parte das regras modificam a estrutura da sentença complexa de forma indesejada, removendo segmentos essenciais à estrutura das sentenças (como sujeitos e objetos), ou então reordenando trechos de forma a tornar a versão simplificada incompreensível e/ou incoerente.

Diferente do caso da simplificação em nível sintático, os resultados da avaliação humana para as versões simplificadas em nível lexical são mais encorajadores. Em mais de 40% dos casos o sistema teve sucesso em criar uma versão mais simples que sua equivalente complexa, e também adicionou erros gramaticais em apenas cerca de 15% das versões simplificadas. Contudo, em quase metade dos casos a versão simplificada perdeu o significado original da sentença complexa, e grande parte das sentenças complexas tiveram apenas uma palavra complexa substituída por uma equivalente simples no processo de simplificação.

Na Seção 6.1 foi constatado que grande parte das versões simplificadas produzidas pelo sistema proposto é ausente de segmentos essenciais à sua gramaticalidade, e também à compreensão do significado da sentença complexa original. Com os resultados da Avaliação de Desempenho de Componentes é possível constatar que o processo responsável por este fenômeno é a etapa de simplificação em nível sintático. Esta limitação do sistema proposto pode ser endereçada por meio de novos algoritmos de inferência de regras de simplificação, e também pela aplicação de técnicas de seleção de regras capazes de identificar e descartar regras de simplificação de baixa qualidade.

6.3 Avaliação de Métricas Alternativas de Ranqueamento

Neste experimento são avaliadas a eficiência de diferentes métricas no ranqueamento de sentenças de acordo com sua simplicidade. O objetivo deste experimento é encontrar alternativas eficientes para a complementação da métrica de pontuação de sentenças simples aplicada pelo Módulo de Ranqueamento do sistema proposto. As métricas escolhidas para avaliação são algumas das descritas em Coh-Metrix [Graesser et al. 2004].

O Coh-Metrix é um sistema que analisa as propriedades das estruturas lexical e sintática de

sentenças da língua inglesa por meio de 108 métricas distintas. Estas métricas avaliam diferentes aspectos da estrutura de sentenças da língua inglesa, como por exemplo sua densidade sintática, complexidade sintática, nível de informação de palavras, legibilidade, coesão e outros. O projeto Coh-Metrix tem como objetivos principais fornecer uma plataforma gratuita para a avaliação automática da qualidade de textos, e também sugerir alternativas para o desenvolvimento de novas estratégias de Tradução, Simplificação, Sumarização e outros.

Para este experimento foram escolhidas 9 métricas de densidade e complexidade sintática dentre as descritas em [Graesser et al. 2004]. Estas métricas calculam a complexidade de uma sentença com base no número de ocorrências de determinadas construções gramaticais em sua estrutura sintática. As descrições das métricas escolhidas são:

- **DRNP:** Número de frases nominais presentes na estrutura sintática da sentença. O segmento "*The boy*" da sentença "*The boy reads comics.*" é um exemplo de frase nominal, composta pelo determinante "*The*" e pelo sujeito "*boy*". O cálculo desta métrica é feito por meio da contagem de ocorrências de nós da classe NP ("*Noun Phrase*") na árvore sintática da sentença.

- **DRVP:** Número de frases verbais presentes na estrutura sintática da sentença. O segmento "*reads comics*" da sentença "*The boy reads comics.*" é um exemplo de frase verbal, composta pelo verbo "*reads*" e pelo objeto "*comics*". O cálculo desta métrica é feito por meio da contagem de ocorrências de nós da classe VP ("*Verbal Phrase*") na árvore sintática da sentença.

- **DRAP:** Número de frases adverbiais presentes na estrutura sintática da sentença. O segmento "*soon*" da sentença "*She will be arriving soon.*" é um exemplo de frase adverbial, composta pelo advérbio de tempo "*soon*". O cálculo desta métrica é feito por meio da contagem de ocorrências de nós da classe $ADVP$ ("*Adverbial Phrase*") na árvore sintática da sentença.

- **DRPP:** Número de frases preposicionais presentes na estrutura sintática da sentença. O segmento "*at home*" da sentença "*I like to work at home.*" é um exemplo de frase preposicional, composta pela preposição "*at*" e pelo objeto "*home*". O cálculo desta métrica é

feito por meio da contagem de ocorrências de nós da classe PP (“*Prepositional Phrase*”) na árvore sintática da sentença.

- **DRPVAL:** Número de segmentos sem agente escritos na voz passiva da sentença. Este tipo de segmento é caracterizado pelo uso da voz passiva, bem como pela ausência do agente que exerce uma determinada ação sobre um objeto. A sentença “*The car was stolen, and now the owner may need to file a robbery.*” é um exemplo de sentença que contém um segmento deste tipo: observa-se que o segmento escrito na voz passiva “*The car was stolen*” não deixa explícito qual o agente que exerceu a ação de roubo sobre o objeto carro. O cálculo desta métrica é feito por meio da contagem de ocorrências de subárvores da estrutura sintática da sentença cujos dois últimos nós folha da esquerda para a direita são, em sequência, das classes VBD (“*Past Tense Verb*”) e VP (“*Verbal Phrase*”).

- **DRNEG:** Número de frases de negação presentes na sentença. O segmento “*not going to miss you*” da sentença “*I am not going to miss you.*” é um exemplo de frase de negação, composta pelo advérbio de negação “*not*” e pela frase verbal “*going to miss you*”. O cálculo desta métrica é feito por meio da contagem de ocorrências de nós da classe VP, $ADVP$ e PP precedidos por advérbios de negação como “*not*” e “*’nt*”.

- **DRGERUND:** Número de verbos no gerúndio presentes na sentença. Os verbos “*swimming*”, “*walking*” e “*judging*” são exemplos de verbos conjugados no tempo gerúndio da língua inglesa. O cálculo desta métrica é feito por meio da contagem de ocorrências de nós da classe VBG (“*Gerund Verb*”) na árvore sintática da sentença.

- **DRINF:** Número de verbos no infinitivo presentes na sentença. Os verbos “*swim*”, “*walk*” e “*judge*” são exemplos de verbos conjugados no tempo infinitivo da língua inglesa. O cálculo desta métrica é feito por meio da contagem de ocorrências de nós da classe VB (“*Infinitive Verb*”) na árvore sintática da sentença.

- **SYNNP:** Número médio de adjetivos por frase nominal da sentença. A frase nominal “*A perky and smiley toddler*” da sentença “*A perky and smiley toddler won the beauty pageant contest.*” possui dois adjetivos: “*perky*” e “*smiley*”. O cálculo desta métrica é

feito pela média de nós de classe JJ presentes nas subárvores da estrutura sintática da sentença enraizadas em nós de classe NP ("*Nominal Phrase*").

A avaliação da eficiência das métricas escolhidas é feita com base na correlação entre ranqueamentos produzidos por estas, e ranqueamentos ótimos (ou "*Gold Standard*") produzidos manualmente por um falante não-nativo da língua inglesa. Para o cálculo de correlação foram selecionadas 30 sentenças complexas de forma arbitrária de uma porção reservada do corpus paralelo utilizado pelo Módulo de Treinamento. O sistema proposto foi empregado na produção das 10 melhores versões simplificadas de cada uma das sentenças complexas, totalizando assim 30 conjuntos de 10 sentenças.

Estes conjuntos de sentenças são os dados de referência os quais são ordenados pelas métricas descritas anteriormente, e então comparados ao "*Gold Standard*" construído. A construção do "*Gold Standard*" é também realizada sobre os 30 conjuntos de sentenças selecionados. No processo de construção do "*Gold Standard*", as 10 versões simplificadas de cada conjunto foram organizadas em ordem decrescente de simplicidade por um falante não-nativo da língua inglesa. A reordenação das sentenças por um falante da língua inglesa garante que o "*Gold Standard*" se aproxime ao máximo dos resultados de ranqueamento ótimos desejados por possíveis usuários do sistema. Este processo permite também que sejam identificadas possíveis limitações do Módulo de Ranqueamento por meio da comparação entre os ranqueamentos originais produzidos pelo sistema, e os ranqueamentos reordenados produzidos pelo falante da língua inglesa.

Para o cálculo da correlação entre os ranqueamentos das métricas escolhidas e o "*Gold Standard*", foram escolhidos os coeficientes de correlação de Spearman [Spearman 1904] e Kendall-Tau [Kendall 1938]. O coeficiente de Spearman calcula a dependência estatística entre dois ranqueamentos distintos com base nas distâncias entre seus índices. A Equação 6.6 descreve o cálculo do coeficiente de correlação de Spearman.

$$\text{Spearman} = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{6.6}$$

As descrições dos componentes da Equação 6.6 são:

- d_i^2**:** Quadrado da diferença d entre os índices de uma determinada sentença i no ranqueamento de referência e o ranqueamento produzido pela métrica escolhida.

- n**:** Número total de sentenças presentes no ranqueamento de referência. Como todos os 30 conjuntos de teste produzidos possuem 10 sentenças, a variável n toma o valor 10 para todos os conjuntos.

O coeficiente de Kendall-Tau é também uma métrica que pode ser empregada no cálculo da correlação entre dois ranqueamentos. Diferente da técnica utilizada por Spearman, o coeficiente de Kendall-Tau é calculado com base na concordância entre cada par de índices dos ranqueamentos em questão. Sejam (m_i, m_j) e (r_i, r_j) os pares de índices nas posições distintas i e j do ranqueamento produzido pela métrica em questão e o ranqueamento de referência, respectivamente. Estes pares são considerados concordantes somente quando $m_i > m_j$ e $r_i > r_j$, ou então quando $m_i < m_j$ e $r_i < r_j$. Com base neste princípio, o cálculo do coeficiente de Kendall-Tau para dois ranqueamentos de n componentes toma a forma representada na Equação 6.7.

$$\text{Kendall-Tau} = \frac{(\text{numero de pares concordantes}) - (\text{numero de pares discordantes})}{\frac{n(n-1)}{2}} \tag{6.7}$$

Ambos coeficientes de Spearman e Kendall-Tau variam entre 1, nos casos onde existe correlação positiva absoluta entre os dois ranqueamentos, e -1, nos casos onde existe correlação negativa absoluta entre os mesmos.

6.3.1 Resultados

Na etapa de construção do "*Gold Standard*" foi constatado que, na maioria dos casos, o Módulo de Ranqueamento não é capaz de ranquear com pontuação máxima a sentença de maior qualidade dentre as 10 versões simplificadas de cada conjunto. Para 28 dos 30 conjuntos de versões simplificadas escolhidos, a sentença de maior simplicidade não é a mesma no ranqueamento original produzido pelo sistema proposto e no ranqueamento reordenado pelo falante não-nativo da língua inglesa. Isto indica que, por mais que o Módulo de Simplificação confeccione versões simplificadas de qualidade para uma determinada sentença complexa, em grande parte dos casos o Módulo de Ranqueamento não será capaz de posicionar estas sentenças de qualidade no topo da lista ranqueada.

Uma vez construído o "*Gold Standard*" deste experimento, cada uma das 9 métricas de ranqueamento de sentenças selecionadas para este experimento foi empregada no ranqueamento

dos 30 conjuntos de 10 versões simplificadas produzidos pelo sistema proposto. As Tabelas 6.4 e 6.5 listam os resultados obtidos pela aplicação dos coeficientes de correlação de Spearman e Kendall-Tau sobre os ranqueamentos produzidos pelas métricas escolhidas e o "*Gold Standard*".

	Média Aritmética	**Máximo Positivo**	**Máximo Negativo**
DRNP	-0.0242	0.8545	-0.9393
DRVP	-0.0614	0.7454	-0.7818
DRAP	-0.0666	0.6121	-0.7454
DRPP	-0.0161	0.7454	-0.8181
DRVAL	-0.0210	0.6121	-0.7454
DRNEG	0.0016	0.6121	-0.6727
DRGERUND	0.0921	0.7818	-0.6727
DRINF	0.0004	0.6121	-0.6727
SYNNP	0.0949	0.8787	-0.7454

Tabela 6.4: Valores de correlação de Spearman das métricas de ranqueamento avaliadas

	Média Aritmética	**Máximo Positivo**	**Máximo Negativo**
DRNP	-0.0088	0.6888	-0.8222
DRVP	-0.0503	0.6000	-0.6000
DRAP	-0.0533	0.4222	-0.5555
DRPP	-0.0162	0.5555	-0.6888
DRVAL	-0.0222	0.4666	-0.5555
DRNEG	-0.0044	0.4666	-0.5555
DRGERUND	0.0622	0.5111	-0.5555
DRINF	-0.0029	0.4666	-0.5555
SYNNP	0.0725	0.6888	-0.5555

Tabela 6.5: Valores de correlação de Kendall-Tau das métricas de ranqueamento avaliadas

Os valores de correlação listados nas Tabelas 6.4 e 6.5 revelam que nenhuma das 9 métricas de ranqueamento escolhidas é capaz de ranquear sentenças de forma similar à especificada no "*Gold Standard*". Os valores médios de correlação variam entre -0.0666 e 0.0949, o que significa que nenhuma das métricas obteve uma similaridade média positiva ou negativa superior a 10% (-0.1 ou 0.1) com relação ao "*Gold Standard*".

Outro fenômeno que pode ser observado nos dados de ambas Tabelas 6.4 e 6.5 é a grande diferença entre os valores de correlação máxima positiva e máxima negativa de cada métrica. Considere por exemplo os valores de correlação máxima e mínima da métrica de ranqueamento DRNP da Tabela 6.4. O valor de correlação máxima desta métrica é 0.8545. Isto significa que, para algum dos 30 conjuntos de sentenças, a métrica DRNP obteve mais de 85% de similaridade com o "*Gold Standard*". Contudo, o valor de correlação máxima negativa é também bastante acentuado, atingindo quase 94% (-0.9393) de dissimilaridade com o "*Gold Standard*" para algum dos 30 conjuntos. A grande variação entre os valores de correlação máxima positiva e negativa das métricas avaliadas revela que a simplicidade da sentença não se relaciona diretamente com a quantidade de determinadas construções gramaticais presentes em sua estrutura sintática.

Os resultados obtidos mostram que nenhuma das 9 métricas do projeto Coh-Metrix avaliadas neste experimento são confiáveis no ranqueamento de sentenças simplificadas. Para que o Módulo de Ranqueamento do sistema proposto possa ser incrementado, haverá de ser conduzida uma nova etapa de avaliações em trabalhos futuros, investigando métricas de ranqueamento mais sofisticadas. Dentre estas métricas estão as medidas de legibilidade OVIX e NR, descritas no trabalho de [Keskisarkka 2012].

Capítulo 7

Considerações Finais

A produção de textos para cobrir as muitas conquistas, transformações e avanços obtidos, tanto campo das tecnologias quanto do relacionamento humano, em termos sociais, políticos e econômicos, têm forçado as pessoas a lerem mais. No entanto, parece que na contramão desse processo, um fenômeno curioso e preocupante vem acontecendo: a quantidade de pessoas com dificuldade de leitura e compreensão vem aumentando na mesma proporção. O uso de recursos linguísticos mais elaborados, sem dúvida alguma, imprime aos textos maior elegância. Contudo, o seu uso pode tornar os textos incompreensíveis por parte de quem os lê. Os motivos que podem levar os textos a se tornarem herméticos para os leitores são vários, mas, dentre eles, destacam-se o uso de estruturas sintáticas muito complexas e a ocorrência de palavras ou expressões de conhecimento restristo ou de domínio específico.

Simplificar documentos manualmente é uma tarefa que requer grande disponibilidade de tempo, com a demanda de altos investimentos na contratação de profissionais qualificados para a tarefa. Por mais que a simplificação manual possa ser empregada na confecção de versões facilmente compreensíveis de grandes obras da literatura e da ciência, esta tarefa não é aplicável no âmbito da distribuição de documentos digitais. Em ambientes como a Internet, por exemplo, um grande volume de informações novas é disponibilizado todos os dias, e a simplificação deste conteúdo demandaria a contratação contínua de uma grande quantidade de profissionais qualificados, o que inviabilizaria a tarefa devido aos grandes custos relacionados.

Para desenvolver uma estratégia de simplificação automática de texto, é necessária uma boa compreensão sobre o problema a ser resolvido, bem como sobre quais as técnicas e ferramentas disponíveis para resolvê-lo. Apesar de existir um número relativamente pequeno de trabalhos relacionados à simplificação automática de texto, a leitura dos mesmos é essencial no desenvol-

vimento de uma nova estratégia de simplificação. É importante saber não só quais estratégias de simplificação já foram testadas, mas também quais suas limitações e de que forma podem ser aprimoradas.

Na etapa de revisão bibliográfica deste trabalho, foi possível notar que grande parte dos trabalhos publicados concentram-se em estudar o potencial de alguma estratégia específica para a simplificação automática, como por exemplo a aplicação de regras de simplificação confeccionadas manualmente, a substituição de palavras complexas por palavras simples, ou também a tradução do inglês complexo ao inglês simples por meio de modelos tradicionais de tradução automática probabilística. No objetivo de endereçar algumas das limitações encontradas nestas estratégias, neste trabalho foi desenvolvido um sistema de simplificação em nível sintático-Lexical, que simplifica sentenças em inglês pelo processo de transdução de árvores.

O sistema proposto, bem como os experimentos realizados, embora ainda requeiram muitas revisões, já nos permitem perceber o seu potencial. A Avaliação de Desempenho Geral, documentada na Seção 6.1, evidenciou que tanto o corpus paralelo de treinamento, quanto o método de produção automática de regras de simplificação do sistema proposto requerem ajustes, uma vez que a qualidade das versões simplificadas é, na maioria dos casos, comprometida pela remoção de segmentos essenciais para a compreensão da sentença. Neste experimento foi constatado também que a ordem da aplicação dos processos de simplificação em nível sintático e lexical não impacta de forma significativa na qualidade da simplificação, e também que existe muito trabalho a ser realizado com respeito ao desenvolvimento de metricas de avaliação automática para sistemas de Simplificação.

Os resultados obtidos na Avaliação de Desempenho de Componentes permitiu que fossem identificadas algumas das limitações do sistema proposto. Apesar das estatísticas com respeito ao desempenho da etapa de simplificação em nível lexical serem encorajadoras, é evidente que a etapa de simplificação em nível sintático deve ser remodelada para que possa produzir versões simplificadas de qualidade. Dentre as possíveis soluções para esta limitação estão a reestruturação do corpus paralelo de treinamento, a elaboração de um novo algoritmo de inferência de regras de simplificação, ou então o uso de técnicas mais inteligentes de seleção de regras.

O Módulo de Ranqueamento também há de ser reestruturado para que o sistema proposto possa confeccionar versões simplificadas de maior qualidade. Na Avaliação de Métricas Alter-

nativas de Ranqueamento foi constatado que, em grande parte dos casos, o Módulo de Ranqueamento não é capaz de determinar com sucesso qual a versão simplificada de maior qualidade dentre as versões confeccionadas pelo Módulo de Simplificação. Considerando que as métricas de ranqueamento do projeto Coh-Metrix avaliadas neste trabalho apresentaram baixa correlação com os resultados ótimos de referência, é possível concluir que a qualidade da sentença simplificada não é diretamente relacionada ao número de certas construções sintáticas em sua estrutura, como frases nominais, verbais, prepositivas e adverbiais. Dentre as atividades que podem levar ao aumento do desempenho do Módulo de Ranqueamento estão um estudo aprofundado com respeito aos aspectos que mais influenciam na qualidade de uma versão simplificada, e também a avaliação de métricas de ranqueamento de sentenças mais sofisticadas.

Considerando o estudo realizado e os resultados obtidos nos experimentos, conclui-se que a simplificação em nível sintático-lexical por meio da transdução de árvores é uma estratégia de simplificação promissora, e que merece ser estudada mais a fundo. Apesar da aparente baixa qualidade das regras de simplificação em nível sintático produzidas pelo sistema proposto, a associação entre os processos de simplificação em nível sintático e lexical permite que sentenças complexas sejam simplificadas de múltiplas formas distintas simultaneamente, como por exemplo pela remoção de segmentos desimportantes e pela substituição de termos complexos. A tarefa de transdução de árvores elimina a necessidade da produção manual de regras de simplificação, porém ainda existe a necessidade de serem investigadas formas mais eficientes da aplicação desta tarefa no processo de simplificação automática de textos.

Dentre as possíveis atividades a serem desenvolvidas como trabalhos futuros, estão:

- A confecção de um novo corpus paralelo de sentenças complexas e simples, contendo apenas pares de sentenças coerentes em significado e ausentes de erros ortográficos e gramaticais.

- O desenvolvimento de novos algoritmos de inferência de regras de simplificação, capazes de produzir regras que representem operações de simplificação que não comprometam a integridade da sentença complexa quando aplicadas.

- A implementação de novos algoritmos de seleção e aplicação de regras, capazes de filtrar regras de má qualidade e decidir quais as regras mais adequadas para uma determinada

sentença complexa com base em suas características.

- O desenvolvimento de um estudo sobre o que caracteriza sentenças simplificadas de qualidade, e a avaliação de novas técnicas de ranqueamento de sentenças capazes de distinguir sentenças simplificadas de qualidade dentre as demais.

Referências Bibliográficas

[Aho, Sethi e Ullman 1986]AHO, A. V.; SETHI, R.; ULLMAN, J. D. *Compilers: principles, techniques, and tools*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1986. ISBN 0-201-10088-6.

[Bach et al. 2011]BACH, N. et al. Tris: A statistical sentence simplifier with log-linear models and margin-based discriminative training. In: *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, 2011. p. 474–482. Disponível em: <http://www.aclweb.org/anthology/I11-1053>.

[Blake et al. 2007]BLAKE, C. et al. Query expansion, lexical simplification and sentence selection strategies for multi-document summarization. In: *Proceedings of Document Understanding Conference*. [S.l.: s.n.], 2007.

[Bott e Saggion 2011]BOTT, S.; SAGGION, H. An unsupervised alignment algorithm for text simplification corpus construction. In: *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. [S.l.]: Association for Computational Linguistics, 2011.

[Brown et al. 1993]BROWN, P. F. et al. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, MIT Press, 1993.

[Callison-Burch et al. 2010]CALLISON-BURCH, C. et al. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In: *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. [S.l.]: Association for Computational Linguistics, 2010.

[Callison-Burch et al. 2011]CALLISON-BURCH, C. et al. Findings of the 2011 workshop on statistical machine translation. In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. [S.l.]: Association for Computational Linguistics, 2011.

[Callison-Burch et al. 2012]CALLISON-BURCH, C. et al. Findings of the 2012 workshop on statistical machine translation. In: *Proceedings of the Seventh Workshop on Statistical Machine Translation*. [S.l.]: Association for Computational Linguistics, 2012.

[Carletta 1996]CARLETTA, J. Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.*, MIT Press, 1996.

[Carroll et al. 1998]CARROLL, J. et al. Practical simplification of english newspaper text to assist aphasic readers. In: *AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*. [S.l.: s.n.], 1998.

[Chandrasekar, Doran e Srinivas 1996]CHANDRASEKAR, R.; DORAN, C.; SRINIVAS, B. Motivations and methods for text simplification. In: *16th International Conference on Computational Linguistics*. [S.l.: s.n.], 1996.

[Chandrasekar e Srinivas 1997]CHANDRASEKAR, R.; SRINIVAS, B. *Automatic Induction of Rules for Text Simplification*. 1997.

[Chen e Goodman 1996]CHEN, S. F.; GOODMAN, J. An empirical study of smoothing techniques for language modeling. In: . Stroudsburg, PA, USA: Association for Computational Linguistics, 1996. (ACL '96).

[Cohn e Lapata 2008]COHN, T.; LAPATA, M. Sentence compression beyond word deletion. In: *Proceedings of the 22nd International Conference on Computational Linguistics*. Manchester, UK: [s.n.], 2008. (COLING '08), p. 137–144.

[Cohn e Lapata 2009]COHN, T.; LAPATA, M. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 2009.

[Coster e Kauchak 2011]COSTER, W.; KAUCHAK, D. Simple english wikipedia: a new text simplification task. In: *Proceedings of the 49th Annual Meeting of the Association for Com-*

putational Linguistics: Human Language Technologies. Portland, Oregon: [s.n.], 2011. p. 665–669.

[Coster e Kauchak 2011]COSTER, W.; KAUCHAK, D. Simple english wikipedia: A new text simplification task. In: *ACL (Short Papers)*. [S.l.]: The Association for Computer Linguistics, 2011. p. 665–669.

[Dagan, Church e Gale 1993]DAGAN, I.; CHURCH, K. W.; GALE, W. A. Robust bilingual word alignment for machine aided translation. In: *In Proceedings of the Workshop on Very Large Corpora*. [S.l.: s.n.], 1993.

[Denkowski e Lavie 2011]DENKOWSKI, M.; LAVIE, A. Meteor 1.3: automatic metric for reliable optimization and evaluation of machine translation systems. In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011.

[Doddington 2002]DODDINGTON, G. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: *Proceedings of the second international conference on Human Language Technology Research*. [S.l.]: Morgan Kaufmann Publishers Inc., 2002. (HLT '02).

[Dunbar 1994]DUNBAR, G. Philip quinlan, oxford psycholinguistics database. oxford: Oxford university press, 1992. *Journal of Child Language*, v. 21, p. 513–516, 5 1994. ISSN 1469-7602.

[Gasperin et al. 2009]GASPERIN, C. et al. Natural language processing for social inclusion: a text simplfication architecture for different literacy levels. In: *SEMISH-XXXVI*. [S.l.: s.n.], 2009.

[Graesser et al. 2004]GRAESSER, A. et al. Coh-Metrix: Analysis of text on cohesion and language. 2004.

[Hinton e Sejnowski 1999]HINTON, G.; SEJNOWSKI, T. *Unsupervised Learning: Foundations of Neural Computation*. Mit Press, 1999. (A Bradford Book). ISBN 9780262581684. Disponível em: <http://books.google.ca/books?id=yj04Y0lje4cC>.

[Jurafsky e Martin 2008]JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing (2nd Edition) (Prentice Hall Series in Artificial Intelligence)*. 2. ed. [S.l.]: Prentice Hall, 2008.

[Kandula, Curtis e Zeng-Treitler 2010]KANDULA, S.; CURTIS, D.; ZENG-TREITLER, Q. A semantic and syntactic text simplification tool for health content. *AMIA Symposium*, 2010.

[Kendall 1938]KENDALL, M. G. A new measure of rank correlation. *Biometrika*, Biometrika Trust, v. 30, p. 81–93, 1938.

[Keskisarkka 2012]KESKISARKKA, R. *Automatic Text Simplification via Synonym Replacement*. Dissertação (Master Thesis) — Linkoping University, 2012.

[Klein e Manning 2003]KLEIN, D.; MANNING, C. D. Accurate unlexicalized parsing. In: *41st Annual Meeting of the Association for Computational Linguistics*. [S.l.: s.n.], 2003.

[Koehn 2005]KOEHN, P. Europarl: A Parallel Corpus for Statistical Machine Translation. In: *Conference Proceedings: the tenth Machine Translation Summit*. [s.n.], 2005. Disponível em: <http://mt-archive.info/MTS-2005-Koehn.pdf>.

[Lal e Ruger 2002]LAL, P.; RUGER, S. Extract-based summarization with simplification. In: *Proceedings of the ACL 2002 Automatic Summarization / DUC 2002 Workshop*. [S.l.: s.n.], 2002.

[Marcus, Marcinkiewicz e Santorini 1993]MARCUS, M. P.; MARCINKIEWICZ, M. A.; SANTORINI, B. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.*, MIT Press, 1993.

[Mooney 2013]MOONEY, R. Natural language processing: Statistical parsing. 2013.

[Och e Ney 2000]OCH, F. J.; NEY, H. Improved statistical alignment models. In: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. [S.l.]: Association for Computational Linguistics, 2000. p. 440–447.

[Och e Ney 2003]OCH, F. J.; NEY, H. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, MIT Press, 2003.

[Osborne 2013]OSBORNE, M. Machine translation word-based models and the em algorithm. 2013.

[Papineni et al. 2002]PAPINENI, K. et al. Bleu: a method for automatic evaluation of machine translation. In: *ACL02*. Philadelphia, Pennsylvania: [s.n.], 2002. p. 311–318.

[Porter 2001]PORTER, M. F. *Snowball: A language for stemming algorithms*. 2001. Disponível em: <http://snowball.tartarus.org/texts/introduction.html>.

[Prather et al. 1997]PRATHER, P. et al. Speed of lexical activation in nonfluent broca's aphasia and fluent wernicke's aphasia. *Brain Lang*, v. 59, 1997.

[Siddharthan 2004]SIDDHARTHAN, A. *Syntactic simplification and text cohesion*. [S.l.], 2004.

[Smith e Eisner 2006]SMITH, D. A.; EISNER, J. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In: *In Proceedings of the HLTNAACL Workshop on Statistical Machine Translation*. [S.l.: s.n.], 2006.

[Spearman 1904]SPEARMAN, C. The proof and measurement of association between two things. *American Journal of Psychology*, p. 88–103, 1904.

[Specia 2010]SPECIA, L. Translating from complex to simplified sentences. In: *Proceedings of the 9th international conference on Computational Processing of the Portuguese Language*. Porto Alegre, RS, Brazil: Springer-Verlag, 2010. (PROPOR'10), p. 30–39.

[Specia 2012]SPECIA, L. Machine translation: Statistical mt. 2012.

[Stevenson 2012]STEVENSON, M. N-gram language modelling. 2012.

[Stolcke 2002]STOLCKE, A. SRILM – an extensible language modeling toolkit. In: *Proceedings of ICSLP*. Denver, USA: [s.n.], 2002.

[Stolcke 2007]STOLCKE, A. *Documentation of the smoothing implementations used by SRILM*. 2007.

[Stolcke et al. 2011]STOLCKE, A. et al. SRILM at sixteen: Update and outlook. In: *IEEE Automatic Speech Recognition and Understanding Workshop*. [s.n.], 2011. Disponível em: <http://t3-1.yum2.net/index/www.speech.sri.com/papers/asru2011-srilm.pdf>.

[Watanabe et al. 2009]WATANABE, W. M. et al. Facilita: reading assistance for low-literacy readers. In: *27th ACM*. [S.l.: s.n.], 2009. (SIGDOC).

[Weisman 2013]WEISMAN, J. *G.O.P. Is Resisting Obama Pressure on Tax Increase*. February 2013.

[Woodsend e Lapata 2011]WOODSEND, K.; LAPATA, M. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In: . [S.l.: s.n.], 2011. (EMNLP). ISBN 978-1-937284-11-4.

[WordNet 1998]WORDNET. [S.l.]: MIT Press, 1998.

[Wubben, Bosch e Krahmer 2012]WUBBEN, S.; BOSCH, A. van den; KRAHMER, E. Sentence simplification by monolingual machine translation. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. [S.l.: s.n.], 2012.

[Yatskar et al. 2010]YATSKAR, M. et al. For the sake of simplicity: unsupervised extraction of lexical simplifications from wikipedia. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. [S.l.]: Association for Computational Linguistics, 2010. (HLT '10), p. 365–368.

[Zhu, Bernhard e Gurevych 2010]ZHU, Z.; BERNHARD, D.; GUREVYCH, I. A monolingual tree-based translation model for sentence simplification. In: *Proceedings of the 23rd International Conference on Computational Linguistics*. Beijing, China: [s.n.], 2010. (COLING '10), p. 1353–1361.